

Unittest for socket_protocol

January 11, 2021

Contents

1	Test Information	4
1.1	Test Candidate Information	4
1.2	Unittest Information	4
1.3	Test System Information	4
2	Statistic	4
2.1	Test-Statistic for testrun with python 2.7.18 (final)	4
2.2	Test-Statistic for testrun with python 3.8.5 (final)	5
2.3	Coverage Statistic	5
3	Tested Requirements	6
3.1	Message Object	6
3.1.1	Status	6
3.1.2	Service-ID	7
3.1.3	Data-ID	8
3.1.4	Data	9
3.2	Communication	10
3.2.1	A full Message Object including the defined properties and data shall be transfered.	10
3.2.2	A checksum shall ensure the correct transmittion	11
3.2.3	An authentication between server and client shall be possible including status feedback methods	12
3.2.4	An automatic authentication shall available	14
3.2.5	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.	15
3.2.6	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts	17
3.2.7	Define a channel name for the server and client after connection is established	19
3.2.8	The User shall be able to define a new service	21
3.2.9	Registration of already registered request Service-ID or response Service-ID shall not be possible .	22
3.3	Callbacks	23
3.3.1	It shall be possible to register a callback for a specific Service- and Data-ID	23
3.3.2	It shall be possible to register a callback for a specific Service-ID and all Data-IDs	24

3.3.3	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs	25
3.3.4	It shall be possible to register a callback for all incoming messages	26
3.3.5	Callback choice, if several callbacks are available (caused by wildcard callbacks)	27
3.4	Some additional Information and Passthrough Methods	28
3.4.1	Connection established information	28
3.4.2	Is connected information	29
3.4.3	Reconnect Method	30
3.5	Deprepeated struct protocol	32
3.5.1	A full Message Object including the defined properties and data shall be transferred.	32
A	Trace for testrun with python 2.7.18 (final)	34
A.1	Tests with status Info (22)	34
A.1.1	Status	34
A.1.2	Service-ID	34
A.1.3	Data-ID	35
A.1.4	Data	36
A.1.5	A full Message Object including the defined properties and data shall be transferred.	37
A.1.6	A checksum shall ensure the correct transmission	43
A.1.7	An authentication between server and client shall be possible including status feedback methods	49
A.1.8	An automatic authentication shall available	63
A.1.9	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.	72
A.1.10	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts	86
A.1.11	Define a channel name for the server and client after connection is established	96
A.1.12	The User shall be able to define a new service	108
A.1.13	Registration of already registered request Service-ID or response Service-ID shall not be possible .	115
A.1.14	It shall be possible to register a callback for a specific Service- and Data-ID	119
A.1.15	It shall be possible to register a callback for a specific Service-ID and all Data-IDs	129
A.1.16	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs	134
A.1.17	It shall be possible to register a callback for all incoming messages	139
A.1.18	Callback choice, if several callbacks are available (caused by wildcard callbacks)	144

A.1.19	Connection established information	156
A.1.20	Is connected information	166
A.1.21	Reconnect Method	169
A.1.22	A full Message Object including the defined properties and data shall be transfered.	175
B	Trace for testrun with python 3.8.5 (final)	179
B.1	Tests with status Info (22)	179
B.1.1	Status	179
B.1.2	Service-ID	180
B.1.3	Data-ID	181
B.1.4	Data	182
B.1.5	A full Message Object including the defined properties and data shall be transfered.	183
B.1.6	A checksum shall ensure the correct transmittion	189
B.1.7	An authentication between server and client shall be possible including status feedback methods	194
B.1.8	An automatic authentication shall available	208
B.1.9	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.	218
B.1.10	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts	232
B.1.11	Define a channel name for the server and client after connection is established	242
B.1.12	The User shall be able to define a new service	254
B.1.13	Registration of already registered request Service-ID or response Service-ID shall not be possible .	261
B.1.14	It shall be possible to register a callback for a specific Service- and Data-ID	265
B.1.15	It shall be possible to register a callback for a specific Service-ID and all Data-IDs	274
B.1.16	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs	279
B.1.17	It shall be possible to register a callback for all incoming messages	285
B.1.18	Callback choice, if several callbacks are available (caused by wildcard callbacks)	290
B.1.19	Connection established information	301
B.1.20	Is connected information	311
B.1.21	Reconnect Method	315
B.1.22	A full Message Object including the defined properties and data shall be transfered.	321
C	Test-Coverage	325
C.1	socket_protocol	325
C.1.1	socket_protocol.__init__.py	325

1 Test Information

1.1 Test Candidate Information

The Module `socket_protocol` is designed for point to point communication for client-server issues. For more Information read the sphinx documentation.

Library Information	
Name	socket_protocol
State	Released
Supported Interpreters	python2, python3
Version	2a04c5bb4a5918290c3998176093afae
Dependencies	
stringtools	09b4d1c41b828c8d1ccb723fa1fd79a9

1.2 Unittest Information

Unittest Information	
Version	da1e6bd881c6d5dd865ce87d511e702f
Testruns with	python 2.7.18 (final), python 3.8.5 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	Linux Mint 20.1 ulyssa
Hostname	ahorn
Kernel	5.4.0-60-generic (#67-Ubuntu SMP Tue Jan 5 18:31:36 UTC 2021)
Machine	x86_64
Path	/user_data/data/dirk/prj/unittest/socket_protocol/unittest
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 2.7.18 (final)

Number of tests	22
Number of successfull tests	22
Number of possibly failed tests	0
Number of failed tests	0
Executionlevel	Full Test (all defined tests)
Time consumption	19.276s

2.2 Test-Statistic for testrun with python 3.8.5 (final)

Number of tests	22
Number of successfull tests	22
Number of possibly failed tests	0
Number of failed tests	0
Executionlevel	Full Test (all defined tests)
Time consumption	19.209s

2.3 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
socket_protocol	100.0%	100.0%
socket_protocol.__init__.py	100.0%	

3 Tested Requirements

3.1 Message Object

A Message Object shall hold the following information for transmission.

3.1.1 Status

Description

The Status shall hold some general information (in most cases it is used by the responder). Examples: Okay, Service or Data unknown, Operation not permitted, Authentication required, ...

Reason for the implementation

Give the possibility to transfer additional status information (e.g. to explain negative responses).

Fitcriterion

A Status is part of the Message Object and it is holding the Status information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.1!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (26)
Start-Time:	2021-01-11 11:25:03,879
Finished-Time:	2021-01-11 11:25:03,880
Time-Consumption	0.001s
Testsummary:	
Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	status is part of the message object is correct ('status' is in the list or dict).
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	status is part of the message object is correct ('status' is in the list or dict).
Success	Content in message object for status is correct (Content 'S' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.1!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (26)
Start-Time:	2021-01-11 11:25:25,021
Finished-Time:	2021-01-11 11:25:25,021
Time-Consumption	0.001s
Testsummary:	

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success	status is part of the message object is correct ('status' is in the list or dict).
Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success	status is part of the message object is correct ('status' is in the list or dict).
Success	Content in message object for status is correct (Content 'S' and Type is <class 'str'>).

3.1.2 Service-ID

Description

The Service-ID shall hold information about the type of the request / corresponding response. Examples: read request, write request, read response, write response, ...

Reason for the implementation

Give the requestor the possibility to use different types (Services) for a transfer.

Fitcriterion

A Service-ID is part of the Message Object and it is holding the Service-ID information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.2!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (27)
Start-Time:	2021-01-11 11:25:03,880
Finished-Time:	2021-01-11 11:25:03,881
Time-Consumption	0.001s

Testsummary:

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
Success	Content in message object for service_id is correct (Content 'SID' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.2!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (27)
Start-Time:	2021-01-11 11:25:25,022
Finished-Time:	2021-01-11 11:25:25,022

Time-Consumption 0.001s

Testsummary:

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success	service_id is part of the message object is correct ('service_id' is in the list or dict).
Success	Content in message object for service_id is correct (Content 'SID' and Type is <class 'str'>).

3.1.3 Data-ID

Description

The Data-ID shall hold information to differtiate the data for a specific Service.

Reason for the implementation

Give the possibility to transfer different information for each Service.

Fitcriterion

A Data-ID is part of the Message Object and it is holding the Data-ID information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.3!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (28)
Start-Time:	2021-01-11 11:25:03,881
Finished-Time:	2021-01-11 11:25:03,881
Time-Consumption	0.001s

Testsummary:

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Success	Content in message object for data_id is correct (Content 'DID' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.3!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (28)
Start-Time:	2021-01-11 11:25:25,022

Finished-Time: 2021-01-11 11:25:25,023
Time-Consumption 0.001s

Testsummary:

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Success	Content in message object for data_id is correct (Content 'DID' and Type is <class 'str'>).

3.1.4 Data

Description

The Data shall hold the data to be transfered. For the most requests not data is transmitted.

Reason for the implementation

Give the possibility to transfer Data.

Fitcriterion

Data is part of the Message Object and it is holding the Data information.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.4!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (29)
Start-Time:	2021-01-11 11:25:03,882
Finished-Time:	2021-01-11 11:25:03,882
Time-Consumption	0.001s

Testsummary:

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	data is part of the message object is correct ('data' is in the list or dict).
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	data is part of the message object is correct ('data' is in the list or dict).
Success	Content in message object for data is correct (Content 'D' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.4!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (29)

Start-Time: 2021-01-11 11:25:25,023
Finished-Time: 2021-01-11 11:25:25,023
Time-Consumption 0.000s

Testsummary:

Info	Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}
Success	data is part of the message object is correct ('data' is in the list or dict).
Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success	data is part of the message object is correct ('data' is in the list or dict).
Success	Content in message object for data is correct (Content 'D' and Type is <class 'str'>).

3.2 Communication

3.2.1 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.5!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/...init....py (33)
Start-Time:	2021-01-11 11:25:03,882
Finished-Time:	2021-01-11 11:25:04,636
Time-Consumption	0.754s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on server side is correct (Content {'u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transfered', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on client side is correct (Content {'u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transfered', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.5!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (33)
Start-Time:	2021-01-11 11:25:25,023
Finished-Time:	2021-01-11 11:25:25,775
Time-Consumption	0.752s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

3.2.2 A checksum shall ensure the correct transmission**Description**

If the checksum does not fit to the checksum of the transferred data, the message will be ignored, because the complete content including the Service- and Data-ID is possibly corrupted.

Reason for the implementation

Ensure correct data transfer.

Fitterion

Corrupted message is not in the receive buffer after transmission.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.6!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (34)
Start-Time:	2021-01-11 11:25:04,636
Finished-Time:	2021-01-11 11:25:05,599
Time-Consumption	0.962s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success	Checksum Error → No message received by server is correct (Content None and Type is <type 'NoneType'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).
Success	Checksum Error → No message received by client is correct (Content None and Type is <type 'NoneType'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.6!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (34)
Start-Time:	2021-01-11 11:25:25,776
Finished-Time:	2021-01-11 11:25:26,730
Time-Consumption	0.955s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Checksum Error → No message received by server is correct (Content None and Type is <class 'NoneType'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success	Checksum Error → No message received by client is correct (Content None and Type is <class 'NoneType'>).

3.2.3 An authentication between server and client shall be possible including status feedback methods

Description

The Client shall have a method to initiate the authentication. In case that the server and the client do have identical secrets, the authentication shall be successfull.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

Check authentication method feedback (client) and authentication feedback (client and server), in case of differing and identical secrets.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.7!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (35)

Start-Time: 2021-01-11 11:25:05,599
Finished-Time: 2021-01-11 11:25:06,567
Time-Consumption 0.968s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	No secret set
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of server is correct (Content True and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <type 'bool'>).
Info	Different secrets set
Success	Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <type 'bool'>).
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <type 'bool'>).
Info	Identical secrets set
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content True and Type is <type 'bool'>).
Success	Authentication state of server is correct (Content True and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <type 'bool'>).
Info	Corrupting the authentication mechanism
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.7!

Testrun: python 3.8.5 (final)
Caller: /user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (35)
Start-Time: 2021-01-11 11:25:26,731
Finished-Time: 2021-01-11 11:25:27,698
Time-Consumption 0.968s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	No secret set
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of server is correct (Content True and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <class 'bool'>).
Info	Different secrets set

Success	Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <class 'bool'>).
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <class 'bool'>).
Info	Identical secrets set
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content True and Type is <class 'bool'>).
Success	Authentication state of server is correct (Content True and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <class 'bool'>).
Info	Corrupting the authentication mechanism
Info	Performing Authentication
Success	Return Value of authentication method is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <class 'bool'>).

3.2.4 An automatic authentication shall available

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

Check authentication feedback (client and server) after connect has been triggered.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.8!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (36)
Start-Time:	2021-01-11 11:25:06,568
Finished-Time:	2021-01-11 11:25:09,278
Time-Consumption	2.710s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set and automatic authentication
Success	Authentication state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <type 'bool'>).
Info	Connecting Server and Client
Success	Authentication state of server is correct (Content True and Type is <type 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.8!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unitest/socket_protocol/unitest/src/tests/...init....py (36)
Start-Time:	2021-01-11 11:25:27,699
Finished-Time:	2021-01-11 11:25:30,406
Time-Consumption	2.707s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set and automatic authentication
Success	Authentication state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Authentication state of server is correct (Content True and Type is <class 'bool'>).
Success	Authentication state of client is correct (Content True and Type is <class 'bool'>).

3.2.5 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.

Description

Communication (rx and tx) shall be disabled, if a secret is given. Except of a response for registered services, saying that a Authentication is required.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

RX and TX is not possible, till a successfull authentication has been performed.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.9!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unitest/socket_protocol/unitest/src/tests/...init....py (37)
Start-Time:	2021-01-11 11:25:09,278
Finished-Time:	2021-01-11 11:25:11,255
Time-Consumption	1.977s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Setting a Server secret and no Client secret
Info	Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 3, u'service_id': 31, u'data': None, u'data_id': 36} and Type is <class 'socket_protocol.data_storage'>).

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Identical secrets set

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Performing Authentication

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.9!

Testrun: python 3.8.5 (final)
 Caller: /user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (37)
 Start-Time: 2021-01-11 11:25:30,407
 Finished-Time: 2021-01-11 11:25:32,380
 Time-Consumption 1.973s

Testsummary:

Info Setting up communication

Info Connecting Server and Client

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Success Received message on server side is correct (Content {'data_id': 36, 'service_id': 31, 'status': 3, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Info Identical secrets set

Info Transferring a message client → server

Success	Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).
Success	Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success	Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info	Performing Authentication
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

3.2.6 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts

Description

It shall be possible to add a specific message, identified by Service-ID and Data-ID, to a whitelist. All messages added to that whitelist shall be transmitted and received, if no authentication was successfull performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentication mechanism.

Fitcriterion

Transmition and Reception will be enabled, after the message has been added to the whitelist.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.10!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/_init_.py (38)
Start-Time:	2021-01-11 11:25:11,256
Finished-Time:	2021-01-11 11:25:13,741
Time-Consumption	2.485s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).
Success	Received message on server side is correct (Content None and Type is <type 'NoneType'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Added msg1 to client whitelist (sid=17, did=34)

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Added msg1 to server whitelist (sid=17, did=34)

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Info Added msg2 to client and server whitelist (sid=17, did=35)

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.10!

Testrun: python 3.8.5 (final)
 Caller: /user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (38)
 Start-Time: 2021-01-11 11:25:32,381
 Finished-Time: 2021-01-11 11:25:34,859
 Time-Consumption 2.479s

Testsummary:

Info Setting up communication

Info Connecting Server and Client

Info Identical secrets set

Info Transferring a message client → server

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

Info Transferring a message server → client

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

```

Info      Added msg1 to client whitelist (sid=17, did=34)
Info      Transferring a message client → server
Success   Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success   Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info      Transferring a message server → client
Success   Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success   Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info      Added msg1 to server whitelist (sid=17, did=34)
Info      Transferring a message client → server
Success   Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success   Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0,
'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).
Info      Transferring a message server → client
Success   Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success   Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info      Added msg2 to client and server whitelist (sid=17, did=35)
Info      Transferring a message client → server
Success   Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success   Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0,
'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).
Info      Transferring a message server → client
Success   Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success   Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4,
'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```

3.2.7 Define a channel name for the server and client after connection is established

Description

After the connection is established, the client will initiate the channel name exchange. The channel name defined on the client side will be dominant.

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

Perform a channel name exchange with no channel name definition, differing channel name definition and identical channel name definition. In all cases, the channel name of the client will be used. Perform two channel name exchanges with only one channel name definition. This definition will be used.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.11!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (39)
Start-Time:	2021-01-11 11:25:13,742
Finished-Time:	2021-01-11 11:25:15,491

Time-Consumption 1.749s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting no Channel name for server and client
Success	Channel name of server is correct (Content None and Type is <type 'NoneType'>).
Success	Channel name of client is correct (Content None and Type is <type 'NoneType'>).
Info	Setting different Channel names for client and Server
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'client' and Type is <type 'str'>).
Success	Channel name of client is correct (Content 'client' and Type is <type 'str'>).
Info	Setting identical Channel names for client and server
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'unittest' and Type is <type 'str'>).
Success	Channel name of client is correct (Content 'unittest' and Type is <type 'str'>).
Info	Setting Channel name for client only
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'client' and Type is <type 'str'>).
Success	Channel name of client is correct (Content 'client' and Type is <type 'str'>).
Info	Setting Channel name for server only
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'server' and Type is <type 'str'>).
Success	Channel name of client is correct (Content 'server' and Type is <type 'str'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.11!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/_init_.py (39)
Start-Time:	2021-01-11 11:25:34,861
Finished-Time:	2021-01-11 11:25:36,604
Time-Consumption	1.744s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting no Channel name for server and client
Success	Channel name of server is correct (Content None and Type is <class 'NoneType'>).
Success	Channel name of client is correct (Content None and Type is <class 'NoneType'>).
Info	Setting different Channel names for client and Server
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'client' and Type is <class 'str'>).
Success	Channel name of client is correct (Content 'client' and Type is <class 'str'>).
Info	Setting identical Channel names for client and server
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'unittest' and Type is <class 'str'>).
Success	Channel name of client is correct (Content 'unittest' and Type is <class 'str'>).

Info	Setting Channel name for client only
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'client' and Type is <class 'str'>).
Success	Channel name of client is correct (Content 'client' and Type is <class 'str'>).
Info	Setting Channel name for server only
Info	Connecting Server and Client
Success	Channel name of server is correct (Content 'server' and Type is <class 'str'>).
Success	Channel name of client is correct (Content 'server' and Type is <class 'str'>).

3.2.8 The User shall be able to define a new service

Description

The service is defined by a Request Service-ID and a Response Service-ID.

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

Define a service and check, that the server will respond on the new Service-ID. The Status shall be "Request has no callback. Data buffered.", because no callback is registered for that request.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.12!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (40)
Start-Time:	2021-01-11 11:25:15,492
Finished-Time:	2021-01-11 11:25:16,353
Time-Consumption	0.861s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server → client
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on server side is correct (Content None and Type is <type 'NoneType'>).
Info	Adding service to server instance for the transmit message
Info	Transferring a message client → server → client
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on server side is correct (Content {u'status': 1, u'service_id': 18, u'data': None, u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.12!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (40)
Start-Time:	2021-01-11 11:25:36,605
Finished-Time:	2021-01-11 11:25:37,462
Time-Consumption	0.857s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server → client
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info	Adding service to server instance for the transmit message
Info	Transferring a message client → server → client
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content {'data_id': 34, 'service_id': 18, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

3.2.9 Registration of already registered request Service-ID or response Service-ID shall not be possible

Description

An exception shall be raised, if a service registration with an existing request SID or response SID is performed.

Reason for the implementation

Changing existing services will create strange situations with already registered callbacks.

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.13!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/___init___py (41)
Start-Time:	2021-01-11 11:25:16,353
Finished-Time:	2021-01-11 11:25:16,710
Time-Consumption	0.357s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Adding a service with an already registered request SID
Success	Expected Exception RequestSidExistsError was triggered
Info	Adding a service with an already registered response SID
Success	Expected Exception ResponseSidExistsError was triggered

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.13!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/...init....py (41)
Start-Time:	2021-01-11 11:25:37,463
Finished-Time:	2021-01-11 11:25:37,817
Time-Consumption	0.355s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Adding a service with an already registered request SID
Success	Expected Exception RequestSidExistsError was triggered
Info	Adding a service with an already registered response SID
Success	Expected Exception ResponseSidExistsError was triggered

3.3 Callbacks

3.3.1 It shall be possible to register a callback for a specific Service- and Data-ID

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.14!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/...init....py (45)
Start-Time:	2021-01-11 11:25:16,710
Finished-Time:	2021-01-11 11:25:17,680
Time-Consumption	0.970s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Overwriting existing Callback using one with faulty (too many) return values
Info	Transferring data
Success	Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {u'status': 2, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing the registered Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content None and Type is <type 'NoneType'>).

Success Message received by client is correct (Content {u'status': 1, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.14!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/...init....py (45)
Start-Time:	2021-01-11 11:25:37,818
Finished-Time:	2021-01-11 11:25:38,778
Time-Consumption	0.960s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).
Info	Overwriting existing Callback using one with faulty (too many) return values
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 2, 'data': None} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing the registered Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content None and Type is <class 'NoneType'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

3.3.2 It shall be possible to register a callback for a specific Service-ID and all Data-IDs

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.15!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/...init....py (46)
Start-Time:	2021-01-11 11:25:17,681
Finished-Time:	2021-01-11 11:25:18,242
Time-Consumption	0.561s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback

Info	Transferring data
Success	Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.15!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (46)
Start-Time:	2021-01-11 11:25:38,779
Finished-Time:	2021-01-11 11:25:39,336
Time-Consumption	0.557s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.3 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.16!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (47)
Start-Time:	2021-01-11 11:25:18,242
Finished-Time:	2021-01-11 11:25:18,801
Time-Consumption	0.559s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.16!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (47)
Start-Time:	2021-01-11 11:25:39,336
Finished-Time:	2021-01-11 11:25:39,894
Time-Consumption	0.557s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.4 It shall be possible to register a callback for all incoming messages**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.17!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (48)
Start-Time:	2021-01-11 11:25:18,802
Finished-Time:	2021-01-11 11:25:19,359
Time-Consumption	0.557s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'u'status': 0, 'u'service_id': 10, 'u'data': 31, 'u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'u'status': 0, 'u'service_id': 11, 'u'data': 33, 'u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.17!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (48)
Start-Time:	2021-01-11 11:25:39,894
Finished-Time:	2021-01-11 11:25:40,450
Time-Consumption	0.556s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.5 Callback choice, if several callbacks are available (caused by wildcard callbacks)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.18!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/_init_.py (49)
Start-Time:	2021-01-11 11:25:19,360
Finished-Time:	2021-01-11 11:25:20,534
Time-Consumption	1.174s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 0, 'service_id': 11, 'data': 33, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data- and Service-ID
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 6, 'service_id': 11, 'data': 34, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Service-ID and all Data-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 6, 'service_id': 11, 'data': 35, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data-ID and all Service-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'status': 0, 'service_id': 10, 'data': 31, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'status': 0, 'service_id': 11, 'data': 36, 'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.18!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (49)
Start-Time:	2021-01-11 11:25:40,450
Finished-Time:	2021-01-11 11:25:41,617
Time-Consumption	1.167s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data- and Service-ID
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 34} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Service-ID and all Data-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 35} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing Callback for a specific Data-ID and all Service-IDs
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 36} and Type is <class 'socket_protocol.data_storage'>).

3.4 Some additional Information and Passthrough Methods

3.4.1 Connection established information

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.19!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/___init___py (53)
Start-Time:	2021-01-11 11:25:20,534
Finished-Time:	2021-01-11 11:25:21,349
Time-Consumption	0.815s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <type 'bool'>).
Success	Server connection status is correct (Content True and Type is <type 'bool'>).
Success	Client connection status is correct (Content False and Type is <type 'bool'>).
Success	Server connection status is correct (Content False and Type is <type 'bool'>).
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <type 'bool'>).
Success	Server connection status is correct (Content True and Type is <type 'bool'>).
Info	Adding secrets to socket_protocol
Success	Client connection status is correct (Content False and Type is <type 'bool'>).
Success	Server connection status is correct (Content False and Type is <type 'bool'>).
Info	Doing authentication
Success	Client connection status is correct (Content True and Type is <type 'bool'>).
Success	Server connection status is correct (Content True and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.19!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/...init....py (53)
Start-Time:	2021-01-11 11:25:41,618
Finished-Time:	2021-01-11 11:25:42,424
Time-Consumption	0.806s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <class 'bool'>).
Success	Server connection status is correct (Content True and Type is <class 'bool'>).
Success	Client connection status is correct (Content False and Type is <class 'bool'>).
Success	Server connection status is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Client connection status is correct (Content True and Type is <class 'bool'>).
Success	Server connection status is correct (Content True and Type is <class 'bool'>).
Info	Adding secrets to socket_protocol
Success	Client connection status is correct (Content False and Type is <class 'bool'>).
Success	Server connection status is correct (Content False and Type is <class 'bool'>).
Info	Doing authentication
Success	Client connection status is correct (Content True and Type is <class 'bool'>).
Success	Server connection status is correct (Content True and Type is <class 'bool'>).

3.4.2 Is connected information

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.20!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unitest/socket_protocol/unitest/src/tests/_init_.py (54)
Start-Time:	2021-01-11 11:25:21,350
Finished-Time:	2021-01-11 11:25:21,705
Time-Consumption	0.355s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Success	Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Info	Disconnecting Server and Client
Success	Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).
Success	Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.20!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unitest/socket_protocol/unitest/src/tests/_init_.py (54)
Start-Time:	2021-01-11 11:25:42,425
Finished-Time:	2021-01-11 11:25:42,780
Time-Consumption	0.356s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Success	Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Info	Disconnecting Server and Client
Success	Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).
Success	Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

3.4.3 Reconnect Method

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.21!

Testrun:	python 2.7.18 (final)
----------	-----------------------

Caller: /user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/...init....py (55)
 Start-Time: 2021-01-11 11:25:21,706
 Finished-Time: 2021-01-11 11:25:22,407
 Time-Consumption 0.701s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Success	Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Info	Disconnecting Server and Client
Success	Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).
Success	Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).
Info	Connecting Server and Client
Success	Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).
Success	Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.21!

Testrun: python 3.8.5 (final)
 Caller: /user_data/data/dirk/prj/unititest/socket_protocol/unititest/src/tests/...init....py (55)
 Start-Time: 2021-01-11 11:25:42,781
 Finished-Time: 2021-01-11 11:25:43,481
 Time-Consumption 0.700s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Success	Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Success	Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Info	Disconnecting Server and Client
Success	Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).
Success	Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).
Success	Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

3.5 Depreceated struct protocol

3.5.1 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.22!

Testrun:	python 2.7.18 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/...init....py (59)
Start-Time:	2021-01-11 11:25:22,408
Finished-Time:	2021-01-11 11:25:23,167
Time-Consumption	0.759s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on server side is correct (Content {'status': 0, 'service_id': 17, 'data': u'msg1_data_to_be_transfered', 'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).
Success	Received message on client side is correct (Content {'status': 4, 'service_id': 17, 'data': u'msg2_data_to_be_transfered', 'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Testresult

This test was passed with the state: **Success**. See also full trace in section B.1.22!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/...init....py (59)
Start-Time:	2021-01-11 11:25:43,481
Finished-Time:	2021-01-11 11:25:44,239
Time-Consumption	0.758s
Testsummary:	
Info	Setting up communication

Info	Connecting Server and Client
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

A Trace for testrun with python 2.7.18 (final)

A.1 Tests with status Info (22)

A.1.1 Status

Description

The Status shall hold some general information (in most cases it is used by the responder). Examples: Okay, Service or Data unknown, Operation not permitted, Authentication required, ...

Reason for the implementation

Give the possibility to transfer additional status information (e.g. to explain negative responses).

Fitcriterion

A Status is part of the Message Object and it is holding the Status information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'status': None, 'service_id': None, 'data':
↪ None, 'data_id': None} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'status': 'S', 'service_id': 'SID', 'data':
↪ 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Success Content in message object for status is correct (Content 'S' and Type is <type 'str'>).

Result (Content in message object for status): 'S' (<type 'str'>)

Expectation (Content in message object for status): result = 'S' (<type 'str'>)

A.1.2 Service-ID

Description

The Service-ID shall hold information about the type of the request / corresponding response. Examples: read request, write request, read response, write response, ...

Reason for the implementation

Give the requestor the possibility to use different types (Services) for a transfer.

Fitcriterion

A Service-ID is part of the Message Object and it is holding the Service-ID information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}

Success service_id is part of the message object is correct ('service_id' is in the list or dict).

Result (service_id is part of the message object): {'status': None, 'service_id': None, 'data': None, 'data_id': None} (<class 'socket_protocol.data_storage'>)

Expectation (service_id is part of the message object): 'service_id' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success service_id is part of the message object is correct ('service_id' is in the list or dict).

Result (service_id is part of the message object): {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (service_id is part of the message object): 'service_id' in result

Success Content in message object for service_id is correct (Content 'SID' and Type is <type 'str'>).

Result (Content in message object for service_id): 'SID' (<type 'str'>)

Expectation (Content in message object for service_id): result = 'SID' (<type 'str'>)

A.1.3 Data-ID**Description**

The Data-ID shall hold information to differtiate the data for a specific Service.

Reason for the implementation

Give the possibility to transfer different information for each Service.

Fitcriterion

A Data-ID is part of the Message Object and it is holding the Data-ID information.

Testresult

This test was passed with the state: **Success**.

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Result (data_id is part of the message object): {'status': None, 'service_id': None, 'data': None, 'data_id': None} (<class 'socket_protocol.data_storage'>)	
Expectation (data_id is part of the message object): 'data_id' in result	
Info	Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Result (data_id is part of the message object): {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)	
Expectation (data_id is part of the message object): 'data_id' in result	
Success	Content in message object for data_id is correct (Content 'DID' and Type is <type 'str'>).
Result (Content in message object for data_id): 'DID' (<type 'str'>)	
Expectation (Content in message object for data_id): result = 'DID' (<type 'str'>)	

A.1.4 Data

Description

The Data shall hold the data to be transfered. For the most requests not data is transmitted.

Reason for the implementation

Give the possibility to transfer Data.

Fitcriterion

Data is part of the Message Object and it is holding the Data information.

Testresult

This test was passed with the state: **Success**.

Info	Creating empty message object: {'status': None, 'service_id': None, 'data': None, 'data_id': None}
Success	data is part of the message object is correct ('data' is in the list or dict).
Result (data is part of the message object): {'status': None, 'service_id': None, 'data': None, 'data_id': None} (<class 'socket_protocol.data_storage'>)	

Expectation (data is part of the message object): 'data' in result

Info Creating a maximum message object: {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'status': 'S', 'service_id': 'SID', 'data': 'D', 'data_id': 'DID'} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Success Content in message object for data is correct (Content 'D' and Type is <type 'str'>).

Result (Content in message object for data): 'D' (<type 'str'>)

Expectation (Content in message object for data): result = 'D' (<type 'str'>)

A.1.5 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incomming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

```

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0

```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':
↳ u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)

Info Transferring a message server → client

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:

↪ "'msg2_data_to_be_transferred'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61

↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61

↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↪ 33 35 7d 20 18 19 e8 3a 3e

comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↪ 33 35 7d 20 18 19 e8 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61

↪ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a

↪ 20 33 35 7d 20 18 19 e8

prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:

↪ "u'msg2_data_to_be_transferred'"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive

↪ method

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {u'status': 4, u'service_id': 17, u'data':
↪ u'msg2_data_to_be_transferred', u'data_id': 35} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
↪ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
↪ 'socket_protocol.data_storage'>)

A.1.6 A checksum shall ensure the correct transmittion

Description

If the checksum does not fit to the checksum of the transferred data, the message will be ignored, because the complete content including the Service- and Data-ID is possibly corrupted.

Reason for the implementation

Ensure correct data transfer.

Fitcriterion

Corrupted message is not in the receive buffer after transmission.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and

↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the

↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the

↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the

↪ authentication whitelist

```

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0

```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9c 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9c
prot-server: Received message has an invalid checksum. Message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Checksum Error → No message received by server is correct (Content None and Type is <type 'NoneType'>).

Result (Checksum Error → No message received by server): None (<type 'NoneType'>)

Expectation (Checksum Error → No message received by server): result = None (<type 'NoneType'>)

Info Transferring a message server → client

```

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"

```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 35 7d 20 18 19 e8
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "u'msg2_data_to_be_transferred"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

```
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)
```

Success Checksum Error → No message received by client is correct (Content None and Type is <type 'NoneType'>).

Result (Checksum Error -> No message received by client): None (<type 'NoneType'>)

Expectation (Checksum Error -> No message received by client): result = None (<type
↪ 'NoneType'>)

A.1.7 An authentication between server and client shall be possible including status feedback methods

Description

The Client shall have a method to initiate the authentication. In case that the server and the client do have identical secrets, the authentication shall be successfull.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

Check authentication method feedback (client) and authentication feedback (client and server), in case of differing and identical secrets.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

```

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
```

```
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info No secret set

Info Performing Authentication

Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).

```
Result (Return Value of authentication method): False (<type 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<type 'bool'>)
```

Success Authentication state of server is correct (Content True and Type is <type 'bool'>).

```
Result (Authentication state of server): True (<type 'bool'>)
```

Expectation (Authentication state of server): result = True (<type 'bool'>)

Success Authentication state of client is correct (Content True and Type is <type 'bool'>).

Result (Authentication state of client): True (<type 'bool'>)

Expectation (Authentication state of client): result = True (<type 'bool'>)

Info Different secrets set

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

Result (Authentication state of server): False (<type 'bool'>)

Expectation (Authentication state of server): result = False (<type 'bool'>)

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

Result (Authentication state of client): False (<type 'bool'>)

Expectation (Authentication state of client): result = False (<type 'bool'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```

comm-client: TX -> (6): 10 4d cd 55 3a 3e
comm-server: RX <- (6): 10 4d cd 55 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'067cdeac66c847e7443344b6e133aeeedcb24ac9ca92d4cf39e52316f45c83d46'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 30 36 37 63 64 65 61 63
↳ 36 36 63 38 34 37 65 37 34 34 33
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 30 36 37 63 64 65 61 63
↳ 36 36 63 38 34 37 65 37 34 34 33
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 33 34 34 62 36 65 31 33 33 61 65 65 64 63 62 32 34 61 63 39 63 61 39
↳ 32 64 34 63 66 33 39 65 35 32 33 31 36 66 34 35 63 38 33 64 34 36 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 0e 37
comm-client: RX <- (64): 33 34 34 62 36 65 31 33 33 61 65 65 64 63 62 32 34 61 63 39 63 61 39
↳ 32 64 34 63 66 33 39 65 35 32 33 31 36 66 34 35 63 38 33 64 34 36 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 0e 37
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (4): 1a c4 3a 3e
comm-client: RX <- (4): 1a c4 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 30 36 37 63 64 65 61 63 36 36
↳ 63 38 34 37 65 37 34 34 33 33 34 34 62 36 65 31 33 33 61 65 65 64 63 62 32 34 61 63 39 63
↳ 61 39 32 64 34 63 66 33 39 65 35 32 33 31 36 66 34 35 63 38 33 64 34 36 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 0e 37 1a c4
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'067cdeac66c847e7443344b6e133aeedcb24ac9ca92d4cf39e52316f45c83d46'"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'8f06975ca7f238f75ac848de5a027758d3ef8374966c14a80bdb8b1520f7f7c495e445c3dcb09b15884cc77'
↳ 8c038e4247392164e49ff3d71526be36787e95cb3'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 38 66 30 36 39 37 35 63
↳ 61 37 66 32 33 38 66 37 35 61 63
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 38 66 30 36 39 37 35 63
↳ 61 37 66 32 33 38 66 37 35 61 63
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 38 34 38 64 65 35 61 30 32 37 37 35 38 64 33 65 66 38 33 37 34 39 36
↳ 36 63 31 34 61 38 30 62 64 62 38 62 31 35 32 30 66 37 66 37 63 34 39 35 65 34 34 35 63 33
↳ 64 63 62 30 39 62 31 35 38 38 34
```

```
comm-server: RX <- (64): 38 34 38 64 65 35 61 30 32 37 37 35 38 64 33 65 66 38 33 37 34 39 36
↳ 36 63 31 34 61 38 30 62 64 62 38 62 31 35 32 30 66 37 66 37 63 34 39 35 65 34 34 35 63 33
↳ 64 63 62 30 39 62 31 35 38 38 34
```

```
comm-client: TX -> (64): 63 63 37 37 38 63 30 33 38 65 34 32 34 37 33 39 32 31 36 34 65 34 39
↳ 66 66 33 64 37 31 35 32 36 62 65 33 36 37 38 37 65 39 35 63 62 33 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d fc 38
```

```
comm-server: RX <- (64): 63 63 37 37 38 63 30 33 38 65 34 32 34 37 33 39 32 31 36 34 65 34 39
↳ 66 66 33 64 37 31 35 32 36 62 65 33 36 37 38 37 65 39 35 63 62 33 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d fc 38
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (4): 09 f0 3a 3e

comm-server: RX <- (4): 09 f0 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 38 66 30 36 39 37 35 63 61 37
↳ 66 32 33 38 66 37 35 61 63 38 34 38 64 65 35 61 30 32 37 37 35 38 64 33 65 66 38 33 37 34
↳ 39 36 36 63 31 34 61 38 30 62 64 62 38 62 31 35 32 30 66 37 66 37 63 34 39 35 65 34 34 35
↳ 63 33 64 63 62 30 39 62 31 35 38 38 34 63 63 37 37 38 63 30 33 38 65 34 32 34 37 33 39 32
↳ 31 36 34 65 34 39 66 66 33 64 37 31 35 32 36 62 65 33 36 37 38 37 65 39 35 63 62 33 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d fc 38 09 f0

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'8f06975ca7f238f75ac848de5a027758d3ef8374966c14a80bdb8b1520f7f7c495e445c3dcb09b15884cc7"
↳ 78c038e4247392164e49ff3d71526be36787e95cb3'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "False"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 66 61 6c 73 65 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 31

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 66 61 6c 73 65 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 31

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (7): 7d a1 48 27 7d 3a 3e

comm-client: RX <- (7): 7d a1 48 27 7d 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (63): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 66 61 6c 73 65 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 31 7d a1 48 27 7d
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "False"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got negative authentication feedback
```

Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).

```
Result (Return Value of authentication method): False (<type 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<type 'bool'>)
```

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of server): False (<type 'bool'>)
```

```
Expectation (Authentication state of server): result = False (<type 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of client): False (<type 'bool'>)
```

```
Expectation (Authentication state of client): result = False (<type 'bool'>)
```

Info Identical secrets set

Info Performing Authentication

```
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (6): 10 4d cd 55 3a 3e

comm-server: RX <- (6): 10 4d cd 55 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'5a11e14e6d5c74fa2396137e409a0d98311e340837c5ef1527217d879a41dd67'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 35 61 31 31 65 31 34 65
↳ 36 64 35 63 37 34 66 61 32 33 39

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 35 61 31 31 65 31 34 65
↳ 36 64 35 63 37 34 66 61 32 33 39

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 36 31 33 37 65 34 30 39 61 30 64 39 38 33 31 31 65 33 34 30 38 33 37
↳ 63 35 65 66 31 35 32 37 32 31 37 64 38 37 39 61 34 31 64 64 36 37 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d d4 ed

```

```
comm-client: RX <- (64): 36 31 33 37 65 34 30 39 61 30 64 39 38 33 31 31 65 33 34 30 38 33 37
↳ 63 35 65 66 31 35 32 37 32 31 37 64 38 37 39 61 34 31 64 64 36 37 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d d4 ed
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (4): 33 50 3a 3e
```

```
comm-client: RX <- (4): 33 50 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 35 61 31 31 65 31 34 65 36 64
↳ 35 63 37 34 66 61 32 33 39 36 31 33 37 65 34 30 39 61 30 64 39 38 33 31 31 65 33 34 30 38
↳ 33 37 63 35 65 66 31 35 32 37 32 31 37 64 38 37 39 61 34 31 64 64 36 37 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d d4 ed 33 50
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'5a11e14e6d5c74fa2396137e409a0d98311e340837c5ef1527217d879a41dd67'"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'e7b6ee4a7e928cce1eaccf2227e2b5227215079842ef1fd7230b585c697fab970f81cf0785afc846d8b201c'
↳ 34c411ebe750d38c8d2b15f7e2bb71bbff2bc13e4'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 65 37 62 36 65 65 34 61
↳ 37 65 39 32 38 63 63 65 31 65 61
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 65 37 62 36 65 65 34 61
↳ 37 65 39 32 38 63 63 65 31 65 61
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 63 63 66 32 32 32 37 65 32 62 35 32 32 37 32 31 35 30 37 39 38 34 32
↳ 65 66 31 66 64 37 32 33 30 62 35 38 35 63 36 39 37 66 61 62 39 37 30 66 38 31 63 66 30 37
↳ 38 35 61 66 63 38 34 36 64 38 62
```

```

comm-server: RX <- (64): 63 63 66 32 32 32 37 65 32 62 35 32 32 37 32 31 35 30 37 39 38 34 32
↳ 65 66 31 66 64 37 32 33 30 62 35 38 35 63 36 39 37 66 61 62 39 37 30 66 38 31 63 66 30 37
↳ 38 35 61 66 63 38 34 36 64 38 62

comm-client: TX -> (64): 32 30 31 63 33 34 63 34 31 31 65 62 65 37 35 30 64 33 38 63 38 64 32
↳ 62 31 35 66 37 65 32 62 62 37 31 62 62 66 66 32 62 63 31 33 65 34 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d 21 c1

comm-server: RX <- (64): 32 30 31 63 33 34 63 34 31 31 65 62 65 37 35 30 64 33 38 63 38 64 32
↳ 62 31 35 66 37 65 32 62 62 37 31 62 62 66 66 32 62 63 31 33 65 34 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d 21 c1

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (4): ac b9 3a 3e
comm-server: RX <- (4): ac b9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 65 37 62 36 65 65 34 61 37 65
↳ 39 32 38 63 63 65 31 65 61 63 63 66 32 32 32 37 65 32 62 35 32 32 37 32 31 35 30 37 39 38
↳ 34 32 65 66 31 66 64 37 32 33 30 62 35 38 35 63 36 39 37 66 61 62 39 37 30 66 38 31 63 66
↳ 30 37 38 35 61 66 63 38 34 36 64 38 62 32 30 31 63 33 34 63 34 31 31 65 62 65 37 35 30 64
↳ 33 38 63 38 64 32 62 31 35 66 37 65 32 62 62 37 31 62 62 66 66 32 62 63 31 33 65 34 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 21 c1 ac b9

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'e7b6ee4a7e928cce1eaccf2227e2b5227215079842ef1fd7230b585c697fab970f81cf0785afc846d8b201"
↳ c34c411ebe750d38c8d2b15f7e2bb71bbff2bc13e4"

prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 11 d3 26 78 3a 3e
comm-client: RX <- (6): 11 d3 26 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback

```

Success Return Value of authentication method is correct (Content True and Type is <type 'bool'>).

```

Result (Return Value of authentication method): True (<type 'bool'>)
Expectation (Return Value of authentication method): result = True (<type 'bool'>)

```

Success Authentication state of server is correct (Content True and Type is <type 'bool'>).

```

Result (Authentication state of server): True (<type 'bool'>)
Expectation (Authentication state of server): result = True (<type 'bool'>)

```

Success Authentication state of client is correct (Content True and Type is <type 'bool'>).

```

Result (Authentication state of client): True (<type 'bool'>)
Expectation (Authentication state of client): result = True (<type 'bool'>)

```

Info Corrupting the authentication mechanism

```

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

```

Info Performing Authentication

```

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 10 4d cd 55 3a 3e
```

```
comm-server: RX <- (6): 10 4d cd 55 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
```

```
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

Success Return Value of authentication method is correct (Content False and Type is <type 'bool'>).

```
Result (Return Value of authentication method): False (<type 'bool'>)
```

```
Expectation (Return Value of authentication method): result = False (<type 'bool'>)
```

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of server): False (<type 'bool'>)
```

```
Expectation (Authentication state of server): result = False (<type 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

```
Result (Authentication state of client): False (<type 'bool'>)
```

```
Expectation (Authentication state of client): result = False (<type 'bool'>)
```

A.1.8 An automatic authentication shall available**Description**

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

Check authentication feedback (client and server) after connect has been triggered.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0

```

prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```

```

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Identical secrets set and automatic authentication

Success Authentication state of server is correct (Content False and Type is <type 'bool'>).

Result (Authentication state of server): False (<type 'bool'>)

Expectation (Authentication state of server): result = False (<type 'bool'>)

Success Authentication state of client is correct (Content False and Type is <type 'bool'>).

Result (Authentication state of client): False (<type 'bool'>)

Expectation (Authentication state of client): result = False (<type 'bool'>)

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"

```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: Connection established...
```

```
comm-server: Cleaning up receive-buffer
```

```
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 10 4d cd 55 3a 3e
comm-server: RX <- (6): 10 4d cd 55 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'d371997e950011742c5a54a0df9c730b2c44e8455a717ccce418941e3ea9c36c'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 64 33 37 31 39 39 37 65
↳ 39 35 30 30 31 31 37 34 32 63 35
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 64 33 37 31 39 39 37 65
↳ 39 35 30 30 31 31 37 34 32 63 35
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 61 35 34 61 30 64 66 39 63 37 33 30 62 32 63 34 34 65 38 34 35 35 61
↳ 37 31 37 63 63 63 65 34 31 38 39 34 31 65 33 65 61 39 63 33 36 63 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 1d 18
comm-client: RX <- (64): 61 35 34 61 30 64 66 39 63 37 33 30 62 32 63 34 34 65 38 34 35 35 61
↳ 37 31 37 63 63 63 65 34 31 38 39 34 31 65 33 65 61 39 63 33 36 63 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 1d 18
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (4): 01 27 3a 3e
comm-client: RX <- (4): 01 27 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 64 33 37 31 39 39 37 65 39 35
↳ 30 30 31 31 37 34 32 63 35 61 35 34 61 30 64 66 39 63 37 33 30 62 32 63 34 34 65 38 34 35
↳ 35 61 37 31 37 63 63 63 65 34 31 38 39 34 31 65 33 65 61 39 63 33 36 63 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 1d 18 01 27
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'd371997e950011742c5a54a0df9c730b2c44e8455a717ccce418941e3ea9c36c"
```

```
prot-client: Executing callback __authenticate_create_key__ to process received data
```

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'165e3227e2694b2966978a66775b7e0407d84cbad6a5b08c21a18b516ee48fb67922f204cff0c457b05b08d'
↳ ccce4a10546b59fc6edc10d19355ea247e462fbb2'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 31 36 35 65 33 32 32 37
↳ 65 32 36 39 34 62 32 39 36 36 39
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 31 36 35 65 33 32 32 37
↳ 65 32 36 39 34 62 32 39 36 36 39
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 37 38 61 36 36 37 37 35 62 37 65 30 34 30 37 64 38 34 63 62 61 64 36
↳ 61 35 62 30 38 63 32 31 61 31 38 62 35 31 36 65 65 34 38 66 62 36 37 39 32 32 66 32 30 34
↳ 63 66 66 30 63 34 35 37 62 30 35
```

```
comm-server: RX <- (64): 37 38 61 36 36 37 37 35 62 37 65 30 34 30 37 64 38 34 63 62 61 64 36
↳ 61 35 62 30 38 63 32 31 61 31 38 62 35 31 36 65 65 34 38 66 62 36 37 39 32 32 66 32 30 34
↳ 63 66 66 30 63 34 35 37 62 30 35
```

```
comm-client: TX -> (64): 62 30 38 64 63 63 63 65 34 61 31 30 35 34 36 62 35 39 66 63 36 65 64
↳ 63 31 30 64 31 39 33 35 35 65 61 32 34 37 65 34 36 32 66 62 62 32 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d 07 66
```

```
comm-server: RX <- (64): 62 30 38 64 63 63 63 65 34 61 31 30 35 34 36 62 35 39 66 63 36 65 64
↳ 63 31 30 64 31 39 33 35 35 65 61 32 34 37 65 34 36 32 66 62 62 32 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d 07 66
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (4): c2 0b 3a 3e

comm-server: RX <- (4): c2 0b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 31 36 35 65 33 32 32 37 65 32
↳ 36 39 34 62 32 39 36 36 39 37 38 61 36 36 37 37 35 62 37 65 30 34 30 37 64 38 34 63 62 61
↳ 64 36 61 35 62 30 38 63 32 31 61 31 38 62 35 31 36 65 65 34 38 66 62 36 37 39 32 32 66 32
↳ 30 34 63 66 66 30 63 34 35 37 62 30 35 62 30 38 64 63 63 63 65 34 61 31 30 35 34 36 62 35
↳ 39 66 63 36 65 64 63 31 30 64 31 39 33 35 35 65 61 32 34 37 65 34 36 32 66 62 62 32 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 07 66 c2 0b

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'165e3227e2694b2966978a66775b7e0407d84cbad6a5b08c21a18b516ee48fb67922f204cff0c457b05b08"
↳ dcccce4a10546b59fc6edc10d19355ea247e462fbb2'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 11 d3 26 78 3a 3e

comm-client: RX <- (6): 11 d3 26 78 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Success Authentication state of server is correct (Content True and Type is <type 'bool'>).

```
Result (Authentication state of server): True (<type 'bool'>)
```

```
Expectation (Authentication state of server): result = True (<type 'bool'>)
```

Success Authentication state of client is correct (Content True and Type is <type 'bool'>).

```
Result (Authentication state of client): True (<type 'bool'>)
```

```
Expectation (Authentication state of client): result = True (<type 'bool'>)
```

A.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.

Description

Communication (rx and tx) shall be disabled, if a secret is given. Except of a response for registered services, saying that a Authentication is required.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

RX and TX is not possible, till a successfull authentication has been performed.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incomming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

prot-server: Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server: Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and ↪ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name ↪ response

```
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

```
prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
```

```
↳ "'msg3_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
```

```
↳ 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 33 5f 64 61
```

```
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
```

```
↳ 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 33 5f 64 61
```

```
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
```

```
↳ 33 36 7d 18 82 9a 08 3a 3e
```

```
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
```

```
↳ 33 36 7d 18 82 9a 08 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
```

```
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
```

```
↳ 63 65 5f 69 64 22 3a 20 33 30 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 33 5f 64 61 74 61
```

```
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
```

```
↳ 20 33 36 7d 18 82 9a 08
```

```
prot-server: RX <- service: execute request, data_id: 36, status: okay, data:
```

```
↳ "u'msg3_data_to_be_transferred'"
```

```
prot-server: Authentication is required. Just sending negative response.
```

```

prot-server: TX -> service: execute response, data_id: 36, status: authentication required,
↳ data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (8): 36 7d 5e 04 41 f5 3a 3e
comm-client: RX <- (8): 36 7d 5e 04 41 f5 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (64): 7b 22 73 74 61 74 75 73 22 3a 20 33 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 33 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 33 36 7d 5e 04 41 f5

prot-client: RX <- service: execute response, data_id: 36, status: authentication required,
↳ data: "None"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 3, u'service_id': 31, u'data': None, u'data_id': 36} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 3, u'service_id': 31, u'data': None, u'data_id': 36} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 3, 'service_id': 31,
↪ 'data': None, 'data_id': 36} (<class 'socket_protocol.data_storage'>)

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪ "'msg2_data_to_be_transferred'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 35 7d 20 18 19 e8 3a 3e

comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 35 7d 20 18 19 e8 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↪ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪ 20 33 35 7d 20 18 19 e8

prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪ "u'msg2_data_to_be_transferred'"

prot-client: Authentification is required. Incomming message will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Identical secrets set

Info Transferring a message client → server

prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↪ data: 'msg1_data_to_be_transferred' will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↪ in buffer.

Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): False (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = False (<type 'bool'>)

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on server side): None (<type 'NoneType'>)

Expectation (Received message on server side): result = None (<type 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentification is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): False (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 10 4d cd 55 3a 3e

comm-server: RX <- (6): 10 4d cd 55 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 10 4d cd 55

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↪ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

```

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'6116ec2c14dd6fcb3a92c0fb2f4a9cc79da9d30d05d8230d3a27c39aaa658805'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 31 31 36 65 63 32 63
↳ 31 34 64 64 36 66 63 62 33 61 39

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 36 31 31 36 65 63 32 63
↳ 31 34 64 64 36 66 63 62 33 61 39

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 32 63 30 66 62 32 66 34 61 39 63 63 37 39 64 61 39 64 33 30 64 30 35
↳ 64 38 32 33 30 64 33 61 32 37 63 33 39 61 61 61 36 35 38 38 30 35 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 5a 75

comm-client: RX <- (64): 32 63 30 66 62 32 66 34 61 39 63 63 37 39 64 61 39 64 33 30 64 30 35
↳ 64 38 32 33 30 64 33 61 32 37 63 33 39 61 61 61 36 35 38 38 30 35 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 5a 75

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (4): 7a d9 3a 3e
comm-client: RX <- (4): 7a d9 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 36 31 31 36 65 63 32 63 31 34
↳ 64 64 36 66 63 62 33 61 39 32 63 30 66 62 32 66 34 61 39 63 63 37 39 64 61 39 64 33 30 64
↳ 30 35 64 38 32 33 30 64 33 61 32 37 63 33 39 61 61 61 36 35 38 38 30 35 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 5a 75 7a d9

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'6116ec2c14dd6fcb3a92c0fb2f4a9cc79da9d30d05d8230d3a27c39aaa658805'"

prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'7ff3e52b9d2d8b22113e1cf68b71f1cba2fbcc01c0095db59702fb60f7a2811dcc6b6b6961fc14ed8d89f5e8j'
↳ 8a709dc00d7e08fc7aaf186d9cdf0c5ff5cb327c1'"

```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 37 66 66 33 65 35 32 62
↪ 39 64 32 64 38 62 32 32 31 31 33
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 37 66 66 33 65 35 32 62
↪ 39 64 32 64 38 62 32 32 31 31 33
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 65 31 63 66 36 38 62 37 31 66 31 63 62 61 32 66 62 63 63 30 31 63 30
↪ 30 39 35 64 62 35 39 37 30 32 66 62 36 30 66 37 61 32 38 31 31 64 63 63 62 36 62 36 39 36
↪ 31 66 63 31 34 65 64 38 64 38 39
```

```
comm-server: RX <- (64): 65 31 63 66 36 38 62 37 31 66 31 63 62 61 32 66 62 63 63 30 31 63 30
↪ 30 39 35 64 62 35 39 37 30 32 66 62 36 30 66 37 61 32 38 31 31 64 63 63 62 36 62 36 39 36
↪ 31 66 63 31 34 65 64 38 64 38 39
```

```
comm-client: TX -> (64): 66 35 65 38 38 61 37 30 39 64 63 30 30 64 37 65 30 38 66 63 37 61 61
↪ 66 31 38 36 64 39 63 64 66 30 63 35 66 66 35 63 62 33 32 37 63 31 22 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 3d 20 31 7d d2 93
```

```
comm-server: RX <- (64): 66 35 65 38 38 61 37 30 39 64 63 30 30 64 37 65 30 38 66 63 37 61 61
↪ 66 31 38 36 64 39 63 64 66 30 63 35 66 66 35 63 62 33 32 37 63 31 22 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 3d 20 31 7d d2 93
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (4): 4e 28 3a 3e
```

```
comm-server: RX <- (4): 4e 28 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 37 66 66 33 65 35 32 62 39 64
↪ 32 64 38 62 32 32 31 31 33 65 31 63 66 36 38 62 37 31 66 31 63 62 61 32 66 62 63 63 30 31
↪ 63 30 30 39 35 64 62 35 39 37 30 32 66 62 36 30 66 37 61 32 38 31 31 64 63 63 62 36 62 36
↪ 39 36 31 66 63 31 34 65 64 38 64 38 39 66 35 65 38 38 61 37 30 39 64 63 30 30 64 37 65 30
↪ 38 66 63 37 61 61 66 31 38 36 64 39 63 64 66 30 63 35 66 66 35 63 62 33 32 37 63 31 22 2c
↪ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d d2 93 4e 28
```

```

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'7ff3e52b9d2d8b22113e1cf68b71f1cba2fbcc01c0095db59702fb60f7a2811dcc6b6961fc14ed8d89f5e
↳ 88a709dc00d7e08fc7aaf186d9cdf0c5ff5cb327c1'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 11 d3 26 78 3a 3e

comm-client: RX <- (6): 11 d3 26 78 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78

prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"

prot-client: Executing callback __authenticate_process_feedback__ to process received data

prot-client: Got positive authentication feedback

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"

```

```

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 34 7d 7a 6c e4 9b 3a 3e

comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 34 7d 7a 6c e4 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↪ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↪ 20 33 34 7d 7a 6c e4 9b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪ "u'msg1_data_to_be_transferred"

prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':

↪ u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

```
Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 35 7d 20 18 19 e8 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 35 7d 20 18 19 e8
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'u'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {u'status': 4, u'service_id': 17, u'data':
↪ u'msg2_data_to_be_transferred', u'data_id': 35} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
↪ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
↪ 'socket_protocol.data_storage'>)

A.1.10 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts

Description

It shall be possible to add a specific message, identified by Service-ID and Data-ID, to a whitelist. All messages added to that whitelist shall be transmitted and received, if no authentication was successfull performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentication mechanism.

Fitcriterion

Transmiton and Reception will be enabled, after the message has been added to the whitelist.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incomming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↪ authentication whitelist

```

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

```

```
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
```

```
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Identical secrets set

Info Transferring a message client → server

```
prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
↳ data: 'msg1_data_to_be_transferred' will be ignored.
```

```
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content False and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): False (<type 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = False (<type 'bool'>)
```

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

```
Result (Received message on server side): None (<type 'NoneType'>)
```

```
Expectation (Received message on server side): result = None (<type 'NoneType'>)
```

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

```
Result (Returnvalue of Server send Method): False (<type 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

```
Result (Received message on client side): None (<type 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<type 'NoneType'>)
```

Info Added msg1 to client whitelist (sid=17, did=34)

```
prot-client: Adding Message (service: 17, data_id: 34) to the authentication whitelist
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Authentication is required. Incomming message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on server side): None (<type 'NoneType'>)

Expectation (Received message on server side): result = None (<type 'NoneType'>)

Info Transferring a message server → client

```

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

```

```

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

```

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): False (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Added msg1 to server whitelist (sid=17, did=34)

prot-server: Adding Message (service: 17, data_id: 34) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↪ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↪ 33 34 7d 7a 6c e4 9b 3a 3e

comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↪ 33 34 7d 7a 6c e4 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"

prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':
↳ u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

Success Returnvalue of Server send Method is correct (Content False and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): False (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<type 'bool'>)

Success Received message on client side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on client side): None (<type 'NoneType'>)

Expectation (Received message on client side): result = None (<type 'NoneType'>)

Info Added msg2 to client and server whitelist (sid=17, did=35)

prot-client: Adding Message (service: 17, data_id: 35) to the authentication whitelist

prot-server: Adding Message (service: 17, data_id: 35) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↪ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↪ 33 34 7d 7a 6c e4 9b 3a 3e

comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20

↪ 33 34 7d 7a 6c e4 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69

↪ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61

↪ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a

↪ 20 33 34 7d 7a 6c e4 9b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:

↪ "u'msg1_data_to_be_transferred'"

prot-server: Message data is stored in buffer and is now ready to be retrieved by receive

↪ method

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content {u'status': 0, u'service_id': 17, u'data': u'msg1_data_to_be_transferred', u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {u'status': 0, u'service_id': 17, u'data':
↪ u'msg1_data_to_be_transferred', u'data_id': 34} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↪ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↪ 'socket_protocol.data_storage'>)

Info Transferring a message server → client

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪ "'msg2_data_to_be_transferred'"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-server: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 35 7d 20 18 19 e8 3a 3e

comm-client: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↪ 33 35 7d 20 18 19 e8 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 35 7d 20 18 19 e8
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "u'msg2_data_to_be_transferred"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content {u'status': 4, u'service_id': 17, u'data': u'msg2_data_to_be_transferred', u'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {u'status': 4, u'service_id': 17, u'data':

↳ u'msg2_data_to_be_transferred', u'data_id': 35} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,

↳ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class

↳ 'socket_protocol.data_storage'>)

A.1.11 Define a channel name for the server and client after connection is established

Description

After the connection is established, the client will initiate the channel name exchange. The channel name defined on the client side will be dominant.

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

Perform a channel name exchange with no channel name definition, differing channel name definition and identical channel name definition. In all cases, the channel name of the client will be used. Perform two channel name exchanges with only one channel name definition. This definition will be used.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

```

comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

```

```

prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Setting no Channel name for server and client

Success Channel name of server is correct (Content None and Type is <type 'NoneType'>).

```
Result (Channel name of server): None (<type 'NoneType'>)
```

```
Expectation (Channel name of server): result = None (<type 'NoneType'>)
```

Success Channel name of client is correct (Content None and Type is <type 'NoneType'>).

```
Result (Channel name of client): None (<type 'NoneType'>)
```

```
Expectation (Channel name of client): result = None (<type 'NoneType'>)
```

Info Setting different Channel names for client and Server

Info Connecting Server and Client

```
comm-client: Connection Lost...
```

```
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
comm-server: Connection Lost...
```

```
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
comm-client: Connection established...
```

```
comm-client: Cleaning up receive-buffer
```

```
prot-client: Cleaning up receive-buffer
```

```
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
```

```
comm-server: Connection established...
```

```
comm-server: Cleaning up receive-buffer
```

```
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
comm-client: TX -> (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
comm-server: RX <- (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 22 63 6c 69 65 6e 74 22 2c 20 22
↳ 64 61 74 61 5f 69 64 22 3a 20 30 7d 93 56 e3 b4
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "u'client'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: overwriting user defined channel name from 'server' to u'client'
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <type 'str'>).

```

Result (Channel name of server): 'client' (<type 'str'>)
Expectation (Channel name of server): result = 'client' (<type 'str'>)

```

Success Channel name of client is correct (Content 'client' and Type is <type 'str'>).

```

Result (Channel name of client): 'client' (<type 'str'>)
Expectation (Channel name of client): result = 'client' (<type 'str'>)

```

Info Setting identical Channel names for client and server

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 75 6e 69 74 74 65 73 74
↳ 22 2c 20 22 64 61 74 61 5f 69 64
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 75 6e 69 74 74 65 73 74
↳ 22 2c 20 22 64 61 74 61 5f 69 64
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

```

```

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (12): 22 3a 3d 20 30 7d b0 bd 92 06 3a 3e

comm-server: RX <- (12): 22 3a 3d 20 30 7d b0 bd 92 06 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (68): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 22 75 6e 69 74 74 65 73 74 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 30 7d b0 bd 92 06

prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "u'unittest'"

prot-server: Executing callback __channel_name_request__ to process received data

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Channel name of server is correct (Content 'unittest' and Type is <type 'str'>).

```
Result (Channel name of server): 'unittest' (<type 'str'>)
Expectation (Channel name of server): result = 'unittest' (<type 'str'>)
```

Success Channel name of client is correct (Content 'unittest' and Type is <type 'str'>).

```
Result (Channel name of client): 'unittest' (<type 'str'>)
Expectation (Channel name of client): result = 'unittest' (<type 'str'>)
```

Info Setting Channel name for client only

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c
↪ 20 22 64 61 74 61 5f 69 64 22 3a
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 22 63 6c 69 65 6e 74 22 2c
↪ 20 22 64 61 74 61 5f 69 64 22 3a
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
comm-client: TX -> (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
```

```
comm-server: RX <- (10): 3d 20 30 7d 93 56 e3 b4 3a 3e
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (66): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 22 63 6c 69 65 6e 74 22 2c 20 22
↪ 64 61 74 61 5f 69 64 22 3a 20 30 7d 93 56 e3 b4
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↪ "u'client'"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: channel name is now 'client'
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <type 'str'>).

```
Result (Channel name of server): 'client' (<type 'str'>)
```

```
Expectation (Channel name of server): result = 'client' (<type 'str'>)
```

Success Channel name of client is correct (Content 'client' and Type is <type 'str'>).

```
Result (Channel name of client): 'client' (<type 'str'>)
```

```
Expectation (Channel name of client): result = 'client' (<type 'str'>)
```

Info Setting Channel name for server only

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer

```

```

prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data:
↪ "'server'"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 22 73 65 72 76 65 72 22 2c
↪ 20 22 64 61 74 61 5f 69 64 22 3a
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 22 73 65 72 76 65 72 22 2c
↪ 20 22 64 61 74 61 5f 69 64 22 3a

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
comm-server: TX -> (10): 3d 20 30 7d 9c 48 3b b3 3a 3e
comm-client: RX <- (10): 3d 20 30 7d 9c 48 3b b3 3a 3e
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 22 73 65 72 76 65 72 22 2c 20 22
↳ 64 61 74 61 5f 69 64 22 3a 20 30 7d 9c 48 3b b3
prot-client: RX <- service: channel name response, data_id: name, status: okay, data:
↳ "u'server'"
prot-client: Executing callback __channel_name_response__ to process received data
prot-client: channel name is now 'server'

```

Success Channel name of server is correct (Content 'server' and Type is <type 'str'>).

Result (Channel name of server): 'server' (<type 'str'>)

Expectation (Channel name of server): result = 'server' (<type 'str'>)

Success Channel name of client is correct (Content 'server' and Type is <type 'str'>).

Result (Channel name of client): 'server' (<type 'str'>)

Expectation (Channel name of client): result = 'server' (<type 'str'>)

A.1.12 The User shall be able to define a new service

Description

The service is defined by a Request Service-ID and a Response Service-ID.

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

Define a service and check, that the server will respond on the new Service-ID. The Status shall be "Request has no callback. Data buffered.", because no callback is registered for that request.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.

```

prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 14 5b 30 5c 3a 3e

comm-client: RX <- (6): 14 5b 30 5c 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"

prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e

comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 18; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Client send Method): True (<type 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)

Success Received message on server side is correct (Content None and Type is <type 'NoneType'>).

Result (Received message on server side): None (<type 'NoneType'>)

Expectation (Received message on server side): result = None (<type 'NoneType'>)

Info Adding service to server instance for the transmit message

prot-server: Adding Service with Request=17 and Response=18

Info Transferring a message client → server → client

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↳ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69

↳ 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73 67 31 5f 64 61

↳ 74 61 5f 74 6f 5f 62 65 5f 74 72

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↳ STP_STATE_STORE_DATA

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
comm-server: RX <- (32): 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a 3d 20
↳ 33 34 7d 7a 6c e4 9b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 64 61 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61
↳ 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 2c 20 22 64 61 74 61 5f 69 64 22 3a
↳ 20 33 34 7d 7a 6c e4 9b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred"
prot-server: Incoming message with no registered callback. Sending negative response.
prot-server: TX -> service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↳ 61 74 61 5f 69 64 22 3a 3d 20 33
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (8): 34 7d e8 ee d8 5c 3a 3e
comm-client: RX <- (8): 34 7d e8 ee d8 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (64): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 33 34 7d e8 ee d8 5c
prot-client: RX <- service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)
```

Success Received message on server side is correct (Content {u'status': 1, u'service_id': 18, u'data': None, u'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {u'status': 1, u'service_id': 18, u'data': None,
↳ u'data_id': 34} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'status': 1, 'service_id': 18,
↳ 'data': None, 'data_id': 34} (<class 'socket_protocol.data_storage'>)
```

A.1.13 Registration of already registered request Service-ID or response Service-ID shall not be possible

Description

An exception shall be raised, if a service registration with an existing request SID or response SID is performed.

Reason for the implementation

Changing existing services will create strange situations with already registered callbacks.

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

This test was passed with the state: **Success.**

Info Setting up communication

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server: Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and ↪ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist

```

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Adding a service with an already registered request SID

```
prot-server: Service with Request-SID=10 and Response-SID=18 not added, because request SID
↳ is already registered
```

Success Expected Exception RequestSidExistsError was triggered

Info Adding a service with an already registered response SID

```
prot-server: Service with Request-SID=17 and Response-SID=11 not added, because response SID
↳ is already registered
```

Success Expected Exception ResponseSidExistsError was triggered

A.1.14 It shall be possible to register a callback for a specific Service- and Data-ID

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
```

```

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response

```

```
prot-client: Adding Service with Request=execute request and Response=execute response
```

```
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
```

```
comm-client: Cleaning up receive-buffer
```

```
prot-client: Cleaning up receive-buffer
```

```
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
```

```
comm-server: Connection established...
```

```
comm-server: Cleaning up receive-buffer
```

```
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6

```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Info Overwriting existing Callback using one with faulty (too many) return values

```

prot-server: Overwriting existing callback '__callback__' for service_id (10) and data_id (0)
↳ to '__callback_error__'!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↪ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback_error__ to process received data
```

```
prot-server: Exception raised. Check callback __callback_error__: "too many values to unpack"
↪ and it's return values for service_id 10 and data_id 0
```

```
prot-server: TX -> service: read data response, data_id: 0, status: callback error, data:
↪ "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↪ 61 74 61 5f 69 64 22 3a 3d 20 30
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↪ 61 74 61 5f 69 64 22 3a 3d 20 30
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d 3f 8f 7d 86 3a 3e
comm-client: RX <- (7): 7d 3f 8f 7d 86 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 73 74 61 74 75 73 22 3a 20 32 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 30 7d 3f 8f 7d 86
prot-client: RX <- service: read data response, data_id: 0, status: callback error, data:
↳ "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {u'status': 2, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {u'status': 2, u'service_id': 11, u'data': None,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'status': 2, 'service_id': 11, 'data':
↳ None, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Info Removing the registered Callback

```

prot-server: Deleting existing callback '__callback_error__' for service_id (10) and data_id
↳ (0)!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↪ 69 64 22 3a 20 30 7d e6 17 fc 16
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Incoming message with no registered callback. Sending negative response.
prot-server: TX -> service: read data response, data_id: 0, status: no callback for service,
↪ data buffered, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↪ 61 74 61 5f 69 64 22 3a 3d 20 30
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64
↪ 61 74 61 5f 69 64 22 3a 3d 20 30

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d 79 5d 48 e2 3a 3e
comm-client: RX <- (7): 7d 79 5d 48 e2 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 20 30 7d 79 5d 48 e2
prot-client: RX <- service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content None and Type is <type 'NoneType'>).

Result (Message stored inside callback): None (<type 'NoneType'>)

Expectation (Message stored inside callback): result = None (<type 'NoneType'>)

Success Message received by client is correct (Content {u'status': 1, u'service_id': 11, u'data': None, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {u'status': 1, u'service_id': 11, u'data': None,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'status': 1, 'service_id': 11, 'data':
↳ None, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

A.1.15 It shall be possible to register a callback for a specific Service-ID and all Data-IDs

Testresult

This test was passed with the state: **Success.**

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↳ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.
prot-client:	Cleaning up receive-buffer
prot-client:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-client:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist

```

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=None
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68

prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service.id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↪ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0}
and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↪ u'data_id': 0} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↪ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

A.1.16 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↪ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↪ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0

prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0

```

prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```

```

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```

prot-server: Adding callback '__callback__' for SID=None and DID=0

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (5): 17 fc 16 3a 3e

comm-server: RX <- (5): 17 fc 16 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Executing callback __callback__ to process received data

prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (5): 02 24 68 3a 3e

comm-client: RX <- (5): 02 24 68 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

A.1.17 It shall be possible to register a callback for all incoming messages

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
```

```
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
```

```

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response

```

prot-client: Initialisation finished.

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (6): 28 3b d3 54 3a 3e

comm-server: RX <- (6): 28 3b d3 54 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
```

```
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=None
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↪ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 17 fc 16 3a 3e
comm-server: RX <- (5): 17 fc 16 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 02 24 68 3a 3e
comm-client: RX <- (5): 02 24 68 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)

```

A.1.18 Callback choice, if several callbacks are available (caused by wildcard callbacks)

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response

```

```

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

```

```
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering all kind of Callbacks

```

prot-server: Adding callback '__callback3__' for SID=None and DID=None

```

```
prot-server: Adding callback '__callback2__' for SID=None and DID=0
```

```
prot-server: Adding callback '__callback1__' for SID=10 and DID=None
```

```
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 33 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 60
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 02 24 68 3a 3e
```

```
comm-client: RX <- (5): 02 24 68 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 33 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 60 02 24 68
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 33, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 33,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 33, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data- and Service-ID

prot-server: Deleting existing callback '__callback__' for service_id (10) and data_id (0)!

Info Transferring data

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
 ↪ 61 5f 69 64 22 3a 3d 20 30 7d e6

comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
 ↪ 61 5f 69 64 22 3a 3d 20 30 7d e6

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (5): 17 fc 16 3a 3e

comm-server: RX <- (5): 17 fc 16 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
 ↪ 69 64 22 3a 20 30 7d e6 17 fc 16

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Executing callback __callback1__ to process received data

prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
 ↪ data: "34"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69
 ↪ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 34 2c 20 22 64 61 74
 ↪ 61 5f 69 64 22 3a 3d 20 30 7d 46

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 34 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 46
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 3f 83 36 3a 3e
```

```
comm-client: RX <- (5): 3f 83 36 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 34 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 46 3f 83 36
```

```
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↳ data: "34"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 6, u'service_id': 11, u'data': 34, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 6, u'service_id': 11, u'data': 34,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 6, 'service_id': 11, 'data':
↳ 34, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Service-ID and all Data-IDs

```
prot-server: Deleting existing callback '__callback1__' for service_id (10) and data_id
↳ (None)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback2__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 35 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e8
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 35 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 57 12 a7 3a 3e
```

```
comm-client: RX <- (5): 57 12 a7 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 35 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e8 57 12 a7
```

```
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 6, u'service_id': 11, u'data': 35, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 6, u'service_id': 11, u'data': 35,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 6, 'service_id': 11, 'data':
↳ 35, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data-ID and all Serice-IDs

```
prot-server: Deleting existing callback '__callback2__' for service_id (None) and data_id (0)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 64 61 74 61 22 3a 3d 20 33 31 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d e6
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 17 fc 16 3a 3e
```

```
comm-server: RX <- (5): 17 fc 16 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 64 61 74 61 22 3a 20 33 31 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d e6 17 fc 16
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback3__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "36"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 36 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 1a
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 64 61 74 61 22 3a 3d 20 33 36 2c 20 22 64 61 74
↳ 61 5f 69 64 22 3a 3d 20 30 7d 1a
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 5b f9 7e 3a 3e
```

```
comm-client: RX <- (5): 5b f9 7e 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 64 61 74 61 22 3a 20 33 36 2c 20 22 64 61 74 61 5f
↳ 69 64 22 3a 20 30 7d 1a 5b f9 7e
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "36"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {u'status': 0, u'service_id': 10, u'data': 31, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {u'status': 0, u'service_id': 10, u'data': 31,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'status': 0, 'service_id': 10,
↳ 'data': 31, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {u'status': 0, u'service_id': 11, u'data': 36, u'data_id': 0} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {u'status': 0, u'service_id': 11, u'data': 36,
↳ u'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'status': 0, 'service_id': 11, 'data':
↳ 36, 'data_id': 0} (<class 'socket_protocol.data_storage'>)
```

A.1.19 Connection established information

Testresult

This test was passed with the state: **Success.**

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incomming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↳ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.
prot-client:	Cleaning up receive-buffer
prot-client:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-client:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist

```

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client connection status): True (<type 'bool'>)
Expectation (Client connection status): result = True (<type 'bool'>)
```

Success Server connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server connection status): True (<type 'bool'>)
Expectation (Server connection status): result = True (<type 'bool'>)
```

Success Client connection status is correct (Content False and Type is <type 'bool'>).

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
Result (Client connection status): False (<type 'bool'>)
Expectation (Client connection status): result = False (<type 'bool'>)
```

Success Server connection status is correct (Content False and Type is <type 'bool'>).

```
Result (Server connection status): False (<type 'bool'>)
Expectation (Server connection status): result = False (<type 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 28 3b d3 54 3a 3e
```

```
comm-server: RX <- (6): 28 3b d3 54 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↪ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Client connection status): True (<type 'bool'>)
Expectation (Client connection status): result = True (<type 'bool'>)

```

Success Server connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Server connection status): True (<type 'bool'>)
Expectation (Server connection status): result = True (<type 'bool'>)

```

Info Adding secrets to socket_protocol

Success Client connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Client connection status): False (<type 'bool'>)
Expectation (Client connection status): result = False (<type 'bool'>)

```

Success Server connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Server connection status): False (<type 'bool'>)
Expectation (Server connection status): result = False (<type 'bool'>)

```

Info Doing authentication

```

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"

```

```
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 10 4d cd 55 3a 3e
```

```
comm-server: RX <- (6): 10 4d cd 55 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 10 4d cd 55
```

```
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'9b3dea97505fe65d3f9dc6b72cf6795c35751a1368e2ae512996d6a5e30e65fa'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 39 62 33 64 65 61 39 37
↳ 35 30 35 66 65 36 35 64 33 66 39
```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 22 39 62 33 64 65 61 39 37
↳ 35 30 35 66 65 36 35 64 33 66 39
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 64 63 36 62 37 32 63 66 36 37 39 35 63 33 35 37 35 31 61 31 33 36 38
↳ 65 32 61 65 35 31 32 39 39 36 64 36 61 35 65 33 30 65 36 35 66 61 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 57 51
comm-client: RX <- (64): 64 63 36 62 37 32 63 66 36 37 39 35 63 33 35 37 35 31 61 31 33 36 38
↳ 65 32 61 65 35 31 32 39 39 36 64 36 61 35 65 33 30 65 36 35 66 61 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 30 7d 57 51
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (4): 03 9e 3a 3e
comm-client: RX <- (4): 03 9e 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (124): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 22 39 62 33 64 65 61 39 37 35 30
↳ 35 66 65 36 35 64 33 66 39 64 63 36 62 37 32 63 66 36 37 39 35 63 33 35 37 35 31 61 31 33
↳ 36 38 65 32 61 65 35 31 32 39 39 36 64 36 61 35 65 33 30 65 36 35 66 61 22 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 20 30 7d 57 51 03 9e
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "u'9b3dea97505fe65d3f9dc6b72cf6795c35751a1368e2ae512996d6a5e30e65fa'"
prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'a86faa64ceab72c286e698766391348ec73969ec0fe70cd5e21991ef6e9b14fe62d69854cb03f460404e074'
↳ c63f7722e06e798cb4e1cccb12654ebbc4bf95ff5'"
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 61 38 36 66 61 61 36 34
↳ 63 65 61 62 37 32 63 32 38 36 65
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 61 38 36 66 61 61 36 34
↳ 63 65 61 62 37 32 63 32 38 36 65
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (64): 36 39 38 37 36 36 33 39 31 33 34 38 65 63 37 33 39 36 39 65 63 30 66
↳ 65 37 30 63 64 35 65 32 31 39 39 31 65 66 36 65 39 62 31 34 66 65 36 32 64 36 39 38 35 34
↳ 63 62 30 33 66 34 36 30 34 30 34

comm-server: RX <- (64): 36 39 38 37 36 36 33 39 31 33 34 38 65 63 37 33 39 36 39 65 63 30 66
↳ 65 37 30 63 64 35 65 32 31 39 39 31 65 66 36 65 39 62 31 34 66 65 36 32 64 36 39 38 35 34
↳ 63 62 30 33 66 34 36 30 34 30 34

comm-client: TX -> (64): 65 30 37 34 63 36 33 66 37 37 32 32 65 30 36 65 37 39 38 63 62 34 65
↳ 31 63 63 63 62 31 32 36 35 34 65 62 62 63 34 62 66 39 35 66 66 35 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d f1 96

comm-server: RX <- (64): 65 30 37 34 63 36 33 66 37 37 32 32 65 30 36 65 37 39 38 63 62 34 65
↳ 31 63 63 63 62 31 32 36 35 34 65 62 62 63 34 62 66 39 35 66 66 35 22 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 3d 20 31 7d f1 96

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (4): c7 98 3a 3e

comm-server: RX <- (4): c7 98 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (188): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 30 2c 20 22 64 61 74 61 22 3a 20 22 61 38 36 66 61 61 36 34 63 65
↳ 61 62 37 32 63 32 38 36 65 36 39 38 37 36 36 33 39 31 33 34 38 65 63 37 33 39 36 39 65 63
↳ 30 66 65 37 30 63 64 35 65 32 31 39 39 31 65 66 36 65 39 62 31 34 66 65 36 32 64 36 39 38
↳ 35 34 63 62 30 33 66 34 36 30 34 30 34 65 30 37 34 63 36 33 66 37 37 32 32 65 30 36 65 37
↳ 39 38 63 62 34 65 31 63 63 63 62 31 32 36 35 34 65 62 62 63 34 62 66 39 35 66 66 35 22 2c
↳ 20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d f1 96 c7 98

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "u'a86faa64ceab72c286e698766391348ec73969ec0fe70cd5e21991ef6e9b14fe62d69854cb03f460404e07_
↳ 4c63f7722e06e798cb4e1cccb12654ebbc4bf95ff5'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d

```

```
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 64 61 74 61 22 3a 3d 20 74 72 75 65 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 31 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 11 d3 26 78 3a 3e
```

```
comm-client: RX <- (6): 11 d3 26 78 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 31 2c 20 22 64 61 74 61 22 3a 20 74 72 75 65 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 31 7d 11 d3 26 78
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Success Client connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client connection status): True (<type 'bool'>)
```

```
Expectation (Client connection status): result = True (<type 'bool'>)
```

Success Server connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server connection status): True (<type 'bool'>)
```

```
Expectation (Server connection status): result = True (<type 'bool'>)
```

A.1.20 Is connected information

Testresult

This test was passed with the state: **Success.**

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incomming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↳ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↳ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.
prot-client:	Cleaning up receive-buffer
prot-client:	Adding Service with Request=authentication request and ↳ Response=authentication response
prot-client:	Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist

```

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Client Communication instance connection status): True (<type 'bool'>)
Expectation (Client Communication instance connection status): result = True (<type 'bool'>)

```

Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```

Result (Server Communication instance connection status): True (<type 'bool'>)
Expectation (Server Communication instance connection status): result = True (<type 'bool'>)

```

Info Disconnecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

```

Success Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Client Communication instance connection status): False (<type 'bool'>)
Expectation (Client Communication instance connection status): result = False (<type 'bool'>)

```

Success Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```

Result (Server Communication instance connection status): False (<type 'bool'>)
Expectation (Server Communication instance connection status): result = False (<type 'bool'>)

```

A.1.21 Reconnect Method

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer

```

```

comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

```

```

prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client Communication instance connection status): True (<type 'bool'>)
Expectation (Client Communication instance connection status): result = True (<type 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server Communication instance connection status): True (<type 'bool'>)
Expectation (Server Communication instance connection status): result = True (<type 'bool'>)
```

Info Disconnecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Success Client Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```
Result (Client Communication instance connection status): False (<type 'bool'>)
Expectation (Client Communication instance connection status): result = False (<type 'bool'>)
```

Success Server Communication instance connection status is correct (Content False and Type is <type 'bool'>).

```
Result (Server Communication instance connection status): False (<type 'bool'>)
Expectation (Server Communication instance connection status): result = False (<type 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-server: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↪ 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↪ 74 61 5f 69 64 22 3a 3d 20 30 7d
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 28 3b d3 54 3a 3e
comm-server: RX <- (6): 28 3b d3 54 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 38 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 28 3b d3 54
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
comm-client: RX <- (64): 3a 3c 7b 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 64 61 74 61 22 3a 3d 20 6e 75 6c 6c 2c 20 22 64 61
↳ 74 61 5f 69 64 22 3a 3d 20 30 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 14 5b 30 5c 3a 3e
comm-client: RX <- (6): 14 5b 30 5c 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 73 65 72 76 69
↳ 63 65 5f 69 64 22 3a 20 39 2c 20 22 64 61 74 61 22 3a 20 6e 75 6c 6c 2c 20 22 64 61 74 61
↳ 5f 69 64 22 3a 20 30 7d 14 5b 30 5c
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Client Communication instance connection status): True (<type 'bool'>)
Expectation (Client Communication instance connection status): result = True (<type 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <type 'bool'>).

```
Result (Server Communication instance connection status): True (<type 'bool'>)
Expectation (Server Communication instance connection status): result = True (<type 'bool'>)
```

A.1.22 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
```

```

comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

```

```

prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
comm-server: RX <- (21): 3a 3c 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (17): 00 00 00 00 00 00 00 08 00 00 00 00 6e 75 6c 6c 13
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (21): 3a 3c 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12 3a 3e
comm-client: RX <- (21): 3a 3c 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (17): 00 00 00 00 00 00 00 09 00 00 00 00 6e 75 6c 6c 12
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (45): 3a 3c 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
comm-server: RX <- (45): 3a 3c 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (41): 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "u'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <type 'bool'>).

```
Result (Returnvalue of Client send Method): True (<type 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<type 'bool'>)
```

Success Received message on server side is correct (Content {'status': 0, 'service_id': 17, 'data': u'msg1_data_to_be_transferred', 'data_id': 34} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'status': 0, 'service_id': 17, 'data':
↳ u'msg1_data_to_be_transferred', 'data_id': 34} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'status': 0, 'service_id': 17,
↳ 'data': 'msg1_data_to_be_transferred', 'data_id': 34} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```

comm-server: TX -> (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e

comm-client: RX <- (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (41): 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b

prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪ "u'msg2_data_to_be_transferred"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

```

Success Returnvalue of Server send Method is correct (Content True and Type is <type 'bool'>).

Result (Returnvalue of Server send Method): True (<type 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<type 'bool'>)

Success Received message on client side is correct (Content {'status': 4, 'service_id': 17, 'data': u'msg2_data_to_be_transferred', 'data_id': 35} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'status': 4, 'service_id': 17, 'data':
↪ u'msg2_data_to_be_transferred', 'data_id': 35} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'status': 4, 'service_id': 17,
↪ 'data': 'msg2_data_to_be_transferred', 'data_id': 35} (<class
↪ 'socket_protocol.data_storage'>)

B Trace for testrun with python 3.8.5 (final)

B.1 Tests with status Info (22)

B.1.1 Status

Description

The Status shall hold some general information (in most cases it is used by the responder). Examples: Okay, Service or Data unknown, Operation not permitted, Authentification required, ...

Reason for the implementation

Give the possibility to transfer additional status information (e.g. to explain negative responses).

Fitcriterion

A Status is part of the Message Object and it is holding the Status information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'data': None, 'data_id': None, 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success status is part of the message object is correct ('status' is in the list or dict).

Result (status is part of the message object): {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (status is part of the message object): 'status' in result

Success Content in message object for status is correct (Content 'S' and Type is <class 'str'>).

Result (Content in message object for status): 'S' (<class 'str'>)

Expectation (Content in message object for status): result = 'S' (<class 'str'>)

B.1.2 Service-ID**Description**

The Service-ID shall hold information about the type of the request / corresponding response. Examples: read request, write request, read response, write response, ...

Reason for the implementation

Give the requestor the possibility to use different types (Services) for a transfer.

Fitcriterion

A Service-ID is part of the Message Object and it is holding the Service-ID information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}

Success service_id is part of the message object is correct ('service_id' is in the list or dict).

Result (service_id is part of the message object): {'data': None, 'data_id': None,
↪ 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)

Expectation (service_id is part of the message object): 'service_id' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success service_id is part of the message object is correct ('service_id' is in the list or dict).

Result (service_id is part of the message object): {'data': 'D', 'data_id': 'DID',
↪ 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (service_id is part of the message object): 'service_id' in result

Success Content in message object for service_id is correct (Content 'SID' and Type is <class 'str'>).

Result (Content in message object for service_id): 'SID' (<class 'str'>)

Expectation (Content in message object for service_id): result = 'SID' (<class 'str'>)

B.1.3 Data-ID**Description**

The Data-ID shall hold information to differtiate the data for a specific Service.

Reason for the implementation

Give the possibility to transfer different information for each Service.

Fitcriterion

A Data-ID is part of the Message Object and it is holding the Data-ID information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}

Success data_id is part of the message object is correct ('data_id' is in the list or dict).

Result (data_id is part of the message object): {'data': None, 'data_id': None, 'service_id':
↪ None, 'status': None} (<class 'socket_protocol.data_storage'>)

Expectation (data_id is part of the message object): 'data_id' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success data_id is part of the message object is correct ('data_id' is in the list or dict).

Result (data_id is part of the message object): {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (data_id is part of the message object): 'data_id' in result

Success Content in message object for data_id is correct (Content 'DID' and Type is <class 'str'>).

Result (Content in message object for data_id): 'DID' (<class 'str'>)

Expectation (Content in message object for data_id): result = 'DID' (<class 'str'>)

B.1.4 Data

Description

The Data shall hold the data to be transfered. For the most requests not data is transmitted.

Reason for the implementation

Give the possibility to transfer Data.

Fitcriterion

Data is part of the Message Object and it is holding the Data information.

Testresult

This test was passed with the state: **Success**.

Info Creating empty message object: {'data': None, 'data_id': None, 'service_id': None, 'status': None}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'data': None, 'data_id': None, 'service_id': None, 'status': None} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Info Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}

Success data is part of the message object is correct ('data' is in the list or dict).

Result (data is part of the message object): {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'} (<class 'socket_protocol.data_storage'>)

Expectation (data is part of the message object): 'data' in result

Success Content in message object for data is correct (Content 'D' and Type is <class 'str'>).

Result (Content in message object for data): 'D' (<class 'str'>)

Expectation (Content in message object for data): result = 'D' (<class 'str'>)

B.1.5 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↔ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↔ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↔ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↔ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↔ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↔ response

```

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer

```

```

prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e

comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"

prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↳ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↳ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)

B.1.6 A checksum shall ensure the correct transmission**Description**

If the checksum does not fit to the checksum of the transferred data, the message will be ignored, because the complete content including the Service- and Data-ID is possibly corrupted.

Reason for the implementation

Ensure correct data transfer.

Fitcriterion

Corrupted message is not in the receive buffer after transmission.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0

```

prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1c 3a 3e

comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1c 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1c

prot-server: Received message has an invalid checksum. Message will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Checksum Error → No message received by server is correct (Content None and Type is <class 'NoneType'>).

```
Result (Checksum Error -> No message received by server): None (<class 'NoneType'>)
```

```
Expectation (Checksum Error -> No message received by server): result = None (<class
↳ 'NoneType'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.

```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Checksum Error → No message received by client is correct (Content None and Type is <class 'NoneType'>).

Result (Checksum Error -> No message received by client): None (<class 'NoneType'>)

Expectation (Checksum Error -> No message received by client): result = None (<class 'NoneType'>)

B.1.7 An authentication between server and client shall be possible including status feedback methods

Description

The Client shall have a method to initiate the authentication. In case that the server and the client do have identical secrets, the authentication shall be successfull.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

Check authentication method feedback (client) and authentication feedback (client and server), in case of differing and identical secrets.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.

```

prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info No secret set

Info Performing Authentication

Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).

```
Result (Return Value of authentication method): False (<class 'bool'>)
Expectation (Return Value of authentication method): result = False (<class 'bool'>)
```

Success Authentication state of server is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of server): True (<class 'bool'>)
Expectation (Authentication state of server): result = True (<class 'bool'>)
```

Success Authentication state of client is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of client): True (<class 'bool'>)
Expectation (Authentication state of client): result = True (<class 'bool'>)
```

Info Different secrets set

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of server): False (<class 'bool'>)
Expectation (Authentication state of server): result = False (<class 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

Result (Authentication state of client): False (<class 'bool'>)

Expectation (Authentication state of client): result = False (<class 'bool'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↪ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

```

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'4616f561fdef4a1932f1f0bdc999f242da125baa8db39f6af838b25e7a0c028e'"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 34 36 31 36

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 34 36 31 36

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (64): 66 35 36 31 66 64 65 66 34 61 31 39 33 32 66 31 66 30 62 64 63 39 39
↳ 39 66 32 34 32 64 61 31 32 35 62 61 61 38 64 62 33 39 66 36 61 66 38 33 38 62 32 35 65 37
↳ 61 30 63 30 32 38 65 22 7d 2a b7

comm-client: RX <- (64): 66 35 36 31 66 64 65 66 34 61 31 39 33 32 66 31 66 30 62 64 63 39 39
↳ 39 66 32 34 32 64 61 31 32 35 62 61 61 38 64 62 33 39 66 36 61 66 38 33 38 62 32 35 65 37
↳ 61 30 63 30 32 38 65 22 7d 2a b7

comm-server: TX -> (4): 7d 88 3a 3e
comm-client: RX <- (4): 7d 88 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 34 36 31 36 66 35 36 31 66 64 65 66 34 61 31 39 33 32 66 31 66 30 62 64 63 39
↳ 39 39 66 32 34 32 64 61 31 32 35 62 61 61 38 64 62 33 39 66 36 61 66 38 33 38 62 32 35 65
↳ 37 61 30 63 30 32 38 65 22 7d 2a b7 7d 88

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'4616f561fdef4a1932f1f0bdc999f242da125baa8db39f6af838b25e7a0c028e'"

prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'f756e52172814b4ef76809dbec28e50b7a5ae8ef1a8ce8fd7514e091d0b0dc5dd31ff0780e6b8850ea6bdbj'
↳ 65b52caa229b0f230b40262f0e520fe2516eb7fe5'"

```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 66 37 35 36
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 66 37 35 36
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (64): 65 35 32 31 37 32 38 31 34 62 34 65 66 37 36 38 30 39 64 62 65 62 63
↪ 32 38 65 35 30 62 37 61 35 61 65 38 65 66 31 61 38 63 65 38 66 64 37 35 31 34 65 30 39 31
↪ 64 30 62 30 64 63 35 64 64 33 31
```

```
comm-server: RX <- (64): 65 35 32 31 37 32 38 31 34 62 34 65 66 37 36 38 30 39 64 62 65 62 63
↪ 32 38 65 35 30 62 37 61 35 61 65 38 65 66 31 61 38 63 65 38 66 64 37 35 31 34 65 30 39 31
↪ 64 30 62 30 64 63 35 64 64 33 31
```

```
comm-client: TX -> (64): 66 66 30 37 38 30 65 36 62 38 38 35 30 65 61 36 62 64 62 36 35 62 35
↪ 32 63 61 61 32 32 39 62 30 66 32 33 30 62 34 30 32 36 32 66 30 65 35 32 30 66 65 32 35 31
↪ 36 65 62 37 66 65 35 22 7d a2 cd
```

```
comm-server: RX <- (64): 66 66 30 37 38 30 65 36 62 38 38 35 30 65 61 36 62 64 62 36 35 62 35
↪ 32 63 61 61 32 32 39 62 30 66 32 33 30 62 34 30 32 36 32 66 30 65 35 32 30 66 65 32 35 31
↪ 36 65 62 37 66 65 35 22 7d a2 cd
```

```
comm-client: TX -> (4): 4e 8c 3a 3e
```

```
comm-server: RX <- (4): 4e 8c 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 22 66 37 35 36 65 35 32 31 37 32 38 31 34 62 34 65 66 37 36 38 30 39 64 62 65 62
↪ 63 32 38 65 35 30 62 37 61 35 61 65 38 65 66 31 61 38 63 65 38 66 64 37 35 31 34 65 30 39
↪ 31 64 30 62 30 64 63 35 64 64 33 31 66 66 30 37 38 30 65 36 62 38 38 35 30 65 61 36 62 64
↪ 62 36 35 62 35 32 63 61 61 32 32 39 62 30 66 32 33 30 62 34 30 32 36 32 66 30 65 35 32 30
↪ 66 65 32 35 31 36 65 62 37 66 65 35 22 7d a2 cd 4e 8c
```

```

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'f756e52172814b4ef76809dbebc28e50b7a5ae8ef1a8ce8fd7514e091d0b0dc5dd31ff0780e6b8850ea6bdb'
↳ 65b52caa229b0f230b40262f0e520fe2516eb7fe5'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "False"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 66 61 6c 73 65

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 66 61 6c 73 65

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (7): 7d ea 0a 5c b4 3a 3e

comm-client: RX <- (7): 7d ea 0a 5c b4 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (63): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 66 61 6c 73 65 7d ea 0a 5c b4

prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "False"

prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got negative authentication feedback

```

Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).

Result (Return Value of authentication method): False (<class 'bool'>)

Expectation (Return Value of authentication method): result = False (<class 'bool'>)

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

Result (Authentication state of server): False (<class 'bool'>)

Expectation (Authentication state of server): result = False (<class 'bool'>)

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

Result (Authentication state of client): False (<class 'bool'>)

Expectation (Authentication state of client): result = False (<class 'bool'>)

Info Identical secrets set

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

```

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'002cbb211c174deef064db64f068d58da371898394753117c79dabc26a390edb'"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 30 30 32 63

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 30 30 32 63

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 62 62 32 31 31 63 31 37 34 64 65 65 66 30 36 34 64 62 36 34 66 30 36
↳ 38 64 35 38 64 61 33 37 31 38 39 38 33 39 34 37 35 33 31 31 37 63 37 39 64 61 62 63 32 36
↳ 61 33 39 30 65 64 62 22 7d d2 93

comm-client: RX <- (64): 62 62 32 31 31 63 31 37 34 64 65 65 66 30 36 34 64 62 36 34 66 30 36
↳ 38 64 35 38 64 61 33 37 31 38 39 38 33 39 34 37 35 33 31 31 37 63 37 39 64 61 62 63 32 36
↳ 61 33 39 30 65 64 62 22 7d d2 93

comm-server: TX -> (4): 45 22 3a 3e

comm-client: RX <- (4): 45 22 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 30 30 32 63 62 62 32 31 31 63 31 37 34 64 65 65 66 30 36 34 64 62 36 34 66 30
↳ 36 38 64 35 38 64 61 33 37 31 38 39 38 33 39 34 37 35 33 31 31 37 63 37 39 64 61 62 63 32
↳ 36 61 33 39 30 65 64 62 22 7d d2 93 45 22

```

```

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'002cbb211c174deef064db64f068d58da371898394753117c79dabc26a390edb'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'3718d9fc8a8676f4cc5816bd2c53eeeb45edc708fc257912d3f8826dd1e7e987a2a01845dcbd2edb532d09d
↳ 7be90f9ab3c227ca9c90617247183246c4b243c01'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 33 37 31 38

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 33 37 31 38

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (64): 64 39 66 63 38 61 38 36 37 36 66 34 63 63 35 38 31 36 62 64 32 63 35
↳ 33 65 65 65 62 34 35 65 64 63 37 30 38 66 63 32 35 37 39 31 32 64 33 66 38 38 32 36 64 64
↳ 31 65 37 65 39 38 37 61 32 61 30

comm-server: RX <- (64): 64 39 66 63 38 61 38 36 37 36 66 34 63 63 35 38 31 36 62 64 32 63 35
↳ 33 65 65 65 62 34 35 65 64 63 37 30 38 66 63 32 35 37 39 31 32 64 33 66 38 38 32 36 64 64
↳ 31 65 37 65 39 38 37 61 32 61 30

comm-client: TX -> (64): 31 38 34 35 64 63 62 64 32 65 64 62 35 33 32 64 30 39 64 37 62 65 39
↳ 30 66 39 61 62 33 63 32 32 37 63 61 39 63 39 30 36 31 37 32 34 37 31 38 33 32 34 36 63 34
↳ 62 32 34 33 63 30 31 22 7d 9a ed

comm-server: RX <- (64): 31 38 34 35 64 63 62 64 32 65 64 62 35 33 32 64 30 39 64 37 62 65 39
↳ 30 66 39 61 62 33 63 32 32 37 63 61 39 63 39 30 36 31 37 32 34 37 31 38 33 32 34 36 63 34
↳ 62 32 34 33 63 30 31 22 7d 9a ed

comm-client: TX -> (4): df ae 3a 3e

comm-server: RX <- (4): df ae 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```

STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 33 37 31 38 64 39 66 63 38 61 38 36 37 36 66 34 63 63 35 38 31 36 62 64 32 63
↳ 35 33 65 65 65 62 34 35 65 64 63 37 30 38 66 63 32 35 37 39 31 32 64 33 66 38 38 32 36 64
↳ 64 31 65 37 65 39 38 37 61 32 61 30 31 38 34 35 64 63 62 64 32 65 64 62 35 33 32 64 30 39
↳ 64 37 62 65 39 30 66 39 61 62 33 63 32 32 37 63 61 39 63 39 30 36 31 37 32 34 37 31 38 33
↳ 32 34 36 63 34 62 32 34 33 63 30 31 22 7d 9a ed df ae

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'3718d9fc8a8676f4cc5816bd2c53eeeb45edc708fc257912d3f8826dd1e7e987a2a01845dcdb2edb532d09d_
↳ 7be90f9ab3c227ca9c90617247183246c4b243c01'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32

prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"

```

prot-client: Executing callback __authenticate_process_feedback__ to process received data

prot-client: Got positive authentication feedback

Success Return Value of authentication method is correct (Content True and Type is <class 'bool'>).

Result (Return Value of authentication method): True (<class 'bool'>)

Expectation (Return Value of authentication method): result = True (<class 'bool'>)

Success Authentication state of server is correct (Content True and Type is <class 'bool'>).

Result (Authentication state of server): True (<class 'bool'>)

Expectation (Authentication state of server): result = True (<class 'bool'>)

Success Authentication state of client is correct (Content True and Type is <class 'bool'>).

Result (Authentication state of client): True (<class 'bool'>)

Expectation (Authentication state of client): result = True (<class 'bool'>)

Info Corrupting the authentication mechanism

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↪ "None"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

Success Return Value of authentication method is correct (Content False and Type is <class 'bool'>).

```
Result (Return Value of authentication method): False (<class 'bool'>)
Expectation (Return Value of authentication method): result = False (<class 'bool'>)
```

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of server): False (<class 'bool'>)
Expectation (Authentication state of server): result = False (<class 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of client): False (<class 'bool'>)
Expectation (Authentication state of client): result = False (<class 'bool'>)
```

B.1.8 An automatic authentication shall available

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

Check authentication feedback (client and server) after connect has been triggered.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server: Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and ↪ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist

```

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Identical secrets set and automatic authentication

Success Authentication state of server is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of server): False (<class 'bool'>)
Expectation (Authentication state of server): result = False (<class 'bool'>)
```

Success Authentication state of client is correct (Content False and Type is <class 'bool'>).

```
Result (Authentication state of client): False (<class 'bool'>)
Expectation (Authentication state of client): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
prot-server: Executing callback __authenticate_create_seed__ to process received data
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'be08b64de6b477e81de8d9cdfe5261fa055d624753cffe8188b70856e8be9d9a'"
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 62 65 30 38

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 62 65 30 38

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 62 36 34 64 65 36 62 34 37 37 65 38 31 64 65 38 64 39 63 64 66 65 35
↳ 32 36 31 66 61 30 35 35 64 36 32 34 37 35 33 63 66 66 65 38 31 38 38 62 37 30 38 35 36 65
↳ 38 62 65 39 64 39 61 22 7d 1a a2

comm-client: RX <- (64): 62 36 34 64 65 36 62 34 37 37 65 38 31 64 65 38 64 39 63 64 66 65 35
↳ 32 36 31 66 61 30 35 35 64 36 32 34 37 35 33 63 66 66 65 38 31 38 38 62 37 30 38 35 36 65
↳ 38 62 65 39 64 39 61 22 7d 1a a2

comm-server: TX -> (4): 76 4a 3a 3e
comm-client: RX <- (4): 76 4a 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 62 65 30 38 62 36 34 64 65 36 62 34 37 37 65 38 31 64 65 38 64 39 63 64 66 65
↳ 35 32 36 31 66 61 30 35 35 64 36 32 34 37 35 33 63 66 66 65 38 31 38 38 62 37 30 38 35 36
↳ 65 38 62 65 39 64 39 61 22 7d 1a a2 76 4a

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'be08b64de6b477e81de8d9cdfe5261fa055d624753cffe8188b70856e8be9d9a'"

```

```

prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'4cab159f47293e0803dbdfd670fe4903d2ee5e1a4b8209943ce891789674944e10773526143c89aefca6495'
↳ 3efbcaf1ed0d7e535c4cee216e190fbb12bdf8d61'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 34 63 61 62
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 34 63 61 62
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (64): 31 35 39 66 34 37 32 39 33 65 30 38 30 33 64 62 64 66 64 36 37 30 66
↳ 65 34 39 30 33 64 32 65 65 35 65 31 61 34 62 38 32 30 39 39 34 33 63 65 38 39 31 37 38 39
↳ 36 37 34 39 34 34 65 31 30 37 37
comm-server: RX <- (64): 31 35 39 66 34 37 32 39 33 65 30 38 30 33 64 62 64 66 64 36 37 30 66
↳ 65 34 39 30 33 64 32 65 65 35 65 31 61 34 62 38 32 30 39 39 34 33 63 65 38 39 31 37 38 39
↳ 36 37 34 39 34 34 65 31 30 37 37
comm-client: TX -> (64): 33 35 32 36 31 34 33 63 38 39 61 65 66 63 61 36 34 39 35 33 65 66 62
↳ 63 61 66 31 65 64 30 64 37 65 35 33 35 63 34 63 65 65 32 31 36 65 31 39 30 66 62 62 31 32
↳ 62 64 66 38 64 36 31 22 7d e4 87
comm-server: RX <- (64): 33 35 32 36 31 34 33 63 38 39 61 65 66 63 61 36 34 39 35 33 65 66 62
↳ 63 61 66 31 65 64 30 64 37 65 35 33 35 63 34 63 65 65 32 31 36 65 31 39 30 66 62 62 31 32
↳ 62 64 66 38 64 36 31 22 7d e4 87
comm-client: TX -> (4): 9a 39 3a 3e
comm-server: RX <- (4): 9a 39 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```

STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 34 63 61 62 31 35 39 66 34 37 32 39 33 65 30 38 30 33 64 62 64 66 64 36 37 30
↳ 66 65 34 39 30 33 64 32 65 65 35 65 31 61 34 62 38 32 30 39 39 34 33 63 65 38 39 31 37 38
↳ 39 36 37 34 39 34 34 65 31 30 37 37 33 35 32 36 31 34 33 63 38 39 61 65 66 63 61 36 34 39
↳ 35 33 65 66 62 63 61 66 31 65 64 30 64 37 65 35 33 35 63 34 63 65 65 32 31 36 65 31 39 30
↳ 66 62 62 31 32 62 64 66 38 64 36 31 22 7d e4 87 9a 39

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'4cab159f47293e0803dbdfd670fe4903d2ee5e1a4b8209943ce891789674944e10773526143c89aefca6495'
↳ 3efbcacf1ed0d7e535c4cee216e190fbb12bdf8d61'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 94 fe 74 32 3a 3e

comm-client: RX <- (6): 94 fe 74 32 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32

prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"

```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback
```

Success Authentication state of server is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of server): True (<class 'bool'>)
```

```
Expectation (Authentication state of server): result = True (<class 'bool'>)
```

Success Authentication state of client is correct (Content True and Type is <class 'bool'>).

```
Result (Authentication state of client): True (<class 'bool'>)
```

```
Expectation (Authentication state of client): result = True (<class 'bool'>)
```

B.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed.

Description

Communication (rx and tx) shall be disabled, if a secret is given. Except of a response for registered services, saying that a Authentication is required.

Reason for the implementation

Message protection (e.g. for secure functions or data)

Fitcriterion

RX and TX is not possible, till a successfull authentication has been performed.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incomming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
```

```

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0

```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

```

prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
↪ "'msg3_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (32): 67 33 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 13 e9 64 3d 3a 3e
```

```
comm-server: RX <- (32): 67 33 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 13 e9 64 3d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 33 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 13 e9 64 3d
```

```
prot-server: RX <- service: execute request, data_id: 36, status: okay, data:
↳ "'msg3_data_to_be_transferred'"
```

```
prot-server: Authentification is required. Just sending negative response.
```

```
prot-server: TX -> service: execute response, data_id: 36, status: authentification required,
↳ data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 33 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 33 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (8): 6c 7d 5d 78 af a4 3a 3e
comm-client: RX <- (8): 6c 7d 5d 78 af a4 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (64): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 36 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 33 31 2c 20 22 73 74 61 74 75 73 22 3a 20 33 2c 20 22 64 61
↳ 74 61 22 3a 20 6e 75 6c 6c 7d 5d 78 af a4
prot-client: RX <- service: execute response, data_id: 36, status: authentication required,
↳ data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Client send Method): True (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)

Success Received message on server side is correct (Content {'data_id': 36, 'service_id': 31, 'status': 3, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {'data_id': 36, 'service_id': 31, 'status': 3,
↳ 'data': None} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'service_id': 31, 'data_id': 36,
↳ 'status': 3, 'data': None} (<class 'socket_protocol.data_storage'>)

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

```

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"

```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
prot-client: Authentication is required. Incoming message will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Identical secrets set

Info Transferring a message client → server

prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
 ↳ data: 'msg1_data_to_be_transferred' will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
 ↳ in buffer.

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Client send Method): False (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = False (<class 'bool'>)

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on server side): None (<class 'NoneType'>)

Expectation (Received message on server side): result = None (<class 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
 ↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
 ↳ in buffer.

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): False (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Performing Authentication

prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
 ↳ "None"

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
```

```
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9
```

```
prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
prot-server: Executing callback __authenticate_create_seed__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'451f13ae0386cc7355a4c1443806ddccea870f3bea831130d1569429cb9891a2'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 34 35 31 66
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 34 35 31 66
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (64): 31 33 61 65 30 33 38 36 63 63 37 33 35 35 61 34 63 31 34 34 33 38 30
↳ 36 64 64 63 63 61 65 38 37 30 66 33 62 65 61 38 33 31 31 33 30 64 31 35 36 39 34 32 39 63
↳ 62 39 38 39 31 61 32 22 7d 64 03

comm-client: RX <- (64): 31 33 61 65 30 33 38 36 63 63 37 33 35 35 61 34 63 31 34 34 33 38 30
↳ 36 64 64 63 63 61 65 38 37 30 66 33 62 65 61 38 33 31 31 33 30 64 31 35 36 39 34 32 39 63
↳ 62 39 38 39 31 61 32 22 7d 64 03

comm-server: TX -> (4): ab ff 3a 3e
comm-client: RX <- (4): ab ff 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 34 35 31 66 31 33 61 65 30 33 38 36 63 63 37 33 35 35 61 34 63 31 34 34 33 38
↳ 30 36 64 64 63 63 61 65 38 37 30 66 33 62 65 61 38 33 31 31 33 30 64 31 35 36 39 34 32 39
↳ 63 62 39 38 39 31 61 32 22 7d 64 03 ab ff

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'451f13ae0386cc7355a4c1443806ddccae870f3bea831130d1569429cb9891a2'"

prot-client: Executing callback __authenticate_create_key__ to process received data
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'0f1138343d3a753e5ed3fbf566ada8f58923b611e7fec2132eabf425ca1ac6653e665048ecb59349fb60d4e_'
↳ 85187c72be3963e33fb1f3d77964c13f26d41c3cc'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 30 66 31 31

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 30 66 31 31

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (64): 33 38 33 34 33 64 33 61 37 35 33 65 35 65 64 33 66 62 66 35 36 36 61
↳ 64 61 38 66 35 38 39 32 33 62 36 31 31 65 37 66 65 63 32 31 33 32 65 61 62 66 34 32 35 63
↳ 61 31 61 63 36 36 35 33 65 36 36

comm-server: RX <- (64): 33 38 33 34 33 64 33 61 37 35 33 65 35 65 64 33 66 62 66 35 36 36 61
↳ 64 61 38 66 35 38 39 32 33 62 36 31 31 65 37 66 65 63 32 31 33 32 65 61 62 66 34 32 35 63
↳ 61 31 61 63 36 36 35 33 65 36 36

comm-client: TX -> (64): 35 30 34 38 65 63 62 35 39 33 34 39 66 62 36 30 64 34 65 38 35 31 38
↳ 37 63 37 32 62 65 33 39 36 33 65 33 33 66 62 31 66 33 64 37 37 39 36 34 63 31 33 66 32 36
↳ 64 34 31 63 33 63 63 22 7d 3c b8

comm-server: RX <- (64): 35 30 34 38 65 63 62 35 39 33 34 39 66 62 36 30 64 34 65 38 35 31 38
↳ 37 63 37 32 62 65 33 39 36 33 65 33 33 66 62 31 66 33 64 37 37 39 36 34 63 31 33 66 32 36
↳ 64 34 31 63 33 63 63 22 7d 3c b8

comm-client: TX -> (4): fe 39 3a 3e

comm-server: RX <- (4): fe 39 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 30 66 31 31 33 38 33 34 33 64 33 61 37 35 33 65 35 65 64 33 66 62 66 35 36 36
↳ 61 64 61 38 66 35 38 39 32 33 62 36 31 31 65 37 66 65 63 32 31 33 32 65 61 62 66 34 32 35
↳ 63 61 31 61 63 36 36 35 33 65 36 36 35 30 34 38 65 63 62 35 39 33 34 39 66 62 36 30 64 34
↳ 65 38 35 31 38 37 63 37 32 62 65 33 39 36 33 65 33 33 66 62 31 66 33 64 37 37 39 36 34 63
↳ 31 33 66 32 36 64 34 31 63 33 63 63 22 7d 3c b8 fe 39

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'0f1138343d3a753e5ed3fbf566ada8f58923b611e7fec2132eabf425ca1ac6653e665048ecb59349fb60d4e_'
↳ 85187c72be3963e33fb1f3d77964c13f26d41c3cc'"

prot-server: Executing callback __authenticate_check_key__ to process received data

prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d

```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32
```

```
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
```

```
prot-client: Executing callback __authenticate_process_feedback__ to process received data
```

```
prot-client: Got positive authentication feedback
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Client send Method): True (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'}) and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)

Info Transferring a message server → client

```

prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"

```

```

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↪ 64 22 7d 73 e9 96 7f 3a 3e

comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↪ 64 22 7d 73 e9 96 7f 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↪ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↪ 65 64 22 7d 73 e9 96 7f

prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↪ "'msg2_data_to_be_transferred'"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↪ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

```
Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↪ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↪ 'socket_protocol.data_storage'>)
```

B.1.10 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts

Description

It shall be possible to add a specific message, identified by Service-ID and Data-ID, to a whitelist. All messages added to that whitelist shall be transmitted and received, if no authentication was successful performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentication mechanism.

Fitcriterion

Transmission and Reception will be enabled, after the message has been added to the whitelist.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response

```

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer

```

```

prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
    
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Identical secrets set

Info Transferring a message client → server

```

prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↳ data: 'msg1_data_to_be_transferred' will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.

```

Success Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Client send Method): False (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = False (<class 'bool'>)

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on server side): None (<class 'NoneType'>)

Expectation (Received message on server side): result = None (<class 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): False (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Added msg1 to client whitelist (sid=17, did=34)

prot-client: Adding Message (service: 17, data_id: 34) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Authentication is required. Incomming message will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Client send Method): True (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on server side): None (<class 'NoneType'>)

Expectation (Received message on server side): result = None (<class 'NoneType'>)

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↳ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↳ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): False (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Added msg1 to server whitelist (sid=17, did=34)

prot-server: Adding Message (service: 17, data_id: 34) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↪ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72

↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22

↪ 64 61 74 61 22 3a 3d 20 22 6d 73

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72

↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22

↪ 64 61 74 61 22 3a 3d 20 22 6d 73

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_STORE_DATA

comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65

↪ 64 22 7d 4c bc bd 1b 3a 3e

comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65

↪ 64 22 7d 4c bc bd 1b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->

↪ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72

↪ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61

↪ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72

↪ 65 64 22 7d 4c bc bd 1b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:

↪ "'msg1_data_to_be_transferred'"

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↪ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↪ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↪ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: Authentification is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.
```

Success Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): False (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)
```

Success Received message on client side is correct (Content None and Type is <class 'NoneType'>).

```
Result (Received message on client side): None (<class 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<class 'NoneType'>)
```

Info Added msg2 to client and server whitelist (sid=17, did=35)

```
prot-client: Adding Message (service: 17, data_id: 35) to the authentification whitelist
```

```
prot-server: Adding Message (service: 17, data_id: 35) to the authentification whitelist
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
```

```
comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↳ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
comm-client: RX <- (32): 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 73 e9 96 7f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 34 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 32 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 73 e9 96 7f
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of Server send Method): True (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35, 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)

B.1.11 Define a channel name for the server and client after connection is established

Description

After the connection is established, the client will initiate the channel name exchange. The channel name defined on the client side will be dominant.

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

Perform a channel name exchange with no channel name definition, differing channel name definition and identical channel name definition. In all cases, the channel name of the client will be used. Perform two channel name exchanges with only one channel name definition. This definition will be used.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

```

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

```

```
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting no Channel name for server and client

Success Channel name of server is correct (Content None and Type is <class 'NoneType'>).

Result (Channel name of server): None (<class 'NoneType'>)

Expectation (Channel name of server): result = None (<class 'NoneType'>)

Success Channel name of client is correct (Content None and Type is <class 'NoneType'>).

Result (Channel name of client): None (<class 'NoneType'>)

Expectation (Channel name of client): result = None (<class 'NoneType'>)

Info Setting different Channel names for client and Server

Info Connecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↪ "'client'"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 63 6c 69 65

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 63 6c 69 65

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (10): 6e 74 22 7d ee af 7b 7e 3a 3e
comm-server: RX <- (10): 6e 74 22 7d ee af 7b 7e 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (66): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 63 6c 69 65 6e 74 22 7d ee af 7b 7e

prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "'client'"

prot-server: Executing callback __channel_name_request__ to process received data
prot-server: overwriting user defined channel name from 'server' to 'client'

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).

```
Result (Channel name of server): 'client' (<class 'str'>)
Expectation (Channel name of server): result = 'client' (<class 'str'>)
```

Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).

```
Result (Channel name of client): 'client' (<class 'str'>)
Expectation (Channel name of client): result = 'client' (<class 'str'>)
```

Info Setting identical Channel names for client and server

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 75 6e 69 74
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 75 6e 69 74
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (12): 74 65 73 74 22 7d f8 f6 c9 e9 3a 3e
comm-server: RX <- (12): 74 65 73 74 22 7d f8 f6 c9 e9 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (68): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 75 6e 69 74 74 65 73 74 22 7d f8 f6 c9 e9
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Channel name of server is correct (Content 'unittest' and Type is <class 'str'>).

```
Result (Channel name of server): 'unittest' (<class 'str'>)
```

```
Expectation (Channel name of server): result = 'unittest' (<class 'str'>)
```

Success Channel name of client is correct (Content 'unittest' and Type is <class 'str'>).

```
Result (Channel name of client): 'unittest' (<class 'str'>)
```

```
Expectation (Channel name of client): result = 'unittest' (<class 'str'>)
```

Info Setting Channel name for client only

Info Connecting Server and Client

```
comm-client: Connection Lost...
```

```
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
comm-server: Connection Lost...
```

```
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

```
comm-client: Connection established...
```

```
comm-client: Cleaning up receive-buffer
```

```
prot-client: Cleaning up receive-buffer
```

```
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
```

```
comm-server: Connection established...
```

```
comm-server: Cleaning up receive-buffer
```

```
prot-server: Cleaning up receive-buffer
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 6c 69 65
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 6c 69 65
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (10): 6e 74 22 7d ee af 7b 7e 3a 3e
comm-server: RX <- (10): 6e 74 22 7d ee af 7b 7e 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 63 6c 69 65 6e 74 22 7d ee af 7b 7e
prot-server: RX <- service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: channel name is now 'client'
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).

```
Result (Channel name of server): 'client' (<class 'str'>)
Expectation (Channel name of server): result = 'client' (<class 'str'>)
```

Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).

```
Result (Channel name of client): 'client' (<class 'str'>)
Expectation (Channel name of client): result = 'client' (<class 'str'>)
```

Info Setting Channel name for server only

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data:
↳ "'server'"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 73 65 72 76
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 73 65 72 76
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (10): 65 72 22 7d ac a3 7b cc 3a 3e
comm-client: RX <- (10): 65 72 22 7d ac a3 7b cc 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (66): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 22 73 65 72 76 65 72 22 7d ac a3 7b cc
prot-client: RX <- service: channel name response, data_id: name, status: okay, data:
↳ "'server'"
prot-client: Executing callback __channel_name_response__ to process received data
prot-client: channel name is now 'server'

```

Success Channel name of server is correct (Content 'server' and Type is <class 'str'>).

```

Result (Channel name of server): 'server' (<class 'str'>)
Expectation (Channel name of server): result = 'server' (<class 'str'>)

```

Success Channel name of client is correct (Content 'server' and Type is <class 'str'>).

```

Result (Channel name of client): 'server' (<class 'str'>)
Expectation (Channel name of client): result = 'server' (<class 'str'>)

```

B.1.12 The User shall be able to define a new service

Description

The service is defined by a Request Service-ID and a Response Service-ID.

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

Define a service and check, that the server will respond on the new Service-ID. The Status shall be "Request has no callback. Data buffered.", because no callback is registered for that request.

Testresult

This test was passed with the state: **Success**.

Info	Setting up communication
comm-client:	Cleaning up receive-buffer
comm-server:	Cleaning up receive-buffer
comm-server:	Waiting for incoming connection
prot-server:	Cleaning up receive-buffer
prot-server:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-server:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication request, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding Message (service: authentication response, data_id: key) to the ↪ authentication whitelist
prot-server:	Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server:	Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server:	Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server:	Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server:	Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server:	Adding Service with Request=channel name request and Response=channel name ↪ response
prot-server:	Adding Message (service: channel name request, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding Message (service: channel name response, data_id: name) to the ↪ authentication whitelist
prot-server:	Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server:	Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server:	Adding Service with Request=read data request and Response=read data response
prot-server:	Adding Service with Request=write data request and Response=write data response
prot-server:	Adding Service with Request=execute request and Response=execute response
prot-server:	Initialisation finished.
prot-client:	Cleaning up receive-buffer
prot-client:	Adding Service with Request=authentication request and ↪ Response=authentication response
prot-client:	Adding Message (service: authentication request, data_id: seed) to the ↪ authentication whitelist
prot-client:	Adding Message (service: authentication response, data_id: seed) to the ↪ authentication whitelist

```

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e

comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 18; data_id: 34) not
↳ in buffer.
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content None and Type is <class 'NoneType'>).

```
Result (Received message on server side): None (<class 'NoneType'>)
```

```
Expectation (Received message on server side): result = None (<class 'NoneType'>)
```

Info Adding service to server instance for the transmit message

```
prot-server: Adding Service with Request=17 and Response=18
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e

comm-server: RX <- (32): 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65
↳ 64 22 7d 4c bc bd 1b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (88): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61
↳ 74 61 22 3a 20 22 6d 73 67 31 5f 64 61 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72
↳ 65 64 22 7d 4c bc bd 1b

prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"

prot-server: Incoming message with no registered callback. Sending negative response.

prot-server: TX -> service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (8): 6c 7d bd 30 46 9b 3a 3e

comm-client: RX <- (8): 6c 7d bd 30 46 9b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (64): 7b 22 64 61 74 61 5f 69 64 22 3a 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61
↪ 74 61 22 3a 20 6e 75 6c 6c 7d bd 30 46 9b
```

```
prot-client: RX <- service: 18, data_id: 34, status: no callback for service, data buffered,
↪ data: "None"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 18, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 18, 'status': 1,
↪ 'data': None} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 18, 'data_id': 34,
↪ 'status': 1, 'data': None} (<class 'socket_protocol.data_storage'>)
```

B.1.13 Registration of already registered request Service-ID or response Service-ID shall not be possible

Description

An exception shall be raised, if a service registration with an existing request SID or response SID is performed.

Reason for the implementation

Changing existing services will create strange situations with already registered callbacks.

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist
```

```

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

```

```
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Adding a service with an already registered request SID

```
prot-server: Service with Request-SID=10 and Response-SID=18 not added, because request SID
↳ is already registered
```

Success Expected Exception RequestSidExistsError was triggered

Info Adding a service with an already registered response SID

prot-server: Service with Request-SID=17 and Response-SID=11 not added, because response SID
↳ is already registered

Success Expected Exception ResponseSidExistsError was triggered

B.1.14 It shall be possible to register a callback for a specific Service- and Data-ID

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
```

```

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer

```

prot-server: Cleaning up receive-buffer

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```

prot-server: Adding callback '__callback__' for SID=10 and DID=0

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Info Overwriting existing Callback using one with faulty (too many) return values

```
prot-server: Overwriting existing callback '__callback__' for service_id (10) and data_id (0)
↳ to '__callback_error__'!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback_error__ to process received data
prot-server: Exception raised. Check callback __callback_error__: "too many values to unpack
↳ (expected 2)" and it's return values for service_id 10 and data_id 0
prot-server: TX -> service: read data response, data_id: 0, status: callback error, data:
↳ "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 32 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d a1 a2 87 f3 3a 3e
comm-client: RX <- (7): 7d a1 a2 87 f3 3a 3e

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (63): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 32 2c 20 22 64 61 74
↪ 61 22 3a 20 6e 75 6c 6c 7d a1 a2 87 f3

prot-client: RX <- service: read data response, data_id: 0, status: callback error, data:
↪ "None"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↪ 'data': 31} (<class 'socket_protocol.data_storage'>)

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 2, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 2, 'data':
↪ None} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'data': None, 'data_id': 0, 'service_id':
↪ 11, 'status': 2} (<class 'socket_protocol.data_storage'>)

Info Removing the registered Callback

prot-server: Deleting existing callback '__callback_error__' for service_id (10) and data_id
↪ (0)!

Info Transferring data

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (5): 5b f5 78 3a 3e

comm-server: RX <- (5): 5b f5 78 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Incomming message with no registered callback. Sending negative response.

prot-server: TX -> service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 6e 75 6c 6c

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (7): 7d 88 6a 33 01 3a 3e
comm-client: RX <- (7): 7d 88 6a 33 01 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (63): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 31 2c 20 22 64 61 74
↳ 61 22 3a 20 6e 75 6c 6c 7d 88 6a 33 01
prot-client: RX <- service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content None and Type is <class 'NoneType'>).

```
Result (Message stored inside callback): None (<class 'NoneType'>)
Expectation (Message stored inside callback): result = None (<class 'NoneType'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 1, 'data':
↳ None} (<class 'socket_protocol.data_storage'>)
Expectation (Message received by client): result = {'data': None, 'data_id': 0, 'service_id':
↳ 11, 'status': 1} (<class 'socket_protocol.data_storage'>)
```

B.1.15 It shall be possible to register a callback for a specific Service-ID and all Data-IDs

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
```

```

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

```

```
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=None
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (5): 5b f5 78 3a 3e
```

```
comm-server: RX <- (5): 5b f5 78 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↪ 61 22 3a 20 33 31 7d b8 5b f5 78
```

```
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
```

```
prot-server: Executing callback __callback__ to process received data
```

```
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

B.1.16 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer

```

comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and ↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name ↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the ↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the ↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and ↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the ↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the ↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the ↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the ↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

```

prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=0
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): e1 8c bb 3a 3e
```

```
comm-client: RX <- (5): e1 8c bb 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

B.1.17 It shall be possible to register a callback for all incoming messages**Testresult**

This test was passed with the state: **Success.**

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
  
```

```

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=None
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb

prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↪ 33} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↪ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

B.1.18 Callback choice, if several callbacks are available (caused by wildcard callbacks)

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↪ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0

prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0

prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1

prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↪ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↪ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0

```

prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
```

```
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
```

```
prot-server: Executing callback __channel_name_request__ to process received data
```

```
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering all kind of Callbacks

```

prot-server: Adding callback '__callback3__' for SID=None and DID=None
prot-server: Adding callback '__callback2__' for SID=None and DID=0
prot-server: Adding callback '__callback1__' for SID=10 and DID=None
prot-server: Adding callback '__callback__' for SID=10 and DID=0

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (5): 5b f5 78 3a 3e

comm-server: RX <- (5): 5b f5 78 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Executing callback __callback__ to process received data

prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "33"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 33 7d e4

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (5): e1 8c bb 3a 3e

comm-client: RX <- (5): e1 8c bb 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 33 7d e4 e1 8c bb
```

```
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 33} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data- and Service-ID

```
prot-server: Deleting existing callback '__callback__' for service_id (10) and data_id (0)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback1__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
↳ data: "34"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 34 7d 53
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 34 7d 53
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 62 51 ca 3a 3e
comm-client: RX <- (5): 62 51 ca 3a 3e

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 64 61 74
↪ 61 22 3a 20 33 34 7d 53 62 51 ca

prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↪ data: "34"

prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↪ 'data': 31} (<class 'socket_protocol.data_storage'>)

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 34} and Type is <class 'socket_protocol.data_storage'>).

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 6, 'data':
↪ 34} (<class 'socket_protocol.data_storage'>)

Expectation (Message received by client): result = {'data': 34, 'data_id': 0, 'service_id':
↪ 11, 'status': 6} (<class 'socket_protocol.data_storage'>)

Info Removing Callback for a specific Service-ID and all Data-IDs

prot-server: Deleting existing callback '__callback1__' for service_id (10) and data_id
↪ (None)!

Info Transferring data

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (5): 5b f5 78 3a 3e

comm-server: RX <- (5): 5b f5 78 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"

prot-server: Executing callback __callback2__ to process received data

prot-server: TX -> service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 35 7d 4a

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 36 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 35 7d 4a

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
comm-server: TX -> (5): 79 60 8b 3a 3e
```

```
comm-client: RX <- (5): 79 60 8b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
```

```
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 36 2c 20 22 64 61 74
↳ 61 22 3a 20 33 35 7d 4a 79 60 8b
```

```
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 35} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 6, 'data':
↳ 35} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 35, 'data_id': 0, 'service_id':
↳ 11, 'status': 6} (<class 'socket_protocol.data_storage'>)
```

Info Removing Callback for a specific Data-ID and all Serice-IDs

```
prot-server: Deleting existing callback '__callback2__' for service_id (None) and data_id (0)!
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (5): 5b f5 78 3a 3e

comm-server: RX <- (5): 5b f5 78 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 31 7d b8 5b f5 78

prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Executing callback __callback3__ to process received data
prot-server: TX -> service: read data response, data_id: 0, status: okay, data: "36"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 36 7d 99

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 36 7d 99

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (5): 96 78 fe 3a 3e
comm-client: RX <- (5): 96 78 fe 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (61): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74
↳ 61 22 3a 20 33 36 7d 99 96 78 fe
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "36"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0,
↳ 'data': 31} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 36} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data':
↳ 36} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Message received by client): result = {'data': 36, 'data_id': 0, 'service_id':
↳ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)
```

B.1.19 Connection established information

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
```

```

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

```

```
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client connection status is correct (Content True and Type is <class 'bool'>).

Result (Client connection status): True (<class 'bool'>)

Expectation (Client connection status): result = True (<class 'bool'>)

Success Server connection status is correct (Content True and Type is <class 'bool'>).

Result (Server connection status): True (<class 'bool'>)

Expectation (Server connection status): result = True (<class 'bool'>)

Success Client connection status is correct (Content False and Type is <class 'bool'>).

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

Result (Client connection status): False (<class 'bool'>)

Expectation (Client connection status): result = False (<class 'bool'>)

Success Server connection status is correct (Content False and Type is <class 'bool'>).

Result (Server connection status): False (<class 'bool'>)

Expectation (Server connection status): result = False (<class 'bool'>)

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↳ STP_STATE_STORE_DATA

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client connection status): True (<class 'bool'>)
```

```
Expectation (Client connection status): result = True (<class 'bool'>)
```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server connection status): True (<class 'bool'>)
```

```
Expectation (Server connection status): result = True (<class 'bool'>)
```

Info Adding secrets to socket_protocol

Success Client connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client connection status): False (<class 'bool'>)
```

```
Expectation (Client connection status): result = False (<class 'bool'>)
```

Success Server connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server connection status): False (<class 'bool'>)
```

```
Expectation (Server connection status): result = False (<class 'bool'>)
```

Info Doing authentication

```
prot-client: TX -> service: authentication request, data_id: seed, status: okay, data:
↳ "None"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d fd 82 a2 a9

prot-server: RX <- service: authentication request, data_id: seed, status: okay, data:
↳ "None"

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'fbf590678b2272ef3b98b70cd136769c503964530e6ba8a5b02731d41f65de02'"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 66 62 66 35

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 66 62 66 35

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

```

```

comm-server: TX -> (64): 39 30 36 37 38 62 32 32 37 32 65 66 33 62 39 38 62 37 30 63 64 31 33
↪ 36 37 36 39 63 35 30 33 39 36 34 35 33 30 65 36 62 61 38 61 35 62 30 32 37 33 31 64 34 31
↪ 66 36 35 64 65 30 32 22 7d 48 a5

comm-client: RX <- (64): 39 30 36 37 38 62 32 32 37 32 65 66 33 62 39 38 62 37 30 63 64 31 33
↪ 36 37 36 39 63 35 30 33 39 36 34 35 33 30 65 36 62 61 38 61 35 62 30 32 37 33 31 64 34 31
↪ 66 36 35 64 65 30 32 22 7d 48 a5

comm-server: TX -> (4): a2 5e 3a 3e

comm-client: RX <- (4): a2 5e 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE

STP: message identified - (124): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 22 66 62 66 35 39 30 36 37 38 62 32 32 37 32 65 66 33 62 39 38 62 37 30 63 64 31
↪ 33 36 37 36 39 63 35 30 33 39 36 34 35 33 30 65 36 62 61 38 61 35 62 30 32 37 33 31 64 34
↪ 31 66 36 35 64 65 30 32 22 7d 48 a5 a2 5e

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↪ "'fbf590678b2272ef3b98b70cd136769c503964530e6ba8a5b02731d41f65de02'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↪ "'475386d87f4d0af53efc7c5251884b62def4943ea909cae1353b96cee08db630820fd760c49c33690d21111'
↪ b91154bbd9cb6864a88d37de001645eea71cd87c7'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 34 37 35 33

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 34 37 35 33

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

```

```
comm-client: TX -> (64): 38 36 64 38 37 66 34 64 30 61 66 35 33 65 66 63 37 63 35 32 35 31 38
↪ 38 34 62 36 32 64 65 66 34 39 34 33 65 61 39 30 39 63 61 65 31 33 35 33 62 39 36 63 65 65
↪ 30 38 64 62 36 33 30 38 32 30 66
```

```
comm-server: RX <- (64): 38 36 64 38 37 66 34 64 30 61 66 35 33 65 66 63 37 63 35 32 35 31 38
↪ 38 34 62 36 32 64 65 66 34 39 34 33 65 61 39 30 39 63 61 65 31 33 35 33 62 39 36 63 65 65
↪ 30 38 64 62 36 33 30 38 32 30 66
```

```
comm-client: TX -> (64): 64 37 36 30 63 34 39 63 33 33 36 39 30 64 32 31 31 31 31 62 39 31 31
↪ 35 34 62 62 64 39 63 62 36 38 36 34 61 38 38 64 33 37 64 65 30 30 31 36 34 35 65 65 61 37
↪ 31 63 64 38 37 63 37 22 7d c0 89
```

```
comm-server: RX <- (64): 64 37 36 30 63 34 39 63 33 33 36 39 30 64 32 31 31 31 31 62 39 31 31
↪ 35 34 62 62 64 39 63 62 36 38 36 34 61 38 38 64 33 37 64 65 30 30 31 36 34 35 65 65 61 37
↪ 31 63 64 38 37 63 37 22 7d c0 89
```

```
comm-client: TX -> (4): 5f 92 3a 3e
```

```
comm-server: RX <- (4): 5f 92 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (188): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 30 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 22 34 37 35 33 38 36 64 38 37 66 34 64 30 61 66 35 33 65 66 63 37 63 35 32 35 31
↪ 38 38 34 62 36 32 64 65 66 34 39 34 33 65 61 39 30 39 63 61 65 31 33 35 33 62 39 36 63 65
↪ 65 30 38 64 62 36 33 30 38 32 30 66 64 37 36 30 63 34 39 63 33 33 36 39 30 64 32 31 31 31
↪ 31 62 39 31 31 35 34 62 62 64 39 63 62 36 38 36 34 61 38 38 64 33 37 64 65 30 30 31 36 34
↪ 35 65 65 61 37 31 63 64 38 37 63 37 22 7d c0 89 5f 92
```

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↪ "'475386d87f4d0af53efc7c5251884b62def4943ea909cae1353b96cee08db630820fd760c49c33690d21111'
↪ b91154bbd9cb6864a88d37de001645eea71cd87c7'"
```

```
prot-server: Executing callback __authenticate_check_key__ to process received data
```

```
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↪ "True"
```

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 74 72 75 65 7d
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
```

```

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 31 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 74 72 75 65 7d 94 fe 74 32
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback

```

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```

Result (Client connection status): True (<class 'bool'>)
Expectation (Client connection status): result = True (<class 'bool'>)

```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```

Result (Server connection status): True (<class 'bool'>)
Expectation (Server connection status): result = True (<class 'bool'>)

```

B.1.20 Is connected information

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

```

```

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response

prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist

prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

```

```
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f

prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Client Communication instance connection status): True (<class 'bool'>)

Expectation (Client Communication instance connection status): result = True (<class 'bool'>)

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Server Communication instance connection status): True (<class 'bool'>)

Expectation (Server Communication instance connection status): result = True (<class 'bool'>)

Info Disconnecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).

Result (Client Communication instance connection status): False (<class 'bool'>)

Expectation (Client Communication instance connection status): result = False (<class 'bool'>)

Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

Result (Server Communication instance connection status): False (<class 'bool'>)

Expectation (Server Communication instance connection status): result = False (<class 'bool'>)

B.1.21 Reconnect Method

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

prot-server: Adding Service with Request=authentication request and
↪ Response=authentication response

prot-server: Adding Message (service: authentication request, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: seed) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication request, data_id: key) to the
↪ authentication whitelist

prot-server: Adding Message (service: authentication response, data_id: key) to the
↪ authentication whitelist

```

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response

```

prot-client: Adding Service with Request=execute request and Response=execute response

prot-client: Initialisation finished.

Info Connecting Server and Client

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
 ↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_STORE_DATA

comm-client: TX -> (6): 53 5e 67 0b 3a 3e

comm-server: RX <- (6): 53 5e 67 0b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
 ↪ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
 ↪ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
 ↪ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"

prot-server: Executing callback __channel_name_request__ to process received data

```

prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Client Communication instance connection status): True (<class 'bool'>)

Expectation (Client Communication instance connection status): result = True (<class 'bool'>)

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

Result (Server Communication instance connection status): True (<class 'bool'>)

Expectation (Server Communication instance connection status): result = True (<class 'bool'>)

Info Disconnecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): False (<class 'bool'>)
Expectation (Client Communication instance connection status): result = False (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): False (<class 'bool'>)
Expectation (Server Communication instance connection status): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-client: TX -> (6): 53 5e 67 0b 3a 3e

comm-server: RX <- (6): 53 5e 67 0b 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 20 38 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↳ 22 3a 20 6e 75 6c 6c 7d 53 5e 67 0b

prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 39 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1

STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: store sync pattern (3a 3d) received => changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_STORE_DATA

comm-server: TX -> (6): 30 59 be 2f 3a 3e

comm-client: RX <- (6): 30 59 be 2f 3a 3e

STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2

STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE

```

```
STP: message identified - (62): 7b 22 64 61 74 61 5f 69 64 22 3a 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 20 39 2c 20 22 73 74 61 74 75 73 22 3a 20 30 2c 20 22 64 61 74 61
↪ 22 3a 20 6e 75 6c 6c 7d 30 59 be 2f
```

```
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
```

```
prot-client: Executing callback __channel_name_response__ to process received data
```

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): True (<class 'bool'>)
```

```
Expectation (Client Communication instance connection status): result = True (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): True (<class 'bool'>)
```

```
Expectation (Server Communication instance connection status): result = True (<class 'bool'>)
```

B.1.22 A full Message Object including the defined properties and data shall be transfered.

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

Send two different messages and compare the received message with each sent message.

Testresult

This test was passed with the state: **Success**.

Info Setting up communication

```
comm-client: Cleaning up receive-buffer
```

```
comm-server: Cleaning up receive-buffer
```

```
comm-server: Waiting for incoming connection
```

```
prot-server: Cleaning up receive-buffer
```

```
prot-server: Adding Service with Request=authentication request and
```

```
↪ Response=authentication response
```

```
prot-server: Adding Message (service: authentication request, data_id: seed) to the
```

```
↪ authentication whitelist
```

```
prot-server: Adding Message (service: authentication response, data_id: seed) to the
```

```
↪ authentication whitelist
```

```
prot-server: Adding Message (service: authentication request, data_id: key) to the
```

```
↪ authentication whitelist
```

```

prot-server: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-server: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-server: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.
prot-client: Cleaning up receive-buffer
prot-client: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-client: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication request, data_id: key) to the
↳ authentication whitelist
prot-client: Adding Message (service: authentication response, data_id: key) to the
↳ authentication whitelist
prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response
prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist
prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0

```

```
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (21): 3a 3c 00 00 00 00 00 00 00 00 08 00 00 00 00 00 6e 75 6c 6c 13 3a 3e
comm-server: RX <- (21): 3a 3c 00 00 00 00 00 00 00 00 08 00 00 00 00 00 6e 75 6c 6c 13 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (17): 00 00 00 00 00 00 00 00 08 00 00 00 00 00 6e 75 6c 6c 13
prot-server: RX <- service: channel name request, data_id: name, status: okay, data: "None"
prot-server: Executing callback __channel_name_request__ to process received data
prot-server: TX -> service: channel name response, data_id: name, status: okay, data: "None"
comm-server: TX -> (21): 3a 3c 00 00 00 00 00 00 00 00 09 00 00 00 00 00 6e 75 6c 6c 12 3a 3e
comm-client: RX <- (21): 3a 3c 00 00 00 00 00 00 00 00 09 00 00 00 00 00 6e 75 6c 6c 12 3a 3e
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↳ STP_STATE_STORE_DATA
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↳ STP_STATE_IDLE
STP: message identified - (17): 00 00 00 00 00 00 00 00 09 00 00 00 00 00 6e 75 6c 6c 12
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (45): 3a 3c 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
comm-server: RX <- (45): 3a 3c 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (41): 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61
↪ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
```

```
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method
```

Success Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Client send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = True (<class 'bool'>)
```

Success Received message on server side is correct (Content {'data_id': 34, 'service_id': 17, 'status': 0, 'data': 'msg1_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
↪ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on server side): result = {'service_id': 17, 'data_id': 34,
↪ 'status': 0, 'data': 'msg1_data_to_be_transferred'} (<class
↪ 'socket_protocol.data_storage'>)
```

Info Transferring a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪ "'msg2_data_to_be_transferred'"
```

```
comm-server: TX -> (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e
```

```
comm-client: RX <- (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↪ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e
```

```
STP: data sync (3a) received => changing state STP_STATE_IDLE -> STP_STATE_ESCAPE_1
```

```
STP: start pattern (3a 3c) received => changing state STP_STATE_ESCAPE_1 ->
↪ STP_STATE_STORE_DATA
```

```
STP: data sync (3a) received => changing state STP_STATE_STORE_DATA -> STP_STATE_ESCAPE_2
```

```
STP: end pattern (3a 3e) received => storing message and changing state STP_STATE_ESCAPE_2 ->
↪ STP_STATE_IDLE
```

```
STP: message identified - (41): 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61
↳ 74 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b
```

```
prot-client: RX <- service: 17, data_id: 35, status: service or data unknown, data:
↳ "'msg2_data_to_be_transferred'"
```

```
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method
```

Success Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of Server send Method): True (<class 'bool'>)
```

```
Expectation (Returnvalue of Server send Method): result = True (<class 'bool'>)
```

Success Received message on client side is correct (Content {'data_id': 35, 'service_id': 17, 'status': 4, 'data': 'msg2_data_to_be_transferred'} and Type is <class 'socket_protocol.data_storage'>).

```
Result (Received message on client side): {'data_id': 35, 'service_id': 17, 'status': 4,
↳ 'data': 'msg2_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

```
Expectation (Received message on client side): result = {'service_id': 17, 'data_id': 35,
↳ 'status': 4, 'data': 'msg2_data_to_be_transferred'} (<class
↳ 'socket_protocol.data_storage'>)
```

C Test-Coverage

C.1 socket_protocol

The line coverage for socket_protocol was 100.0%

The branch coverage for socket_protocol was 100.0%

C.1.1 socket_protocol.__init__.py

The line coverage for socket_protocol.__init__.py was 100.0%

The branch coverage for socket_protocol.__init__.py was 100.0%

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 #
4 """
5 socket_protocol (Socket Protocol)
6
7
8 **Author:**
9
10 * Dirk Alders <sudo-dirk@mount-mockery.de>
11
12 **Description:**
13
14 This Module supports point to point communication for client-server issues.
15
16 **Submodules:**
17
```

```

18 * :class:`socket_protocol.data_storage`
19 * :class:`socket_protocol.pure_json_protocol`
20 * :class:`socket_protocol.struct_json_protocol`
21
22 **Unittest:**
23
24     See also the :download:`unittest <socket_protocol/_testresults_/unittest.pdf>`
        documentation.
25
26 **Module Documentation:**
27
28 """
29 __DEPENDENCIES__ = ['stringtools']
30
31 import stringtools
32
33 import binascii
34 import hashlib
35 import json
36 import logging
37 import os
38 import struct
39 import sys
40 import time
41
42
43 try:
44     from config import APP_NAME as ROOT_LOGGER_NAME
45 except ImportError:
46     ROOT_LOGGER_NAME = 'root'
47 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
48
49
50 __DESCRIPTION__ = """The Module {\\tt %s} is designed for point to point communication for client
    -server issues.
51 For more Information read the sphinx documentation.""" % __name__.replace('-', '\\-')
52 """The Module Description"""
53 __INTERPRETER__ = (2, 3)
54 """The Tested Interpreter-Versions"""
55
56 SID_AUTH_REQUEST = 0
57 """SID for authentication request"""
58 SID_AUTH_RESPONSE = 1
59 """SID for authentication response"""
60 DID_AUTH_SEED = 0
61 """DID for authentication (seed)"""
62 DID_AUTH_KEY = 1
63 """DID for authentication (key)"""
64 SID_CHANNEL_NAME_REQUEST = 8
65 """SID for channel name exchange request """
66 SID_CHANNEL_NAME_RESPONSE = 9
67 """SID for channel name exchange response"""
68 DID_CHANNEL_NAME = 0
69 """DID for channel name """
70 SID_READ_REQUEST = 10
71 """SID for a read data request"""
72 SID_READ_RESPONSE = 11
73 """SID for read data response"""
74 SID_WRITE_REQUEST = 20
75 """SID for a write data request"""
76 SID_WRITE_RESPONSE = 21
77 """SID for a write data response"""
78 SID_EXECUTE_REQUEST = 30
    
```

```

79 """SID for a execute request"""
80 SID.EXECUTE_RESPONSE = 31
81 """SID for a execute response"""
82
83 STATUS.OKAY = 0
84 """Status for 'okay'"""
85 STATUS.BUFFERING_UNHANDLED_REQUEST = 1
86 """Status for 'unhandled request'"""
87 STATUS.CALLBACK_ERROR = 2
88 """Status for 'callback errors'"""
89 STATUS.AUTH_REQUIRED = 3
90 """Status for 'authentication is required'"""
91 STATUS.SERVICE_OR_DATA_UNKNOWN = 4
92 """Status for 'service or data unknown'"""
93 STATUS.CHECKSUM_ERROR = 5
94 """Status for 'checksum error'"""
95 STATUS.OPERATION_NOT_PERMITTED = 6
96 """Status for 'operation not permitted'"""
97 STATUS.LOG_LVL = {
98     STATUS.OKAY: logging.INFO,
99     STATUS.BUFFERING_UNHANDLED_REQUEST: logging.WARNING,
100     STATUS.CALLBACK_ERROR: logging.ERROR,
101     STATUS.AUTH_REQUIRED: logging.WARNING,
102     STATUS.SERVICE_OR_DATA_UNKNOWN: logging.ERROR,
103     STATUS.CHECKSUM_ERROR: logging.ERROR,
104     STATUS.OPERATION_NOT_PERMITTED: logging.WARNING,
105 }
106 """Status depending log level for messages"""
107
108 AUTH.STATE_UNTRUSTED_CONNECTION = 0
109 """Authentication Status for an 'Untrusted Connection'"""
110 AUTH.STATE_SEED_REQUESTED = 1
111 """Authentication Status for 'Seed was requested'"""
112 AUTH.STATE_SEED_TRANSFERRED = 2
113 """Authentication Status for 'Seed has been sent'"""
114 AUTH.STATE_KEY_TRANSFERRED = 3
115 """Authentication Status for 'Key has been sent'"""
116 AUTH.STATE_TRUSTED_CONNECTION = 4
117 """Authentication Status for a 'Trusted Connection'"""
118 AUTH.STATE_NAMES = {AUTH.STATE_UNTRUSTED_CONNECTION: 'Untrusted Connection',
119                     AUTH.STATE_SEED_REQUESTED: 'Seed was requested',
120                     AUTH.STATE_SEED_TRANSFERRED: 'Seed has been sent',
121                     AUTH.STATE_KEY_TRANSFERRED: 'Key has been sent',
122                     AUTH.STATE_TRUSTED_CONNECTION: 'Trusted Connection'}
123 """Authentication Status names for previous defined authentication states"""
124
125
126 class RequestSidExistsError(Exception):
127     pass
128
129
130 class ResponseSidExistsError(Exception):
131     pass
132
133
134 class _callback_storage(dict):
135     DEFAULT_CHANNEL_NAME = 'all_others'
136
137     def __init__(self, channel_name, log_prefix):
138         self.init_channel_name(channel_name)
139         self.__log_prefix__ = log_prefix
140         dict.__init__(self)
141

```

```

142 def init_channel_name(self, channel_name):
143     if channel_name is None:
144         self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
DEFAULT_CHANNEL_NAME)
145     else:
146         self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
channel_name)
147
148 def get(self, service_id, data_id):
149     if dict.get(self, service_id, {}).get(data_id, None) is not None:
150         return self[service_id][data_id]
151     elif dict.get(self, service_id, {}).get(None, None) is not None:
152         return self[service_id][None]
153     elif dict.get(self, None, {}).get(data_id, None) is not None:
154         return self[None][data_id]
155     elif dict.get(self, None, {}).get(None, None) is not None:
156         return self[None][None]
157     else:
158         return (None, None, None)
159
160 def add(self, service_id, data_id, callback, *args, **kwargs):
161     cb_data = self.get(service_id, data_id)
162     if dict.get(self, service_id, {}).get(data_id, None) is not None:
163         if callback is None:
164             self.logger.warning("%s Deleting existing callback %s for service_id (%s) and
data_id (%s)!", self.__log_prefix__(), repr(cb_data[0].__name__), repr(service_id), repr(
data_id))
165             del(self[service_id][data_id])
166             return
167         else:
168             self.logger.warning("%s Overwriting existing callback %s for service_id (%s) and
data_id (%s) to %s!", self.__log_prefix__(), repr(cb_data[0].__name__), repr(service_id),
repr(data_id), repr(callback.__name__))
169     else:
170         self.logger.debug("%s Adding callback %s for SID=%s and DID=%s", self.__log_prefix__
(), repr(callback.__name__), repr(service_id), repr(data_id))
171         if service_id not in self:
172             self[service_id] = {}
173         self[service_id][data_id] = (callback, args, kwargs)
174
175
176 class data_storage(dict):
177     """
178     This is a storage object for socket_protocol messages.
179
180     :param status: The message status.
181     :type status: int
182     :param service_id: The Service-ID.
183     :type service_id: int
184     :param data_id: The Data-ID.
185     :type data_id: int
186     :param data: The transfered data.
187     :type data: any
188     """
189
190     KEY_STATUS = 'status'
191     KEY_SERVICE_ID = 'service_id'
192     KEY_DATA_ID = 'data_id'
193     KEY_DATA = 'data'
194     ALL_KEYS = [KEY_DATA, KEY_DATA_ID, KEY_SERVICE_ID, KEY_STATUS]

```

```

196     def __init__(self, *args, **kwargs):
197         dict.__init__(self, *args, **kwargs)
198         for key in self.ALL_KEYS:
199             if key not in self:
200                 self[key] = None
201
202     def get_status(self, default=None):
203         """
204         This Method returns the message status.
205
206         :param default: The default value, if no data is available.
207         """
208         return self.get(self.KEY_STATUS, default)
209
210     def get_service_id(self, default=None):
211         """
212         This Method returns the message Service-ID.
213
214         :param default: The default value, if no data is available.
215         """
216         return self.get(self.KEY_SERVICE_ID, default)
217
218     def get_data_id(self, default=None):
219         """
220         This Method returns the message Data-ID.
221
222         :param default: The default value, if no data is available.
223         """
224         return self.get(self.KEY_DATA_ID, default)
225
226     def get_data(self, default=None):
227         """
228         This Method returns the message data.
229
230         :param default: The default value, if no data is available.
231         """
232         return self.get(self.KEY_DATA, default)
233
234
235     class pure_json_protocol(object):
236         """
237         This `class` supports to transfer a message and it's data.
238
239         :param comm_instance: A communication instance.
240         :type comm_instance: instance
241         :param secret: An optional secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
242         :type secret: str
243         :param auto_auth: An optional parameter to enable (True) automatic authentication,
244             otherwise you need to do it manually, if needed.
245         :type auto_auth: bool
246         :param channel_name: An optional parameter to set a channel name for logging of the
247             communication.
248         :type channel_name: str
249
250         .. hint::
251
252             * The Service-ID is designed to identify the type of the communication (e.g. :const:`
253               READ_REQUEST`, :const:`WRITE_REQUEST`, :const:`READ_RESPONSE`, :const:`WRITE_RESPONSE`, ...)
254             * The Data-ID is designed to identify the requests / responses using the same Service-ID.
255
256         .. note:: The :class:`comm_instance` needs to have at least the following interface:

```

```

255     * A Method :func:`comm_instance.init_channel_name` to set the channel name.
256     * A Constant :const:`comm_instance.IS_CLIENT` to identify that the :class:`comm_instance`
    is a client (True) or a server (False).
257     * A Method :func:`comm_instance.is_connected` to identify if the instance is connected (
    True) or not (False).
258     * A Method :func:`comm_instance.reconnect` to initiate a reconnect.
259     * A Method :func:`comm_instance.register_callback` to register a data available callback.
260     * A Method :func:`comm_instance.register_connect_callback` to register a connect callback
    .
261     * A Method :func:`comm_instance.register_disconnect_callback` to register a disconnect
    callback.
262     * A Method :func:`comm_instance.send` to send data via the :class:`comm_instance`.
263
264 .. note:: The parameter :const:`auto_auth` is only relevant, if a secret is given and the :
    class:`comm_instance` is a client. The authentication is initiated directly after the
    connection is established.
265
266 .. note:: The :const:`channel_name`-exchange will be initiated by the client directly after
    the the connection is established.
267
268     * If a channel_name is given at both communication sides and they are different, the
    client name is taken over and the server will log a warning message.
269 """
270 DEFAULT_CHANNEL_NAME = 'all_others'
271
272 def __init__(self, comm_instance, secret=None, auto_auth=False, channel_name=None):
273     self.__comm_inst__ = comm_instance
274     self.__secret__ = secret
275     self.__auto_auth__ = auto_auth
276     #
277     self.__auth_whitelist__ = {}
278     self.__sid_response_dict__ = {}
279     self.__sid_name_dict__ = {}
280     self.__did_name_dict__ = {}
281     #
282     self.__status_name_dict = {}
283     self.add_status(STATUS_OKAY, 'okay')
284     self.add_status(STATUS_BUFFERING_UNHANDLED_REQUEST, 'no callback for service, data
    buffered')
285     self.add_status(STATUS_CALLBACK_ERROR, 'callback error')
286     self.add_status(STATUS_AUTH_REQUIRED, 'authentication required')
287     self.add_status(STATUS_SERVICE_OR_DATA_UNKNOWN, 'service or data unknown')
288     self.add_status(STATUS_CHECKSUM_ERROR, 'checksum error')
289     self.add_status(STATUS_OPERATION_NOT_PERMITTED, 'operation not permitted')
290     #
291     self.__callbacks__ = _callback_storage(channel_name, self.__log_prefix__)
292     self.__init_channel_name__(channel_name)
293     #
294     self.__clean_receive_buffer__()
295
296     self.add_service(SID_AUTH_REQUEST, SID_AUTH_RESPONSE, 'authentication request', '
    authentication response')
297     self.add_data((SID_AUTH_REQUEST, SID_AUTH_RESPONSE), DID_AUTH_SEED, 'seed')
298     self.add_data(SID_AUTH_REQUEST, DID_AUTH_KEY, 'key')
299     self.add_data(SID_AUTH_RESPONSE, DID_AUTH_KEY, 'key')
300     self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_SEED)
301     self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_SEED)
302     self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_KEY)
303     self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_KEY)
304     self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_SEED, self.
    __authenticate_create_seed__)
    
```

```

305         self.__callbacks__.add(SID_AUTH.RESPONSE, DID_AUTH.SEED, self.
__authenticate_create_key__)
306         self.__callbacks__.add(SID_AUTH.REQUEST, DID_AUTH.KEY, self.__authenticate_check_key__)
307         self.__callbacks__.add(SID_AUTH.RESPONSE, DID_AUTH.KEY, self.
__authenticate_process_feedback__)
308         self.__authentication_state_reset__()
309
310         self.add_service(SID_CHANNEL_NAME.REQUEST, SID_CHANNEL_NAME.RESPONSE, 'channel name
request', 'channel name response')
311         self.add_data((SID_CHANNEL_NAME.REQUEST, SID_CHANNEL_NAME.RESPONSE), DID_CHANNEL_NAME, '
name')
312         self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME.REQUEST, DID_CHANNEL_NAME)
313         self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME.RESPONSE, DID_CHANNEL_NAME)
314         self.__callbacks__.add(SID_CHANNEL_NAME.REQUEST, DID_CHANNEL_NAME, self.
__channel_name_request__)
315         self.__callbacks__.add(SID_CHANNEL_NAME.RESPONSE, DID_CHANNEL_NAME, self.
__channel_name_response__)
316
317         self.add_service(SID_READ.REQUEST, SID_READ.RESPONSE, 'read data request', 'read data
response')
318         self.add_service(SID_WRITE.REQUEST, SID_WRITE.RESPONSE, 'write data request', 'write data
response')
319         self.add_service(SID_EXECUTE.REQUEST, SID_EXECUTE.RESPONSE, 'execute request', 'execute
response')
320
321         self.__seed__ = None
322         self.__comm_inst__.register_callback(self.__data_available_callback__)
323         self.__comm_inst__.register_connect_callback(self.__connection_established__)
324         self.__comm_inst__.register_disconnect_callback(self.__authentication_state_reset__)
325         logger.info('%s Initialisation finished.', self.__log_prefix__)
326
327     def __analyse_frame__(self, frame):
328         if sys.version.info >= (3, 0):
329             return data_storage(json.loads(frame[:-4].decode('utf-8')))
330         else:
331             return data_storage(json.loads(frame[:-4]))
332
333     def __authenticate_check_key__(self, msg):
334         key = msg.get_data()
335         if key == self.__authenticate_salt_and_hash__(self.__seed__):
336             self.__authentication_state__ = AUTH.STATE.TRUSTED_CONNECTION
337             return STATUS.OKAY, True
338         else:
339             self.__authentication_state__ = AUTH.STATE.UNTRUSTED_CONNECTION
340             return STATUS.OKAY, False
341
342     def __authenticate_create_key__(self, msg):
343         self.__authentication_state__ = AUTH.STATE.KEY_TRANSFERRED
344         seed = msg.get_data()
345         key = self.__authenticate_salt_and_hash__(seed)
346         self.send(SID_AUTH.REQUEST, DID_AUTH.KEY, key)
347
348     def __authenticate_create_seed__(self, msg):
349         self.__authentication_state__ = AUTH.STATE.SEED_TRANSFERRED
350         if sys.version.info >= (3, 0):
351             self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
352         else:
353             self.__seed__ = binascii.hexlify(os.urandom(32))
354         return STATUS.OKAY, self.__seed__
355

```

```

356 def __authenticate_process_feedback__(self, msg):
357     feedback = msg.get_data()
358     if feedback:
359         self.__authentication_state__ = AUTH.STATE.TRUSTED.CONNECTION
360         self.logger.info("%s Got positive authentication feedback", self.__log_prefix__())
361     else:
362         self.__authentication_state__ = AUTH.STATE.UNTRUSTED.CONNECTION
363         self.logger.warning("%s Got negative authentication feedback", self.__log_prefix__())
364     return STATUS.OKAY, None
365
366 def __authenticate_salt_and_hash__(self, seed):
367     if sys.version_info >= (3, 0):
368         return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
369     else:
370         return hashlib.sha512(seed.encode('utf-8') + self.__secret__.encode('utf-8')).hexdigest()
371
372 def __authentication_state_reset__(self):
373     self.logger.info("%s Resetting authentication state to AUTH.STATE.UNTRUSTED.CONNECTION", self.__log_prefix__())
374     self.__authentication_state__ = AUTH.STATE.UNTRUSTED.CONNECTION
375
376 def __authentication_required__(self, service_id, data_id):
377     return data_id not in self.__auth_whitelist__.get(service_id, [])
378
379 def __buffer_received_data__(self, msg):
380     if not msg.get_service_id() in self.__msg_buffer__:
381         self.__msg_buffer__[msg.get_service_id()] = {}
382     if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
383         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
384     self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
385     self.logger.debug("%s Message data is stored in buffer and is now ready to be retrieved by receive method", self.__log_prefix__())
386
387 def __build_frame__(self, msg):
388     data_frame = json.dumps(self.__mk_msg__(msg.get_status(), msg.get_service_id(), msg.get_data_id(), msg.get_data()))
389     if sys.version_info >= (3, 0):
390         data_frame = bytes(data_frame, 'utf-8')
391     checksum = self.__calc_chksum__(data_frame)
392     return data_frame + checksum
393
394 def __calc_chksum__(self, raw_data):
395     return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)
396
397 @property
398 def __channel_name__(self):
399     cn = self.logger.name.split('.')[ -1]
400     if cn != self.DEFAULT_CHANNEL_NAME:
401         return cn
402
403 def __channel_name_response__(self, msg):
404     data = msg.get_data()
405     if self.__channel_name__ is None and data is not None:
406         self.__init_channel_name__(data)
407         self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.__channel_name__))
408     return STATUS.OKAY, None
409

```

```

410 def __channel_name_request__(self, msg):
411     data = msg.get_data()
412     if data is None:
413         return STATUS_OKAY, self.__channel_name__
414     else:
415         prev_channel_name = self.__channel_name__
416         self.__init_channel_name__(data)
417         if prev_channel_name is not None and prev_channel_name != data:
418             self.logger.warning('%s overwriting user defined channel name from %s to %s',
self.__log_prefix__(), repr(prev_channel_name), repr(data))
419         elif prev_channel_name is None:
420             self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.
__channel_name__))
421         return STATUS_OKAY, None
422
423 def __check_frame_checksum__(self, frame):
424     return self.__calc_chksum__(frame[-4:]) == frame[-4:]
425
426 def __clean_receive_buffer__(self):
427     self.logger.debug('%s Cleaning up receive-buffer', self.__log_prefix__())
428     self.__msg_buffer__ = {}
429
430 def __connection_established__(self):
431     self.__clean_receive_buffer__()
432     if self.__comm_inst__.IS_CLIENT:
433         self.send(SID_CHANNEL_NAME_REQUEST, 0, self.__channel_name__)
434     if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
435         self.authenticate()
436
437 def __log_msg__(self, msg, rx_tx_prefix):
438     self.logger.log(
439         self.__status_log_lvl__(msg.get_status()),
440         '%s %s %s, %s, data: "%s"',
441         self.__log_prefix__(),
442         rx_tx_prefix,
443         self.__get_message_name__(msg.get_service_id(), msg.get_data_id()),
444         self.__get_status_name__(msg.get_status()),
445         repr(msg.get_data())
446     )
447
448 def __data_available_callback__(self, comm_inst):
449     frame = comm_inst.receive()
450     msg = self.__analyse_frame__(frame)
451     if not self.__check_frame_checksum__(frame):
452         # Wrong Checksum
453         self.logger.log(self.__status_log_lvl__(STATUS_CHECKSUM_ERROR), '%s Received message
has an invalid checksum. Message will be ignored.', self.__log_prefix__())
454         return # No response needed
455     elif not self.check_authentication_state() and self.__authentication_required__(msg.
get_service_id(), msg.get_data_id()):
456         # Authentication required
457         self.__log_msg__(msg, 'RX <-')
458         if msg.get_service_id() in self.__sid_response_dict__.keys():
459             self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED), '%s
Authentication is required. Just sending negative response.', self.__log_prefix__())
460             status = STATUS_AUTH_REQUIRED
461             data = None
462         else:
463             self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED), '%s
Authentication is required. Incoming message will be ignored.', self.__log_prefix__())
464             return # No response needed

```

```

465         else:
466             # Valid message
467             self.__log_msg__(msg, 'RX <-')
468             callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id
469             ())
470             if msg.get_service_id() in self.__sid_response_dict__.keys():
471                 #
472                 # REQUEST RECEIVED
473                 #
474                 if callback is None:
475                     self.logger.warning("%s Incoming message with no registered callback.
476                     Sending negative response.", self.__log_prefix__())
477                     status = STATUS.BUFFERING.UNHANDLED_REQUEST
478                     data = None
479                 else:
480                     try:
481                         self.logger.debug("%s Executing callback %s to process received data",
482                         self.__log_prefix__(), callback.__name__)
483                         status, data = callback(msg, *args, **kwargs)
484                     except Exception as e:
485                         logger.error('{lp} Exception raised. Check callback {callback_name}: "{
486                         message}" and it\'s return values for service_id {service_id} and data_id {data_id}'.format(
487                         lp=self.__log_prefix__(), callback_name=callback.__name__, message=str(e), service_id=repr(
488                         msg.get_service_id()), data_id=repr(msg.get_data_id())))
489                         status = STATUS.CALLBACK.ERROR
490                         data = None
491             else:
492                 #
493                 # RESPONSE RECEIVED
494                 #
495                 if callback is None:
496                     self.__buffer_received_data__(msg)
497                 else:
498                     self.logger.debug("%s Executing callback %s to process received data", self.
499                     __log_prefix__(), callback.__name__)
500                     callback(msg, *args, **kwargs)
501                     return # No response needed
502                 self.send(self.__sid_response_dict__[msg.get_service_id()], msg.get_data_id(), data,
503                 status=status)
504
505     def __get_message_name__(self, service_id, data_id):
506         return 'service: %s, data_id: %s' % (
507             self.__sid_name_dict__.get(service_id, repr(service_id)),
508             self.__did_name_dict__.get(service_id, {}).get(data_id, repr(data_id)),
509         )
510
511     def __get_status_name__(self, status):
512         return 'status: %s' % (self.__status_name_dict__.get(status, 'unknown status: %s' % repr(
513         status)))
514
515     def __init_channel_name__(self, channel_name):
516         self.__comm_inst__.init_channel_name(channel_name)
517         self.__callbacks__.init_channel_name(channel_name)
518         if channel_name is None:
519             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.
520             DEFAULT_CHANNEL_NAME)
521         else:
522             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' +
523             channel_name)
524
525     def __log_prefix__(self):
526         return 'prot-client:' if self.__comm_inst__.IS_CLIENT else 'prot-server:'

```

```

516
517     def __mk_msg__(self, status, service_id, data_id, data):
518         return data_storage({data_storage.KEY_DATA.ID: data_id, data_storage.KEY_SERVICE.ID:
519                               service_id, data_storage.KEY_STATUS: status, data_storage.KEY_DATA: data})
520
521     def __status_log_lvl__(self, status):
522         return STATUS_LOG_LVL.get(status, logging.CRITICAL)
523
524     def add_data(self, service_id, data_id, name):
525         """
526         Method to add a name for a specific message.
527
528         :param service_id: The Service-ID of the message. See class definitions starting with ``
529         SID_``.
530         :type service_id: int or list of ints
531         :param data_id: The Data-ID of the message.
532         :type data_id: int
533         :param name: The Name for the transfered message.
534         :type name: str
535         """
536         try:
537             iter(service_id)
538         except Exception:
539             service_id = (service_id, )
540
541         for sid in service_id:
542             if sid not in self.__did_name_dict__:
543                 self.__did_name_dict__[sid] = {}
544             self.__did_name_dict__[sid][data_id] = name
545
546     def add_msg_to_auth_whitelist(self, service_id, data_id):
547         """
548         Method to add a specific message to the list, where no authentication is required.
549
550         :param service_id: The Service-ID of the message. See class definitions starting with ``
551         SID_``.
552         :type service_id: int
553         :param data_id: The Data-ID of the message.
554         :type data_id: int
555         """
556         if service_id not in self.__auth_whitelist__:
557             self.__auth_whitelist__[service_id] = []
558             self.__auth_whitelist__[service_id].append(data_id)
559             logger.debug('%s Adding Message(%s) to the authentication whitelist', self.
560                          __log_prefix__(), self.__get_message_name__(service_id, data_id))
561
562     def add_service(self, req_sid, resp_sid, req_name=None, resp_name=None):
563         """
564         Method to add a Service defined by Request- and Response Service-ID.
565
566         :param req_sid: The Request Service-ID.
567         :type req_sid: int
568         :param resp_sid: The Response Service-ID.
569         :type resp_sid: int
570         """
571         if req_sid in self.__sid_response_dict__:
572             logger.error('%s Service with Request-SID=%d and Response-SID=%d not added, because
573 request SID is already registered', self.__log_prefix__(), req_sid, resp_sid)
574             raise RequestSidExistsError("Request for this Service is already registered")
575         elif resp_sid in self.__sid_response_dict__.values():
576             logger.error('%s Service with Request-SID=%d and Response-SID=%d not added, because
577 response SID is already registered', self.__log_prefix__(), req_sid, resp_sid)
578             raise ResponseSidExistsError("Response for this Service is already registered")

```

```

573         else:
574             self.__sid_response_dict__[req_sid] = resp_sid
575             if req_name is not None:
576                 self.__sid_name_dict__[req_sid] = req_name
577             if resp_name is not None:
578                 self.__sid_name_dict__[resp_sid] = resp_name
579             logger.debug('%s Adding Service with Request=%s and Response=%s', self.__log_prefix__
580                          (), req_name or repr(req_sid), resp_name or repr(resp_sid))
581
582     def add_status(self, status, name):
583         """
584         Method to add a name for a status.
585
586         :param status: The Status. See class definitions starting with ``STATUS``.
587         :type status: int
588         :param name: The Name for the Status.
589         :type name: str
590         """
591         self.__status_name_dict__[status] = name
592
593     def authenticate(self, timeout=2):
594         """
595         This method authenticates the client at the server.
596
597         :param timeout: The timeout for the authentication (requesting seed, sending key and
598         getting authentication-feedback).
599         :type timeout: float
600         :returns: True, if authentication was successfull; False, if not.
601         :rtype: bool
602
603         .. note:: An authentication will only processed, if a secret had been given on
604         initialisation.
605
606         .. note:: Client and Server needs to use the same secret.
607         """
608         if self.__secret__ is not None:
609             self.__authentication_state__ = AUTH.STATE.SEED.REQUESTED
610             self.send(SID.AUTH.REQUEST, DID.AUTH.SEED, None)
611             cnt = 0
612             while cnt < timeout * 10:
613                 time.sleep(0.1)
614                 if self.__authentication_state__ == AUTH.STATE.TRUSTED.CONNECTION:
615                     return True
616                 elif self.__authentication_state__ == AUTH.STATE.UNTRUSTED.CONNECTION:
617                     break
618                 cnt += 1
619             return False
620
621     def check_authentication_state(self):
622         """
623         This Method return the Authitification State as boolean value.
624
625         :return: True, if authentication state is okay, otherwise False
626         :rtype: bool
627         """
628         return self.__secret__ is None or self.__authentication_state__ ==
629             AUTH.STATE.TRUSTED.CONNECTION
630
631     def connection_established(self):

```

```

628     """
629     This Method returns the Connection state including authentication as a boolean value.
630
631     :return: True, if the connection is established (incl. authentication, if a secret has
        been given)
632     :rtype: bool
633     """
634     return self.is_connected() and (self.__secret__ is None or self.
        check_authentication_state())
635
636 def is_connected(self):
637     """
638     This Methods returns Connection state of the Communication Instance :func:`comm_instance.
        is_connected`.
639
640     :return: True if the :class:`comm_instance` is connected, otherwise False..
641     :rtype: bool
642     """
643     return self.__comm_inst__.is_connected()
644
645 def receive(self, service_id, data_id, timeout=1):
646     """
647     This Method returns a message object for a defined message or None, if this message is
        not available after the given timeout.
648
649     :param service_id: The Service-ID for the message. See class definitions starting with ``
        SID_``.
650     :type service_id: int
651     :param data_id: The Data-ID for the message.
652     :type data_id: int
653     :param timeout: The timeout for receiving.
654     :type timeout: float
655     :returns: The received data storage object or None, if no data was received.
656     :rtype: data_storage
657     """
658     data = None
659     cnt = 0
660     while data is None and cnt < timeout * 10:
661         try:
662             data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
663         except IndexError:
664             data = None
665             cnt += 1
666             time.sleep(0.1)
667     if data is None and cnt >= timeout * 10:
668         self.logger.warning('"%s TIMEOUT (%ss): Requested data (service_id: %s; data_id: %s)
        not in buffer.', self.__log_prefix__, repr(timeout), repr(service_id), repr(data_id))
669     return data
670
671 def reconnect(self):
672     """
673     This methods initiates a reconnect by calling :func:`comm_instance.reconnect`.
674     """
675     return self.__comm_inst__.reconnect()
676
677 def register_callback(self, service_id, data_id, callback, *args, **kwargs):
678     """
679     This method registers a callback for the given parameters. Giving ``None`` means, that
        all Service-IDs or all Data-IDs are used.
680     If a message hitting these parameters has been received, the callback will be executed.
681
    
```

```

682         :param service_id: The Service-ID for the message. See class definitions starting with ``
SID_``.
683         :type service_id: int
684         :param data_id: The Data-ID for the message.
685         :type data_id: int
686
687         .. note:: The :func:`callback` is prioritised in the following order:
688
689             * Callbacks with defined Service-ID and Data-ID.
690             * Callbacks with a defined Service-ID and all Data-IDs.
691             * Callbacks with a defined Data-ID and all Service-IDs.
692             * Unspecific Callbacks.
693
694         .. note:: The :func:`callback` is executed with these arguments:
695
696             **Parameters given at the callback call:**
697
698             * The first Arguments is the received message as :class:`data_storage` object.
699             * Further arguments given at registration.
700             * Further keyword arguments given at registration.
701
702             **Return value of the callback:**
703
704             If the Callback is a Request Callback for a registered Service, the return value has
to be a tuple or list with
705
706             * :const:`response_status`: The response status (see class definitions starting with
:const:`STA_*`).
707             * :const:`response_data`: A JSON iterable object to be used as data for the response.
708
709         .. note:: Only registered services will respond via the callbacks return values with the
same data_id.
710         """
711         self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)
712
713     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2):
714         """
715         This methods sends out a message with the given content.
716
717         :param service_id: The Service-ID for the message. See class definitions starting with ``
SERVICE_``.
718         :type service_id: int
719         :param data_id: The Data-ID for the message.
720         :type data_id: int
721         :param data: The data to be transfered. The data needs to be json compatible.
722         :type data: str
723         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
724         :type status: int
725         :param timeout: The timeout for sending data (e.g. time to establish new connection).
726         :type timeout: float
727         :return: True if data had been sent, otherwise False.
728         :rtype: bool
729         """
730         if (self.check_authentication_state() or not self.__authentication_required__(
service_id, data_id)) or (service_id in self.__sid_response_dict__.values() and status ==
STATUS_AUTH_REQUIRED and data is None):
731             msg = data_storage(service_id=service_id, data_id=data_id, data=data, status=status)
732             self.__log_msg__(msg, 'TX ->')
733             return self.__comm_inst__.send(self.__build_frame__(msg), timeout=timeout)
734         else:
735             # Authentication required

```

```

736         self.logger.warning("%s Authentication is required. TX-Message %s, %s, data: %s
will be ignored.", self.__log_prefix__(), self.__get_message_name__(service_id, data_id),
self.__get_status_name__(status), repr(data))
737         return False
738
739
740 class struct_json_protocol(pure_json_protocol):
741     """
742     This Class has the same functionality like :class:`pure_json_protocol`. The message length is
less than for :class:`pure_json_protocol`, but the functionality and compatibility is
reduced.
743     See also parent :py:class:`pure_json_protocol`.
744
745     .. note::
746         This class is depreceated and here for compatibility reasons (to support old clients or
servers). Usage of :class:`pure_json_protocol` is recommended.
747     """
748     def __init__(self, *args, **kwargs):
749         pure_json_protocol.__init__(self, *args, **kwargs)
750
751     def __analyse_frame__(self, frame):
752         status, service_id, data_id = struct.unpack('>III', frame[0:12])
753         if sys.version_info >= (3, 0):
754             data = json.loads(frame[12:-1].decode('utf-8'))
755         else:
756             data = json.loads(frame[12:-1])
757         return self.__mk_msg__(status, service_id, data_id, data)
758
759     def __build_frame__(self, msg):
760         frame = struct.pack('>III', msg.get_status(), msg.get_service_id(), msg.get_data_id())
761         if sys.version_info >= (3, 0):
762             frame += bytes(json.dumps(msg.get_data()), 'utf-8')
763             frame += self.__calc_chksum__(frame)
764         else:
765             frame += json.dumps(msg.get_data())
766             frame += self.__calc_chksum__(frame)
767         return frame
768
769     def __calc_chksum__(self, raw_data):
770         chksum = 0
771         for b in raw_data:
772             if sys.version_info >= (3, 0):
773                 chksum ^= b
774             else:
775                 chksum ^= ord(b)
776         if sys.version_info >= (3, 0):
777             return bytes([chksum])
778         else:
779             return chr(chksum)
780
781     def __check_frame_checksum__(self, frame):
782         return self.__calc_chksum__(frame[:-1]) == frame[-1:]

```