Unittest for `socket_protocol`

January 14, 2021

Contents

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	ef6753738744ce1f4abee68df99df35c
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	19.392s

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	19.231s

2.3 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
`socket_protocol`	99.5%	100.0%
`socket_protocol.__init__.py`	99.5%	

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, 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**. 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:	2021-01-14 00:25:50,831
Finished-Time:	2021-01-14 00:25:50,832
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:	2021-01-14 00:26:12,121
Finished-Time:	2021-01-14 00:26:12,122
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/unittest/socket_protocol/unittest/src/tests/__init__.py (27)
Start-Time:	2021-01-14 00:25:50,832
Finished-Time:	2021-01-14 00:25:50,832
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/unittest/socket_protocol/unittest/src/tests/__init__.py (27)
Start-Time:	2021-01-14 00:26:12,122
Finished-Time:	2021-01-14 00:26:12,122

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/unittest/socket_protocol/unittest/src/tests/__init__.py (28)
Start-Time:	2021-01-14 00:25:50,833
Finished-Time:	2021-01-14 00:25:50,833
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/unittest/socket_protocol/unittest/src/tests/__init__.py (28)
Start-Time:	2021-01-14 00:26:12,122

Finished-Time:	2021-01-14 00:26:12,123
Time-Consumption	0.000s

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:	2021-01-14 00:25:50,833
Finished-Time:	2021-01-14 00:25:50,834
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:	2021-01-14 00:26:12,123
Finished-Time:	2021-01-14 00:26:12,124
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:	2021-01-14 00:25:50,834
Finished-Time:	2021-01-14 00:25:51,588
Time-Consumption	0.754s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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:	2021-01-14 00:26:12,124
Finished-Time:	2021-01-14 00:26:12,877
Time-Consumption	0.753s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).

3.2.2 A checksumm shall ensure the correct transmition

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**. 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:	2021-01-14 00:25:51,588
Finished-Time:	2021-01-14 00:25:52,553
Time-Consumption	0.965s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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:	2021-01-14 00:26:12,877
Finished-Time:	2021-01-14 00:26:13,839
Time-Consumption	0.962s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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 authentification between server and client shall be possible including status feedback methods

Description

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

Reason for the implementation

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

Fitcriterion

Check authentification method feedback (client) and authentification 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:	2021-01-14 00:25:52,553
Finished-Time:	2021-01-14 00:25:53,628
Time-Consumption	1.075s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	No secret set
Info	Performing Authentification
Success	Return Value of authentification method is correct (Content False and Type is <type 'bool'>).
Success	Authentification state of server is correct (Content True and Type is <type 'bool'>).
Success	Authentification state of client is correct (Content True and Type is <type 'bool'>).
Info	Different secrets set
Success	Authentification state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentification state of client is correct (Content False and Type is <type 'bool'>).
Info	Performing Authentification
Success	Return Value of authentification method is correct (Content False and Type is <type 'bool'>).
Success	Authentification state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentification state of client is correct (Content False and Type is <type 'bool'>).
Info	Identical secrets set
Info	Performing Authentification
Success	Return Value of authentification method is correct (Content True and Type is <type 'bool'>).
Success	Authentification state of server is correct (Content True and Type is <type 'bool'>).
Success	Authentification state of client is correct (Content True and Type is <type 'bool'>).
Info	Corrupting the authentification mechanism
Info	Performing Authentification
Success	Return Value of authentification method is correct (Content False and Type is <type 'bool'>).
Success	Authentification state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentification 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:	2021-01-14 00:26:13,840
Finished-Time:	2021-01-14 00:26:14,808
Time-Consumption	0.969s

Testsummary:

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

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

3.2.4 An automatic authentification shall available

Description

An authentification is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentification.

Fitcriterion

Check authentification 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:	2021-01-14 00:25:53,629
Finished-Time:	2021-01-14 00:25:56,338
Time-Consumption	2.710s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set and automatic authentification
Success	Authentification state of server is correct (Content False and Type is <type 'bool'>).
Success	Authentification state of client is correct (Content False and Type is <type 'bool'>).
Info	Connecting Server and Client
Success	Authentification state of server is correct (Content True and Type is <type 'bool'>).
Success	Authentification 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/unittest/socket_protocol/unittest/src/tests/__init__.py (36)
Start-Time:	2021-01-14 00:26:14,810
Finished-Time:	2021-01-14 00:26:17,520
Time-Consumption	2.710s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set and automatic authentification
Success	Authentification state of server is correct (Content False and Type is <class 'bool'>).
Success	Authentification state of client is correct (Content False and Type is <class 'bool'>).
Info	Connecting Server and Client
Success	Authentification state of server is correct (Content True and Type is <class 'bool'>).
Success	Authentification 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 authentification 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 Authentification 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 authentification 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/unittest/socket_protocol/unittest/src/tests/__init__.py (37)
Start-Time:	2021-01-14 00:25:56,339
Finished-Time:	2021-01-14 00:25:58,318
Time-Consumption	1.979s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Setting a Server secret and no Client secret
Info	Transfering 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	Transfering 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	Transfering 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	Transfering 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 Authentification
Info	Transfering 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	Transfering 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.1.9!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (37)
Start-Time:	2021-01-14 00:26:17,520
Finished-Time:	2021-01-14 00:26:19,493
Time-Consumption	1.972s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting a Server secret and no Client secret
Info	Transfering 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	Transfering 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	Transfering 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	Transfering 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 Authentification
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} 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 authentification was successfull performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentification 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/unittest/socket_protocol/unittest/src/tests/__init__.py (38)
Start-Time:	2021-01-14 00:25:58,318
Finished-Time:	2021-01-14 00:26:00,802
Time-Consumption	2.483s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set
Info	Transfering 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	Transfering 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	Transfering 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	Transfering 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	Transfering 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	Transfering 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	Transfering 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	Transfering 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.1.10!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (38)
Start-Time:	2021-01-14 00:26:19,494
Finished-Time:	2021-01-14 00:26:21,974
Time-Consumption	2.480s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set
Info	Transfering 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	Transfering 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	Transfering 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	Transfering 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	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} 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:	2021-01-14 00:26:00,802
Finished-Time:	2021-01-14 00:26:02,552

Time-Consumption	1.750s

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/unittest/socket_protocol/unittest/src/tests/__init__.py (39)
Start-Time:	2021-01-14 00:26:21,975
Finished-Time:	2021-01-14 00:26:23,721
Time-Consumption	1.746s

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/unittest/socket_protocol/unittest/src/tests/__init__.py (40)
Start-Time:	2021-01-14 00:26:02,553
Finished-Time:	2021-01-14 00:26:03,414
Time-Consumption	0.861s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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/unittest/socket_protocol/unittest/src/tests/__init__.py (40)
Start-Time:	2021-01-14 00:26:23,721
Finished-Time:	2021-01-14 00:26:24,580
Time-Consumption	0.859s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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/unittest/socket_protocol/unittest/src/tests/__init__.py (41)
Start-Time:	2021-01-14 00:26:03,415
Finished-Time:	2021-01-14 00:26:03,768
Time-Consumption	0.353s

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/unittest/socket_protocol/unittest/src/tests/__init__.py (41)
Start-Time:	2021-01-14 00:26:24,581
Finished-Time:	2021-01-14 00:26:24,937
Time-Consumption	0.356s
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/unittest/socket_protocol/unittest/src/tests/__init__.py (45)
Start-Time:	2021-01-14 00:26:03,769
Finished-Time:	2021-01-14 00:26:04,735
Time-Consumption	0.966s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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	Transfering 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	Transfering 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:	2021-01-14 00:26:24,937
Finished-Time:	2021-01-14 00:26:25,902
Time-Consumption	0.965s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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	Transfering 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	Transfering 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:	2021-01-14 00:26:04,735
Finished-Time:	2021-01-14 00:26:05,295
Time-Consumption	0.560s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback

Info	Transfering 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:	2021-01-14 00:26:25,902
Finished-Time:	2021-01-14 00:26:26,459
Time-Consumption	0.557s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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:	2021-01-14 00:26:05,296
Finished-Time:	2021-01-14 00:26:05,855
Time-Consumption	0.560s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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:	2021-01-14 00:26:26,459
Finished-Time:	2021-01-14 00:26:27,014
Time-Consumption	0.554s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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 incomming 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:	2021-01-14 00:26:05,856
Finished-Time:	2021-01-14 00:26:06,413
Time-Consumption	0.557s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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:	2021-01-14 00:26:27,014
Finished-Time:	2021-01-14 00:26:27,569
Time-Consumption	0.555s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transfering 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:	2021-01-14 00:26:06,414
Finished-Time:	2021-01-14 00:26:07,587
Time-Consumption	1.173s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transfering 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	Removing Callback for a specific Data- and Service-ID
Info	Transfering 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’: 6, u’service_id’: 11, u’data’: 34, u’data_id’: 0} and Type is <class ’socket_protocol.data_storage’>).
Info	Removing Callback for a specific Service-ID and all Data-IDs
Info	Transfering 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’: 6, u’service_id’: 11, u’data’: 35, u’data_id’: 0} and Type is <class ’socket_protocol.data_storage’>).
Info	Removing Callback for a specific Data-ID and all Serice-IDs
Info	Transfering 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’: 36, 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.1.18!

Testrun:	python 3.8.5 (final)
Caller:	/user_data/data/dirk/prj/unittest/socket_protocol/unittest/src/tests/__init__.py (49)
Start-Time:	2021-01-14 00:26:27,570
Finished-Time:	2021-01-14 00:26:28,738
Time-Consumption	1.169s

Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering all kind of Callbacks
Info	Transfering 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	Transfering 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	Transfering 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 Serice-IDs
Info	Transfering 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:	2021-01-14 00:26:07,588
Finished-Time:	2021-01-14 00:26:08,403
Time-Consumption	0.816s

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 authentification
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:	2021-01-14 00:26:28,739
Finished-Time:	2021-01-14 00:26:29,546
Time-Consumption	0.807s

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 authentification
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/unittest/socket_protocol/unittest/src/tests/__init__.py (54)
Start-Time:	2021-01-14 00:26:08,404
Finished-Time:	2021-01-14 00:26:08,764
Time-Consumption	0.360s

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/unittest/socket_protocol/unittest/src/tests/__init__.py (54)
Start-Time:	2021-01-14 00:26:29,546
Finished-Time:	2021-01-14 00:26:29,903
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/unittest/socket_protocol/unittest/src/tests/__init__.py (55)
Start-Time:	2021-01-14 00:26:08,764
Finished-Time:	2021-01-14 00:26:09,469
Time-Consumption	0.705s

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/unittest/socket_protocol/unittest/src/tests/__init__.py (55)
Start-Time:	2021-01-14 00:26:29,903
Finished-Time:	2021-01-14 00:26:30,603
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:	2021-01-14 00:26:09,470
Finished-Time:	2021-01-14 00:26:10,235
Time-Consumption	0.764s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Transfering 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	Transfering 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:	2021-01-14 00:26:30,604
Finished-Time:	2021-01-14 00:26:31,362
Time-Consumption	0.759s
Testsummary:	
Info	Setting up communication

Info	Connecting Server and Client
Info	Transfering 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_transfered'} and Type is <class 'socket_protocol.data_storage'>).
Info	Transfering 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_transfered'} 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, 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: {'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=authentification request and
↪  Response=authentification response
```

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

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

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

```
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
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_transfered', 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_transfered', 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_transfered', 'data_id': 34} (<class
↪ 'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
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_transfered'"
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_transfered', 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_transfered', 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_transfered', 'data_id': 35} (<class
↪  'socket_protocol.data_storage'>)
```

A.1.6 A checksumm shall ensure the correct transmition

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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
```

```
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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 Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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 authentification between server and client shall be possible including status feedback methods

Description

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

Reason for the implementation

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

Fitcriterion

Check authentification method feedback (client) and authentification 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 incomming connection
```

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

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

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

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

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

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

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

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

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

```
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Authentification

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

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

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

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

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

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

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

Info Different secrets set

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

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

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

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

Info Performing Authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'a7297103014445de4fed37f28d74079232166883ca7da2b50d2d5bac6c494eee'"
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 61 37 32 39 37 31 30 33
↪  30 31 34 34 34 35 64 65 34 66 65
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 61 37 32 39 37 31 30 33
↪  30 31 34 34 34 35 64 65 34 66 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-server: TX -> (64): 64 33 37 66 32 38 64 37 34 30 37 39 32 33 32 31 36 36 38 38 33 63 61
↪  37 64 61 32 62 35 30 64 32 64 35 62 61 63 36 63 34 39 34 65 65 65 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d 4c 2a
comm-client: RX <- (64): 64 33 37 66 32 38 64 37 34 30 37 39 32 33 32 31 36 36 38 38 33 63 61
↪  37 64 61 32 62 35 30 64 32 64 35 62 61 63 36 63 34 39 34 65 65 65 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d 4c 2a
STP: data sync (3a) received => changing state STP_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): 29 94 3a 3e
comm-client: RX <- (4): 29 94 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 61 37 32 39 37 31 30 33 30 31
↪  34 34 34 35 64 65 34 66 65 64 33 37 66 32 38 64 37 34 30 37 39 32 33 32 31 36 36 38 38 33
↪  63 61 37 64 61 32 62 35 30 64 32 64 35 62 61 63 36 63 34 39 34 65 65 65 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d 4c 2a 29 94
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "u'a7297103014445de4fed37f28d74079232166883ca7da2b50d2d5bac6c494eee'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'58b04824dff37282c1dc3d1c1dccccc213500146431569e6fb23272b89a7891ec218701125b5468b90c3c83⌋
↪  91482cdb7cec3a5b820a4f4510e0b3973d1e60989'"
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 35 38 62 30 34 38 32 34
↪  64 66 66 33 37 32 38 32 63 31 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 30 2c 20 22 64 61 74 61 22 3a 3d 20 22 35 38 62 30 34 38 32 34
↪  64 66 66 33 37 32 38 32 63 31 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 -> (64): 63 33 64 31 63 31 64 63 63 63 63 63 32 31 33 35 30 30 31 34 36 34 33
↪  31 35 36 39 65 36 66 62 32 33 32 37 32 62 38 39 61 37 38 39 31 65 63 32 31 38 37 30 31 31
↪  32 35 62 35 34 36 38 62 39 30 63
comm-server: RX <- (64): 63 33 64 31 63 31 64 63 63 63 63 63 32 31 33 35 30 30 31 34 36 34 33
↪  31 35 36 39 65 36 66 62 32 33 32 37 32 62 38 39 61 37 38 39 31 65 63 32 31 38 37 30 31 31
↪  32 35 62 35 34 36 38 62 39 30 63
comm-client: TX -> (64): 33 63 38 33 39 31 34 38 32 63 64 62 37 63 65 63 33 61 35 62 38 32 30
↪  61 34 66 34 35 31 30 65 30 62 33 39 37 33 64 31 65 36 30 39 38 39 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 1a 9b
comm-server: RX <- (64): 33 63 38 33 39 31 34 38 32 63 64 62 37 63 65 63 33 61 35 62 38 32 30
↪  61 34 66 34 35 31 30 65 30 62 33 39 37 33 64 31 65 36 30 39 38 39 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 1a 9b
STP: data sync (3a) received => changing state STP_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): a8 57 3a 3e
comm-server: RX <- (4): a8 57 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 35 38 62 30 34 38 32 34 64 66
↪  66 33 37 32 38 32 63 31 64 63 33 64 31 63 31 64 63 63 63 63 63 32 31 33 35 30 30 31 34 36
↪  34 33 31 35 36 39 65 36 66 62 32 33 32 37 32 62 38 39 61 37 38 39 31 65 63 32 31 38 37 30
↪  31 31 32 35 62 35 34 36 38 62 39 30 63 33 63 38 33 39 31 34 38 32 63 64 62 37 63 65 63 33
↪  61 35 62 38 32 30 61 34 66 34 35 31 30 65 30 62 33 39 37 33 64 31 65 36 30 39 38 39 22 2c
↪  20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 1a 9b a8 57
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "u'58b04824dff37282c1dc3d1c1dccccc213500146431569e6fb23272b89a7891ec218701125b5468b90c3c8⌟
↪  391482cdb7cec3a5b820a4f4510e0b3973d1e60989'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "False"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got negative authentification feedback
```

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

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

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

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

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

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

Info Identical secrets set

Info Performing Authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'7d481b5970b38472dc2791845810a58d0ca6bc63c3e4757fcbe7db4cc70ae5da'"
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 37 64 34 38 31 62 35 39
↪  37 30 62 33 38 34 37 32 64 63 32
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 37 64 34 38 31 62 35 39
↪  37 30 62 33 38 34 37 32 64 63 32
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): 37 39 31 38 34 35 38 31 30 61 35 38 64 30 63 61 36 62 63 36 33 63 33
↪  65 34 37 35 37 66 63 62 65 37 64 62 34 63 63 37 30 61 65 35 64 61 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d ff db
```

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

```
STP: data sync (3a) received => changing state STP_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): d2 d9 3a 3e
```

```
comm-client: RX <- (4): d2 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 37 64 34 38 31 62 35 39 37 30
↪  62 33 38 34 37 32 64 63 32 37 39 31 38 34 35 38 31 30 61 35 38 64 30 63 61 36 62 63 36 33
↪  63 33 65 34 37 35 37 66 63 62 65 37 64 62 34 63 63 37 30 61 65 35 64 61 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d ff db d2 d9
```

```
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "u'7d481b5970b38472dc2791845810a58d0ca6bc63c3e4757fcbe7db4cc70ae5da'"
```

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

```
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'4fba2e0f52793e9f458824ba0c9064e6395c6d6a41e823969afedc1879a142673ca63122106f508e2d268fa↲
↪  0d92ea49e8c78b59a37632c75e041e34557183725'"
```

```
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 34 66 62 61 32 65 30 66
↪  35 32 37 39 33 65 39 66 34 35 38
```

```
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 34 66 62 61 32 65 30 66
↪  35 32 37 39 33 65 39 66 34 35 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
```

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

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

```
comm-client: TX -> (64): 36 38 66 61 30 64 39 32 65 61 34 39 65 38 63 37 38 62 35 39 61 33 37
↪  36 33 32 63 37 35 65 30 34 31 65 33 34 35 35 37 31 38 33 37 32 35 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 54 a7
```

```
comm-server: RX <- (64): 36 38 66 61 30 64 39 32 65 61 34 39 65 38 63 37 38 62 35 39 61 33 37
↪  36 33 32 63 37 35 65 30 34 31 65 33 34 35 35 37 31 38 33 37 32 35 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 54 a7
```

```
STP: data sync (3a) received => changing state STP_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): a3 5f 3a 3e
```

```
comm-server: RX <- (4): a3 5f 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 34 66 62 61 32 65 30 66 35 32
↪  37 39 33 65 39 66 34 35 38 38 32 34 62 61 30 63 39 30 36 34 65 36 33 39 35 63 36 64 36 61
↪  34 31 65 38 32 33 39 36 39 61 66 65 64 63 31 38 37 39 61 31 34 32 36 37 33 63 61 36 33 31
↪  32 32 31 30 36 66 35 30 38 65 32 64 32 36 38 66 61 30 64 39 32 65 61 34 39 65 38 63 37 38
↪  62 35 39 61 33 37 36 33 32 63 37 35 65 30 34 31 65 33 34 35 35 37 31 38 33 37 32 35 22 2c
↪  20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 54 a7 a3 5f
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "u'4fba2e0f52793e9f458824ba0c9064e6395c6d6a41e823969afedc1879a142673ca63122106f508e2d268f ⌋
↪  a0d92ea49e8c78b59a37632c75e041e34557183725'"
```

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

```
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

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

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

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

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

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

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

Info Corrupting the authentification mechanism

```
prot-server: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Info Performing Authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
```

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

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

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

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

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

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

A.1.8 An automatic authentification shall available

Description

An authentification is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentification.

Fitcriterion

Check authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification

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

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

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

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

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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: authentification 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: authentification request, data_id: seed, status: okay, data:
↪ "None"
prot-server: Executing callback __authentificate_create_seed__ 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 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
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪ "'bf329bbd77793fb1bc3649da46b75eea905e42b069b668a521c1e4840e565078'"
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 62 66 33 32 39 62 62 64
↪  37 37 37 39 33 66 62 31 62 63 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 62 66 33 32 39 62 62 64
↪  37 37 37 39 33 66 62 31 62 63 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): 36 34 39 64 61 34 36 62 37 35 65 65 61 39 30 35 65 34 32 62 30 36 39
↪  62 36 36 38 61 35 32 31 63 31 65 34 38 34 30 65 35 36 35 30 37 38 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d 53 32
comm-client: RX <- (64): 36 34 39 64 61 34 36 62 37 35 65 65 61 39 30 35 65 34 32 62 30 36 39
↪  62 36 36 38 61 35 32 31 63 31 65 34 38 34 30 65 35 36 35 30 37 38 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d 53 32
STP: data sync (3a) received => changing state STP_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): d6 9b 3a 3e
comm-client: RX <- (4): d6 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 - (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 62 66 33 32 39 62 62 64 37 37
↪  37 39 33 66 62 31 62 63 33 36 34 39 64 61 34 36 62 37 35 65 65 61 39 30 35 65 34 32 62 30
↪  36 39 62 36 36 38 61 35 32 31 63 31 65 34 38 34 30 65 35 36 35 30 37 38 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d 53 32 d6 9b
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "u'bf329bbd77793fb1bc3649da46b75eea905e42b069b668a521c1e4840e565078'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'d07b71e43499a4d36bbfc6be15bb79ad03d84efaf1057d576102661f164748d3c757b6f6ecafb28862f8a14⌋
↪  23d629edf1f50b49778de6ed8d8ecc2e45c44ea8a'"
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 64 30 37 62 37 31 65 34
↪  33 34 39 39 61 34 64 33 36 62 62
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 64 30 37 62 37 31 65 34
↪  33 34 39 39 61 34 64 33 36 62 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
comm-client: TX -> (64): 66 63 36 62 65 31 35 62 62 37 39 61 64 30 33 64 38 34 65 66 61 66 31
↪  30 35 37 64 35 37 36 31 30 32 36 36 31 66 31 36 34 37 34 38 64 33 63 37 35 37 62 36 66 36
↪  65 63 61 66 62 32 38 38 36 32 66
comm-server: RX <- (64): 66 63 36 62 65 31 35 62 62 37 39 61 64 30 33 64 38 34 65 66 61 66 31
↪  30 35 37 64 35 37 36 31 30 32 36 36 31 66 31 36 34 37 34 38 64 33 63 37 35 37 62 36 66 36
↪  65 63 61 66 62 32 38 38 36 32 66
comm-client: TX -> (64): 38 61 31 34 32 33 64 36 32 39 65 64 66 31 66 35 30 62 34 39 37 37 38
↪  64 65 36 65 64 38 64 38 65 63 63 32 65 34 35 63 34 34 65 61 38 61 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 37 c3
comm-server: RX <- (64): 38 61 31 34 32 33 64 36 32 39 65 64 66 31 66 35 30 62 34 39 37 37 38
↪  64 65 36 65 64 38 64 38 65 63 63 32 65 34 35 63 34 34 65 61 38 61 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 37 c3
STP: data sync (3a) received => changing state STP_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): 51 9c 3a 3e
comm-server: RX <- (4): 51 9c 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 64 30 37 62 37 31 65 34 33 34
↪  39 39 61 34 64 33 36 62 62 66 63 36 62 65 31 35 62 62 37 39 61 64 30 33 64 38 34 65 66 61
↪  66 31 30 35 37 64 35 37 36 31 30 32 36 36 31 66 31 36 34 37 34 38 64 33 63 37 35 37 62 36
↪  66 36 65 63 61 66 62 32 38 38 36 32 66 38 61 31 34 32 33 64 36 32 39 65 64 66 31 66 35 30
↪  62 34 39 37 37 38 64 65 36 65 64 38 64 38 65 63 63 32 65 34 35 63 34 34 65 61 38 61 22 2c
↪  20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 37 c3 51 9c
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "u'd07b71e43499a4d36bbfc6be15bb79ad03d84efaf1057d576102661f164748d3c757b6f6ecafb28862f8a1⌟
↪  423d629edf1f50b49778de6ed8d8ecc2e45c44ea8a'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

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

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

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

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

A.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentification 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 Authentification 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 authentification 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=authentification request and
↪  Response=authentification response
```

```
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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
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
prot-server: Cleaning up receive-buffer
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 a Server secret and no Client secret

Info Transfering a message client → server

```
prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
↪  "'msg3_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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 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: authentification 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 Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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 Transfering a message client → server

```
prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↪  data: 'msg1_data_to_be_transfered' 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 Transfering 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_transfered' 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 Authentification

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

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

```
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'0e1603ea8f893c447bda8cb4ac9dccc8fdff71bf4124ab36d57e42f1ece1fea0'"
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 65 31 36 30 33 65 61
↪  38 66 38 39 33 63 34 34 37 62 64
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 65 31 36 30 33 65 61
↪  38 66 38 39 33 63 34 34 37 62 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-server: TX -> (64): 61 38 63 62 34 61 63 39 64 63 63 63 38 66 64 66 66 37 31 62 66 34 31
↪  32 34 61 62 33 36 64 35 37 65 34 32 66 31 65 63 65 31 66 65 61 30 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d c3 5a
comm-client: RX <- (64): 61 38 63 62 34 61 63 39 64 63 63 63 38 66 64 66 66 37 31 62 66 34 31
↪  32 34 61 62 33 36 64 35 37 65 34 32 66 31 65 63 65 31 66 65 61 30 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d c3 5a
STP: data sync (3a) received => changing state STP_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): 1d 65 3a 3e
comm-client: RX <- (4): 1d 65 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 65 31 36 30 33 65 61 38 66
↪  38 39 33 63 34 34 37 62 64 61 38 63 62 34 61 63 39 64 63 63 63 38 66 64 66 66 37 31 62 66
↪  34 31 32 34 61 62 33 36 64 35 37 65 34 32 66 31 65 63 65 31 66 65 61 30 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d c3 5a 1d 65
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "u'0e1603ea8f893c447bda8cb4ac9dccc8fdff71bf4124ab36d57e42f1ece1fea0'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'573df7dec49c17a589a829082385e72a44be9d1d151859d51d4ca44eb095c6469a8ec5cce6194833d4ed873⌋
↪  351f960c08ea4dbe9366d0ed03fd9a9f2a7304544'"
```

```
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 35 37 33 64 66 37 64 65
↪  63 34 39 63 31 37 61 35 38 39 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 35 37 33 64 66 37 64 65
↪  63 34 39 63 31 37 61 35 38 39 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): 38 32 39 30 38 32 33 38 35 65 37 32 61 34 34 62 65 39 64 31 64 31 35
↪  31 38 35 39 64 35 31 64 34 63 61 34 34 65 62 30 39 35 63 36 34 36 39 61 38 65 63 35 63 63
↪  65 36 31 39 34 38 33 33 64 34 65
```

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

```
comm-client: TX -> (64): 64 38 37 33 33 35 31 66 39 36 30 63 30 38 65 61 34 64 62 65 39 33 36
↪  36 64 30 65 64 30 33 66 64 39 61 39 66 32 61 37 33 30 34 35 34 34 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d e2 d3
```

```
comm-server: RX <- (64): 64 38 37 33 33 35 31 66 39 36 30 63 30 38 65 61 34 64 62 65 39 33 36
↪  36 64 30 65 64 30 33 66 64 39 61 39 66 32 61 37 33 30 34 35 34 34 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d e2 d3
```

```
STP: data sync (3a) received => changing state STP_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): 6a 9c 3a 3e
```

```
comm-server: RX <- (4): 6a 9c 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 35 37 33 64 66 37 64 65 63 34
↪  39 63 31 37 61 35 38 39 61 38 32 39 30 38 32 33 38 35 65 37 32 61 34 34 62 65 39 64 31 64
↪  31 35 31 38 35 39 64 35 31 64 34 63 61 34 34 65 62 30 39 35 63 36 34 36 39 61 38 65 63 35
↪  63 63 65 36 31 39 34 38 33 33 64 34 65 64 38 37 33 33 35 31 66 39 36 30 63 30 38 65 61 34
↪  64 62 65 39 33 36 36 64 30 65 64 30 33 66 64 39 61 39 66 32 61 37 33 30 34 35 34 34 22 2c
↪  20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d e2 d3 6a 9c
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "u'573df7dec49c17a589a829082385e72a44be9d1d151859d51d4ca44eb095c6469a8ec5cce6194833d4ed87⌟
↪  3351f960c08ea4dbe9366d0ed03fd9a9f2a7304544'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
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_transfered', 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_transfered', 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_transfered', 'data_id': 34} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered', 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_transfered', 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_transfered', '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 authentification was successfull performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentification 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**.

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=authentification request and
↪ Response=authentification response
```

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

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

```
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↪  data: 'msg1_data_to_be_transfered' 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 Transfering 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_transfered' 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 authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
prot-server: Authentification 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 Transfering 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_transfered' 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 authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
```

```
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_transfered', 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_transfered', 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_transfered', 'data_id': 34} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering 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_transfered' 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 authentification whitelist
```

```
prot-server: Adding Message (service: 17, data_id: 35) to the authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered', 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_transfered', 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_transfered', 'data_id': 34} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered', 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_transfered', 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_transfered', '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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
```

```
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
```

```
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
```

```
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 Transfering a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
prot-server: Incomming 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪ authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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=authentification request and
↪ Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
```

```
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
```

```
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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	Transfering 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 Transfering 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: read data request, 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 Transfering 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: 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 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=None
```

Info Transfering 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪ authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering 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 incomming 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 incomming connection
```

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

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

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

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

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

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

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

```
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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	Transfering 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.1.18 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
```

```
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪ authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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=authentification request and
↪ Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering 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 Transfering 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 Transfering 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 Transfering 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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 authentification

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

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

```
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'5d3e3e3ae297c94184ace5320bbad8523b8752006324a312c78d0002ee4b157e'"
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 64 33 65 33 65 33 61
↪  65 32 39 37 63 39 34 31 38 34 61
```

```
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 64 33 65 33 65 33 61
↪  65 32 39 37 63 39 34 31 38 34 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-server: TX -> (64): 63 65 35 33 32 30 62 62 61 64 38 35 32 33 62 38 37 35 32 30 30 36 33
↪  32 34 61 33 31 32 63 37 38 64 30 30 30 32 65 65 34 62 31 35 37 65 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d 05 6d
comm-client: RX <- (64): 63 65 35 33 32 30 62 62 61 64 38 35 32 33 62 38 37 35 32 30 30 36 33
↪  32 34 61 33 31 32 63 37 38 64 30 30 30 32 65 65 34 62 31 35 37 65 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 30 7d 05 6d
STP: data sync (3a) received => changing state STP_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): 6a 07 3a 3e
comm-client: RX <- (4): 6a 07 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 64 33 65 33 65 33 61 65 32
↪  39 37 63 39 34 31 38 34 61 63 65 35 33 32 30 62 62 61 64 38 35 32 33 62 38 37 35 32 30 30
↪  36 33 32 34 61 33 31 32 63 37 38 64 30 30 30 32 65 65 34 62 31 35 37 65 22 2c 20 22 64 61
↪  74 61 5f 69 64 22 3a 20 30 7d 05 6d 6a 07
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "u'5d3e3e3ae297c94184ace5320bbad8523b8752006324a312c78d0002ee4b157e'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'d8212054b96be9ce14c4d528e5c63407bedf9fffe731306e960a7e64b28d8fabbdee29042c95f4f7a6d34d0⌟
↪  1f22d3ab1b2a0e27a50f7456e6bd39320cc7488f5'"
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 64 38 32 31 32 30 35 34
↪  62 39 36 62 65 39 63 65 31 34 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 64 38 32 31 32 30 35 34
↪  62 39 36 62 65 39 63 65 31 34 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): 34 64 35 32 38 65 35 63 36 33 34 30 37 62 65 64 66 39 66 66 66 65 37
↪  33 31 33 30 36 65 39 36 30 61 37 65 36 34 62 32 38 64 38 66 61 62 62 64 65 65 32 39 30 34
↪  32 63 39 35 66 34 66 37 61 36 64
comm-server: RX <- (64): 34 64 35 32 38 65 35 63 36 33 34 30 37 62 65 64 66 39 66 66 66 65 37
↪  33 31 33 30 36 65 39 36 30 61 37 65 36 34 62 32 38 64 38 66 61 62 62 64 65 65 32 39 30 34
↪  32 63 39 35 66 34 66 37 61 36 64
comm-client: TX -> (64): 33 34 64 30 31 66 32 32 64 33 61 62 31 62 32 61 30 65 32 37 61 35 30
↪  66 37 34 35 36 65 36 62 64 33 39 33 32 30 63 63 37 34 38 38 66 35 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 4c 6a
comm-server: RX <- (64): 33 34 64 30 31 66 32 32 64 33 61 62 31 62 32 61 30 65 32 37 61 35 30
↪  66 37 34 35 36 65 36 62 64 33 39 33 32 30 63 63 37 34 38 38 66 35 22 2c 20 22 64 61 74 61
↪  5f 69 64 22 3a 3d 20 31 7d 4c 6a
STP: data sync (3a) received => changing state STP_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): f4 ef 3a 3e
comm-server: RX <- (4): f4 ef 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 64 38 32 31 32 30 35 34 62 39
↪  36 62 65 39 63 65 31 34 63 34 64 35 32 38 65 35 63 36 33 34 30 37 62 65 64 66 39 66 66 66
↪  65 37 33 31 33 30 36 65 39 36 30 61 37 65 36 34 62 32 38 64 38 66 61 62 62 64 65 65 32 39
↪  30 34 32 63 39 35 66 34 66 37 61 36 64 33 34 64 30 31 66 32 32 64 33 61 62 31 62 32 61 30
↪  65 32 37 61 35 30 66 37 34 35 36 65 36 62 64 33 39 33 32 30 63 63 37 34 38 38 66 35 22 2c
↪  20 22 64 61 74 61 5f 69 64 22 3a 20 31 7d 4c 6a f4 ef
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "u'd8212054b96be9ce14c4d528e5c63407bedf9fffe731306e960a7e64b28d8fabbdee29042c95f4f7a6d34d⌋
↪  01f22d3ab1b2a0e27a50f7456e6bd39320cc7488f5'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
```

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

```
prot-client: Got positive authentification 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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
```

```
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪ authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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=authentification request and
↪ Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
```

```
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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:
↪  "u'msg1_data_to_be_transfered'"
```

```
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_transfered', '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_transfered', '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_transfered', 'data_id': 34} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered', '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_transfered', '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_transfered', '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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info	Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

B.1.6 A checksumm shall ensure the correct transmition

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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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 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 Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
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_transfered'"
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 authentification between server and client shall be possible including status feedback methods

Description

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

Reason for the implementation

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

Fitcriterion

Check authentification method feedback (client) and authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪ Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪ authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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 Authentification

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

```
Result (Return Value of authentification method): False (<class 'bool'>)
Expectation (Return Value of authentification method): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of server): True (<class 'bool'>)
Expectation (Authentification state of server): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of client): True (<class 'bool'>)
Expectation (Authentification state of client): result = True (<class 'bool'>)
```

Info Different secrets set

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
```

```
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

Info Performing Authentification

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

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

```
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'fe8dc24eafa66c1497558342bdab3b316547584dbb6ed6a41baf127fe490309c'"
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 65 38 64
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 65 38 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
STP: data sync (3a) received => changing state STP_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): 63 32 34 65 61 66 61 36 36 63 31 34 39 37 35 35 38 33 34 32 62 64 61
↪  62 33 62 33 31 36 35 34 37 35 38 34 64 62 62 36 65 64 36 61 34 31 62 61 66 31 32 37 66 65
↪  34 39 30 33 30 39 63 22 7d 24 7d
comm-client: RX <- (64): 63 32 34 65 61 66 61 36 36 63 31 34 39 37 35 35 38 33 34 32 62 64 61
↪  62 33 62 33 31 36 35 34 37 35 38 34 64 62 62 36 65 64 36 61 34 31 62 61 66 31 32 37 66 65
↪  34 39 30 33 30 39 63 22 7d 24 7d
comm-server: TX -> (4): 17 7c 3a 3e
comm-client: RX <- (4): 17 7c 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 65 38 64 63 32 34 65 61 66 61 36 36 63 31 34 39 37 35 35 38 33 34 32 62 64
↪  61 62 33 62 33 31 36 35 34 37 35 38 34 64 62 62 36 65 64 36 61 34 31 62 61 66 31 32 37 66
↪  65 34 39 30 33 30 39 63 22 7d 24 7d 17 7c
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'fe8dc24eafa66c1497558342bdab3b316547584dbb6ed6a41baf127fe490309c'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'7d5542f88dc166eda258a1dafbe347bea05768e3941f3d7b51c1736201f1ffafd0a895b8b3ffeed611c58db⌋
↪  32fdc44b65bb3b1b3416c320c56dd68287ba71b79'"
```

```
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 37 64 35 35
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 37 64 35 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-client: TX -> (64): 34 32 66 38 38 64 63 31 36 36 65 64 61 32 35 38 61 31 64 61 66 62 65
↪ 33 34 37 62 65 61 30 35 37 36 38 65 33 39 34 31 66 33 64 37 62 35 31 63 31 37 33 36 32 30
↪ 31 66 31 66 66 61 66 64 30 61 38
comm-server: RX <- (64): 34 32 66 38 38 64 63 31 36 36 65 64 61 32 35 38 61 31 64 61 66 62 65
↪ 33 34 37 62 65 61 30 35 37 36 38 65 33 39 34 31 66 33 64 37 62 35 31 63 31 37 33 36 32 30
↪ 31 66 31 66 66 61 66 64 30 61 38
comm-client: TX -> (64): 39 35 62 38 62 33 66 66 65 65 64 36 31 31 63 35 38 64 62 33 32 66 64
↪ 63 34 34 62 36 35 62 62 33 62 31 62 33 34 31 36 63 33 32 30 63 35 36 64 64 36 38 32 38 37
↪ 62 61 37 31 62 37 39 22 7d 98 01
comm-server: RX <- (64): 39 35 62 38 62 33 66 66 65 65 64 36 31 31 63 35 38 64 62 33 32 66 64
↪ 63 34 34 62 36 35 62 62 33 62 31 62 33 34 31 36 63 33 32 30 63 35 36 64 64 36 38 32 38 37
↪ 62 61 37 31 62 37 39 22 7d 98 01
comm-client: TX -> (4): 66 00 3a 3e
comm-server: RX <- (4): 66 00 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 37 64 35 35 34 32 66 38 38 64 63 31 36 36 65 64 61 32 35 38 61 31 64 61 66 62
↪ 65 33 34 37 62 65 61 30 35 37 36 38 65 33 39 34 31 66 33 64 37 62 35 31 63 31 37 33 36 32
↪ 30 31 66 31 66 66 61 66 64 30 61 38 39 35 62 38 62 33 66 66 65 65 64 36 31 31 63 35 38 64
↪ 62 33 32 66 64 63 34 34 62 36 35 62 62 33 62 31 62 33 34 31 36 63 33 32 30 63 35 36 64 64
↪ 36 38 32 38 37 62 61 37 31 62 37 39 22 7d 98 01 66 00
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'7d5542f88dc166eda258a1dafbe347bea05768e3941f3d7b51c1736201f1ffafd0a895b8b3ffeed611c58db⌋
↪  32fdc44b65bb3b1b3416c320c56dd68287ba71b79'"
```

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

```
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "False"
```

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

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

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

```
Result (Return Value of authentification method): False (<class 'bool'>)
```

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

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

Info Identical secrets set

Info Performing Authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'d53003b580d496debd0dd6f930dcc3437112a3ed05bd0b74b498c1ecbedb72b3'"
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 64 35 33 30
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 64 35 33 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 -> (64): 30 33 62 35 38 30 64 34 39 36 64 65 62 64 30 64 64 36 66 39 33 30 64
↪  63 63 33 34 33 37 31 31 32 61 33 65 64 30 35 62 64 30 62 37 34 62 34 39 38 63 31 65 63 62
↪  65 64 62 37 32 62 33 22 7d 8e 28
comm-client: RX <- (64): 30 33 62 35 38 30 64 34 39 36 64 65 62 64 30 64 64 36 66 39 33 30 64
↪  63 63 33 34 33 37 31 31 32 61 33 65 64 30 35 62 64 30 62 37 34 62 34 39 38 63 31 65 63 62
↪  65 64 62 37 32 62 33 22 7d 8e 28
comm-server: TX -> (4): 37 54 3a 3e
comm-client: RX <- (4): 37 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 - (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 64 35 33 30 30 33 62 35 38 30 64 34 39 36 64 65 62 64 30 64 64 36 66 39 33 30
↪  64 63 63 33 34 33 37 31 31 32 61 33 65 64 30 35 62 64 30 62 37 34 62 34 39 38 63 31 65 63
↪  62 65 64 62 37 32 62 33 22 7d 8e 28 37 54
```

```
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'d53003b580d496debd0dd6f930dcc3437112a3ed05bd0b74b498c1ecbedb72b3'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'f67e465ccb6d5037bd908737a3bc6c95278fe13a26088182f87f65bdb3db8c3e8fd6781e762c7ccd31d95c5⌋
↪  cbaab8bfa9e35f754ad6ff958b8bfa50f1c712b48'"
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 36 37 65
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 36 37 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 -> (64): 34 36 35 63 63 62 36 64 35 30 33 37 62 64 39 30 38 37 33 37 61 33 62
↪  63 36 63 39 35 32 37 38 66 65 31 33 61 32 36 30 38 38 31 38 32 66 38 37 66 36 35 62 64 62
↪  33 64 62 38 63 33 65 38 66 64 36
comm-server: RX <- (64): 34 36 35 63 63 62 36 64 35 30 33 37 62 64 39 30 38 37 33 37 61 33 62
↪  63 36 63 39 35 32 37 38 66 65 31 33 61 32 36 30 38 38 31 38 32 66 38 37 66 36 35 62 64 62
↪  33 64 62 38 63 33 65 38 66 64 36
comm-client: TX -> (64): 37 38 31 65 37 36 32 63 37 63 63 64 33 31 64 39 35 63 35 63 62 61 61
↪  62 38 62 66 61 39 65 33 35 66 37 35 34 61 64 36 66 66 39 35 38 62 38 62 66 61 35 30 66 31
↪  63 37 31 32 62 34 38 22 7d 65 15
comm-server: RX <- (64): 37 38 31 65 37 36 32 63 37 63 63 64 33 31 64 39 35 63 35 63 62 61 61
↪  62 38 62 66 61 39 65 33 35 66 37 35 34 61 64 36 66 66 39 35 38 62 38 62 66 61 35 30 66 31
↪  63 37 31 32 62 34 38 22 7d 65 15
comm-client: TX -> (4): 31 b5 3a 3e
comm-server: RX <- (4): 31 b5 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 36 37 65 34 36 35 63 63 62 36 64 35 30 33 37 62 64 39 30 38 37 33 37 61 33
↪  62 63 36 63 39 35 32 37 38 66 65 31 33 61 32 36 30 38 38 31 38 32 66 38 37 66 36 35 62 64
↪  62 33 64 62 38 63 33 65 38 66 64 36 37 38 31 65 37 36 32 63 37 63 63 64 33 31 64 39 35 63
↪  35 63 62 61 61 62 38 62 66 61 39 65 33 35 66 37 35 34 61 64 36 66 66 39 35 38 62 38 62 66
↪  61 35 30 66 31 63 37 31 32 62 34 38 22 7d 65 15 31 b5
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'f67e465ccb6d5037bd908737a3bc6c95278fe13a26088182f87f65bdb3db8c3e8fd6781e762c7ccd31d95c5⌟
↪  cbaab8bfa9e35f754ad6ff958b8bfa50f1c712b48'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
```

```
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

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

```
Result (Return Value of authentification method): True (<class 'bool'>)
Expectation (Return Value of authentification method): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of server): True (<class 'bool'>)
Expectation (Authentification state of server): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of client): True (<class 'bool'>)
Expectation (Authentification state of client): result = True (<class 'bool'>)
```

Info Corrupting the authentification mechanism

```
prot-server: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
prot-client: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Info Performing Authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
```

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

```
Result (Return Value of authentification method): False (<class 'bool'>)
Expectation (Return Value of authentification method): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

B.1.8 An automatic authentification shall available

Description

An authentification is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentification.

Fitcriterion

Check authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification

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

```
Result (Authentification state of server): False (<class 'bool'>)
Expectation (Authentification state of server): result = False (<class 'bool'>)
```

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

```
Result (Authentification state of client): False (<class 'bool'>)
Expectation (Authentification state of client): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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: authentification 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
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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'1460017567fec628b855c6ae8bff3b72bb4630b3168d1b4956fc6efec032d255'"
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 31 34 36 30
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 31 34 36 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 -> (64): 30 31 37 35 36 37 66 65 63 36 32 38 62 38 35 35 63 36 61 65 38 62 66
↪  66 33 62 37 32 62 62 34 36 33 30 62 33 31 36 38 64 31 62 34 39 35 36 66 63 36 65 66 65 63
↪  30 33 32 64 32 35 35 22 7d 76 ff
comm-client: RX <- (64): 30 31 37 35 36 37 66 65 63 36 32 38 62 38 35 35 63 36 61 65 38 62 66
↪  66 33 62 37 32 62 62 34 36 33 30 62 33 31 36 38 64 31 62 34 39 35 36 66 63 36 65 66 65 63
↪  30 33 32 64 32 35 35 22 7d 76 ff
comm-server: TX -> (4): 5f 74 3a 3e
comm-client: RX <- (4): 5f 74 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 31 34 36 30 30 31 37 35 36 37 66 65 63 36 32 38 62 38 35 35 63 36 61 65 38 62
↪  66 66 33 62 37 32 62 62 34 36 33 30 62 33 31 36 38 64 31 62 34 39 35 36 66 63 36 65 66 65
↪  63 30 33 32 64 32 35 35 22 7d 76 ff 5f 74
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'1460017567fec628b855c6ae8bff3b72bb4630b3168d1b4956fc6efec032d255'"
```

```
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'f82396ff393988f31566f33fbdc3cbcccec0aeaea9178fa801d2231107ced1fc02c72699df0a181003df0d0⌟
↪  e55140ec80c772b73bc3938bf17f2749e2f652997'"
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 38 32 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 66 38 32 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): 39 36 66 66 33 39 33 39 38 38 66 33 31 35 36 36 66 33 33 66 62 64 63
↪  33 63 62 63 63 63 65 63 30 61 65 61 65 61 39 31 37 38 66 61 38 30 31 64 32 32 33 31 31 30
↪  37 63 65 64 31 66 63 30 32 63 37
comm-server: RX <- (64): 39 36 66 66 33 39 33 39 38 38 66 33 31 35 36 36 66 33 33 66 62 64 63
↪  33 63 62 63 63 63 65 63 30 61 65 61 65 61 39 31 37 38 66 61 38 30 31 64 32 32 33 31 31 30
↪  37 63 65 64 31 66 63 30 32 63 37
comm-client: TX -> (64): 32 36 39 39 64 66 30 61 31 38 31 30 30 33 64 66 30 64 30 65 35 35 31
↪  34 30 65 63 38 30 63 37 37 32 62 37 33 62 63 33 39 33 38 62 66 31 37 66 32 37 34 39 65 32
↪  66 36 35 32 39 39 37 22 7d ba 69
comm-server: RX <- (64): 32 36 39 39 64 66 30 61 31 38 31 30 30 33 64 66 30 64 30 65 35 35 31
↪  34 30 65 63 38 30 63 37 37 32 62 37 33 62 63 33 39 33 38 62 66 31 37 66 32 37 34 39 65 32
↪  66 36 35 32 39 39 37 22 7d ba 69
comm-client: TX -> (4): b5 c1 3a 3e
comm-server: RX <- (4): b5 c1 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 38 32 33 39 36 66 66 33 39 33 39 38 38 66 33 31 35 36 36 66 33 33 66 62 64
↪  63 33 63 62 63 63 63 65 63 30 61 65 61 65 61 39 31 37 38 66 61 38 30 31 64 32 32 33 31 31
↪  30 37 63 65 64 31 66 63 30 32 63 37 32 36 39 39 64 66 30 61 31 38 31 30 30 33 64 66 30 64
↪  30 65 35 35 31 34 30 65 63 38 30 63 37 37 32 62 37 33 62 63 33 39 33 38 62 66 31 37 66 32
↪  37 34 39 65 32 66 36 35 32 39 39 37 22 7d ba 69 b5 c1
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'f82396ff393988f31566f33fbdc3cbcccec0aeaea9178fa801d2231107ced1fc02c72699df0a181003df0d0⌟
↪  e55140ec80c772b73bc3938bf17f2749e2f652997'"
```

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

```
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
```

```
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

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

```
Result (Authentification state of server): True (<class 'bool'>)
Expectation (Authentification state of server): result = True (<class 'bool'>)
```

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

```
Result (Authentification state of client): True (<class 'bool'>)
Expectation (Authentification state of client): result = True (<class 'bool'>)
```

B.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentification 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 Authentification 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 authentification 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=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
```

```
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
↪  "'msg3_data_to_be_transfered'"
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_transfered'"
```

```
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: authentification 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 Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
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 <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 Transfering a message client → server

```
prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↪  data: 'msg1_data_to_be_transfered' 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 Transfering 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_transfered' 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 Authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'b6080a2dd0172fde98ba6270f5d2c08c412d0e4d21af4de23f219a4013bd9e9c'"
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 36 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 36 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): 30 61 32 64 64 30 31 37 32 66 64 65 39 38 62 61 36 32 37 30 66 35 64
↪  32 63 30 38 63 34 31 32 64 30 65 34 64 32 31 61 66 34 64 65 32 33 66 32 31 39 61 34 30 31
↪  33 62 64 39 65 39 63 22 7d 0d c9
comm-client: RX <- (64): 30 61 32 64 64 30 31 37 32 66 64 65 39 38 62 61 36 32 37 30 66 35 64
↪  32 63 30 38 63 34 31 32 64 30 65 34 64 32 31 61 66 34 64 65 32 33 66 32 31 39 61 34 30 31
↪  33 62 64 39 65 39 63 22 7d 0d c9
comm-server: TX -> (4): bf a4 3a 3e
comm-client: RX <- (4): bf 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 - (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 36 30 38 30 61 32 64 64 30 31 37 32 66 64 65 39 38 62 61 36 32 37 30 66 35
↪  64 32 63 30 38 63 34 31 32 64 30 65 34 64 32 31 61 66 34 64 65 32 33 66 32 31 39 61 34 30
↪  31 33 62 64 39 65 39 63 22 7d 0d c9 bf a4
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'b6080a2dd0172fde98ba6270f5d2c08c412d0e4d21af4de23f219a4013bd9e9c'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'c37648c08b6b21da5dc2a8ca1d38c3f287f0363e6c48ee0ebee0cdf45c5abd8379e0ea8b32fbec7aa0f0dfa⌟
↪  cd4038beedfd395858c4412586c0fdc6f46ee79bb'"
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 63 33 37 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 63 33 37 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): 34 38 63 30 38 62 36 62 32 31 64 61 35 64 63 32 61 38 63 61 31 64 33
↪  38 63 33 66 32 38 37 66 30 33 36 33 65 36 63 34 38 65 65 30 65 62 65 65 30 63 64 66 34 35
↪  63 35 61 62 64 38 33 37 39 65 30
comm-server: RX <- (64): 34 38 63 30 38 62 36 62 32 31 64 61 35 64 63 32 61 38 63 61 31 64 33
↪  38 63 33 66 32 38 37 66 30 33 36 33 65 36 63 34 38 65 65 30 65 62 65 65 30 63 64 66 34 35
↪  63 35 61 62 64 38 33 37 39 65 30
comm-client: TX -> (64): 65 61 38 62 33 32 66 62 65 63 37 61 61 30 66 30 64 66 61 63 64 34 30
↪  33 38 62 65 65 64 66 64 33 39 35 38 35 38 63 34 34 31 32 35 38 36 63 30 66 64 63 36 66 34
↪  36 65 65 37 39 62 62 22 7d e1 21
comm-server: RX <- (64): 65 61 38 62 33 32 66 62 65 63 37 61 61 30 66 30 64 66 61 63 64 34 30
↪  33 38 62 65 65 64 66 64 33 39 35 38 35 38 63 34 34 31 32 35 38 36 63 30 66 64 63 36 66 34
↪  36 65 65 37 39 62 62 22 7d e1 21
comm-client: TX -> (4): 7c 4a 3a 3e
comm-server: RX <- (4): 7c 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 - (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 63 33 37 36 34 38 63 30 38 62 36 62 32 31 64 61 35 64 63 32 61 38 63 61 31 64
↪  33 38 63 33 66 32 38 37 66 30 33 36 33 65 36 63 34 38 65 65 30 65 62 65 65 30 63 64 66 34
↪  35 63 35 61 62 64 38 33 37 39 65 30 65 61 38 62 33 32 66 62 65 63 37 61 61 30 66 30 64 66
↪  61 63 64 34 30 33 38 62 65 65 64 66 64 33 39 35 38 35 38 63 34 34 31 32 35 38 36 63 30 66
↪  64 63 36 66 34 36 65 65 37 39 62 62 22 7d e1 21 7c 4a
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'c37648c08b6b21da5dc2a8ca1d38c3f287f0363e6c48ee0ebee0cdf45c5abd8379e0ea8b32fbec7aa0f0dfa⌟
↪  cd4038beedfd395858c4412586c0fdc6f46ee79bb'"
prot-server: Executing callback __authentificate_check_key__ to process received data
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
prot-client: Executing callback __authentificate_process_feedback__ to process received data
prot-client: Got positive authentification feedback
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪ 'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪ "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<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 authentification was successfull performed.

Reason for the implementation

Give the user the possibility to define messages which will not be protected behind the authentification 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**.

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=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪ authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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=authentification request and
↪ Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪ authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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 Transfering a message client → server

```
prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↪ data: 'msg1_data_to_be_transfered' 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 Transfering 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_transfered' 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 authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
prot-server: Authentification 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 Transfering 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_transfered' 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 authentification whitelist
```

Info Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering 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_transfered' will be ignored.
```

```
prot-client: TIMEOUT (0.2870553359683794 5s): 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
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_transfered'"
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_transfered'} 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_transfered'} (<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_transfered'} (<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 incomming connection
```

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

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

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

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

```
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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
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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪ Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪ authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪ authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪ authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪ authentification 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=authentification request and
↪ Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪ authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪ authentification whitelist
```

```
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
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_transfered'"
```

```
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 Transfering a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
prot-server: Incomming 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 incomming connection
prot-server: Cleaning up receive-buffer
```

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

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

```
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
```

```
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering 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 Transfering 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: read data request, 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 Transfering 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
```

```
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=None
```

Info Transfering 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
```

```
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering 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 incomming 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=None
```

Info Transfering 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering 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 Transfering 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 Transfering 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 Transfering 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
```

```
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
```

```
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 authentification

```
prot-client: TX -> service: authentification 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: authentification request, data_id: seed, status: okay, data:
↪  "None"
prot-server: Executing callback __authentificate_create_seed__ to process received data
prot-server: TX -> service: authentification response, data_id: seed, status: okay, data:
↪  "'4381366da0b29c3979c36e7116cabd7b7024a01405f21554b4ff3cedd66d417a'"
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 33 38 31
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 33 38 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 -> (64): 33 36 36 64 61 30 62 32 39 63 33 39 37 39 63 33 36 65 37 31 31 36 63
↪  61 62 64 37 62 37 30 32 34 61 30 31 34 30 35 66 32 31 35 35 34 62 34 66 66 33 63 65 64 64
↪  36 36 64 34 31 37 61 22 7d 31 b8
comm-client: RX <- (64): 33 36 36 64 61 30 62 32 39 63 33 39 37 39 63 33 36 65 37 31 31 36 63
↪  61 62 64 37 62 37 30 32 34 61 30 31 34 30 35 66 32 31 35 35 34 62 34 66 66 33 63 65 64 64
↪  36 36 64 34 31 37 61 22 7d 31 b8
comm-server: TX -> (4): 42 3d 3a 3e
comm-client: RX <- (4): 42 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 - (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 33 38 31 33 36 36 64 61 30 62 32 39 63 33 39 37 39 63 33 36 65 37 31 31 36
↪  63 61 62 64 37 62 37 30 32 34 61 30 31 34 30 35 66 32 31 35 35 34 62 34 66 66 33 63 65 64
↪  64 36 36 64 34 31 37 61 22 7d 31 b8 42 3d
prot-client: RX <- service: authentification response, data_id: seed, status: okay, data:
↪  "'4381366da0b29c3979c36e7116cabd7b7024a01405f21554b4ff3cedd66d417a'"
prot-client: Executing callback __authentificate_create_key__ to process received data
prot-client: TX -> service: authentification request, data_id: key, status: okay, data:
↪  "'6c79228613fc62db89ad4d3c8c86c1f81530cbd9796dbe56e181bd64d205c3aeb2590ea31b0bfa0d244e058┘
↪  d39bc0ca2a7eb19671551b3f441b0cba9f19aea42'"
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 36 63 37 39
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 36 63 37 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
STP: data sync (3a) received => changing state STP_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): 32 32 38 36 31 33 66 63 36 32 64 62 38 39 61 64 34 64 33 63 38 63 38
↪  36 63 31 66 38 31 35 33 30 63 62 64 39 37 39 36 64 62 65 35 36 65 31 38 31 62 64 36 34 64
↪  32 30 35 63 33 61 65 62 32 35 39
```

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

```
comm-client: TX -> (64): 30 65 61 33 31 62 30 62 66 61 30 64 32 34 34 65 30 35 38 64 33 39 62
↪  63 30 63 61 32 61 37 65 62 31 39 36 37 31 35 35 31 62 33 66 34 34 31 62 30 63 62 61 39 66
↪  31 39 61 65 61 34 32 22 7d b3 f5
```

```
comm-server: RX <- (64): 30 65 61 33 31 62 30 62 66 61 30 64 32 34 34 65 30 35 38 64 33 39 62
↪  63 30 63 61 32 61 37 65 62 31 39 36 37 31 35 35 31 62 33 66 34 34 31 62 30 63 62 61 39 66
↪  31 39 61 65 61 34 32 22 7d b3 f5
```

```
comm-client: TX -> (4): a9 cf 3a 3e
```

```
comm-server: RX <- (4): a9 cf 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 36 63 37 39 32 32 38 36 31 33 66 63 36 32 64 62 38 39 61 64 34 64 33 63 38 63
↪  38 36 63 31 66 38 31 35 33 30 63 62 64 39 37 39 36 64 62 65 35 36 65 31 38 31 62 64 36 34
↪  64 32 30 35 63 33 61 65 62 32 35 39 30 65 61 33 31 62 30 62 66 61 30 64 32 34 34 65 30 35
↪  38 64 33 39 62 63 30 63 61 32 61 37 65 62 31 39 36 37 31 35 35 31 62 33 66 34 34 31 62 30
↪  63 62 61 39 66 31 39 61 65 61 34 32 22 7d b3 f5 a9 cf
```

```
prot-server: RX <- service: authentification request, data_id: key, status: okay, data:
↪  "'6c79228613fc62db89ad4d3c8c86c1f81530cbd9796dbe56e181bd64d205c3aeb2590ea31b0bfa0d244e058⌋
↪  d39bc0ca2a7eb19671551b3f441b0cba9f19aea42'"
```

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

```
prot-server: TX -> service: authentification 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: authentification response, data_id: key, status: okay, data:
↪  "True"
```

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

```
prot-client: Got positive authentification 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 incomming connection
```

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

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

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

```
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
```

```
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 incomming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentification request and
↪  Response=authentification response
prot-server: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
```

```
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 authentification state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentification 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 incomming connection
```

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

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

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

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

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

```
prot-server: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-server: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentification 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
↪  authentification whitelist
prot-server: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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=authentification request and
↪  Response=authentification response
prot-client: Adding Message (service: authentification request, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: seed) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification request, data_id: key) to the
↪  authentification whitelist
prot-client: Adding Message (service: authentification response, data_id: key) to the
↪  authentification whitelist
prot-client: Adding callback '__authentificate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authentificate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authentificate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authentificate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentification 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
↪  authentification whitelist
prot-client: Adding Message (service: channel name response, data_id: name) to the
↪  authentification 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 Transfering a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪  "'msg1_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

Info Transfering a message server → client

```
prot-server: TX -> service: 17, data_id: 35, status: service or data unknown, data:
↪  "'msg2_data_to_be_transfered'"
```

```
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_transfered'"
```

```
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_transfered'} 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_transfered'} (<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_transfered'} (<class
↪  'socket_protocol.data_storage'>)
```

C Test-Coverage

C.1 socket_protocol

The line coverage for socket_protocol was 99.5%
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 99.5%
The branch coverage for socket_protocol.__init__.py was 100.0%

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
"""
socket_protocol (Socket Protocol)
=================================

**Author:**

* Dirk Alders <sudo-dirk@mount-mockery.de>

**Description:**

    This Module supports point to point communication for client-server issues.

**Submodules:**

```

```
* :class:`socket_protocol.data_storage`
* :class:`socket_protocol.pure_json_protocol`
* :class:`socket_protocol.struct_json_protocol`

**Unittest:**

        See also the :download:`unittest <socket_protocol/_testresults_/unittest.pdf>` documentation.

**Module Documentation:**

"""
__DEPENDENCIES__ = ['stringtools']

import stringtools

import binascii
import hashlib
import json
import logging
import os
import struct
import sys
import time


try:
    from config import APP_NAME as ROOT_LOGGER_NAME
except ImportError:
    ROOT_LOGGER_NAME = 'root'
logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)


__DESCRIPTION__ = """The Module {\\tt %s} is designed for point to point communication for client-server issues.
For more Information read the sphinx documentation.""" % __name__.replace('_', '\_')
"""The Module Description"""
__INTERPRETER__ = (2, 3)
"""The Tested Interpreter-Versions"""

SID_AUTH_REQUEST = 0
"""SID for authentification request"""
SID_AUTH_RESPONSE = 1
"""SID for authentification response"""
DID_AUTH_SEED = 0
"""DID for authentification (seed)"""
DID_AUTH_KEY = 1
"""DID for authentification (key)"""
SID_CHANNEL_NAME_REQUEST = 8
"""SID for channel name exchange request """
SID_CHANNEL_NAME_RESPONSE = 9
"""SID for channel name exchange response"""
DID_CHANNEL_NAME = 0
"""DID for channel name """
SID_READ_REQUEST = 10
"""SID for a read data request"""
SID_READ_RESPONSE = 11
"""SID for read data response"""
SID_WRITE_REQUEST = 20
"""SID for a write data request"""
SID_WRITE_RESPONSE = 21
"""SID for a write data response"""
SID_EXECUTE_REQUEST = 30
```

```
"""SID for a execute request"""
SID_EXECUTE_RESPONSE = 31
"""SID for a execute response"""

STATUS_OKAY = 0
"""Status for 'okay'"""
STATUS_BUFFERING_UNHANDLED_REQUEST = 1
"""Status for 'unhandled request'"""
STATUS_CALLBACK_ERROR = 2
"""Status for 'callback errors'"""
STATUS_AUTH_REQUIRED = 3
"""Status for 'authentification is required'"""
STATUS_SERVICE_OR_DATA_UNKNOWN = 4
"""Status for 'service or data unknown'"""
STATUS_CHECKSUM_ERROR = 5
"""Status for 'checksum error'"""
STATUS_OPERATION_NOT_PERMITTED = 6
"""Status for 'operation not permitted'"""
STATUS_LOG_LVL = {
    STATUS_OKAY: logging.INFO,
    STATUS_BUFFERING_UNHANDLED_REQUEST: logging.WARNING,
    STATUS_CALLBACK_ERROR: logging.ERROR,
    STATUS_AUTH_REQUIRED: logging.WARNING,
    STATUS_SERVICE_OR_DATA_UNKNOWN: logging.ERROR,
    STATUS_CHECKSUM_ERROR: logging.ERROR,
    STATUS_OPERATION_NOT_PERMITTED: logging.WARNING,
}
"""Status depending log level for messages"""

AUTH_STATE_UNTRUSTED_CONNECTION = 0
"""Authentification Status for an 'Untrusted Connection'"""
AUTH_STATE_SEED_REQUESTED = 1
"""Authentification Status for 'Seed was requested'"""
AUTH_STATE_SEED_TRANSFERRED = 2
"""Authentification Status for 'Seed has been sent'"""
AUTH_STATE_KEY_TRANSFERRED = 3
"""Authentification Status for 'Key has been sent'"""
AUTH_STATE_TRUSTED_CONNECTION = 4
"""Authentification Status for a 'Trusted Connection'"""
AUTH_STATE__NAMES = {AUTH_STATE_UNTRUSTED_CONNECTION: 'Untrusted Connection',
                     AUTH_STATE_SEED_REQUESTED: 'Seed was requested',
                     AUTH_STATE_SEED_TRANSFERRED: 'Seed has been sent',
                     AUTH_STATE_KEY_TRANSFERRED: 'Key has been sent',
                     AUTH_STATE_TRUSTED_CONNECTION: 'Trusted Connection'}
"""Authentification Status names for previous defined authentification states"""


class RequestSidExistsError(Exception):
    pass


class ResponseSidExistsError(Exception):
    pass


class _callback_storage(dict):
    DEFAULT_CHANNEL_NAME = 'all_others'

    def __init__(self, channel_name, log_prefix):
        self.init_channel_name(channel_name)
        self.__log_prefix__ = log_prefix
        dict.__init__(self)

```

```
    def init_channel_name(self, channel_name):
        if channel_name is None:
            self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.DEFAULT_CHANNEL_NAME)
        else:
            self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + channel_name)

    def get(self, service_id, data_id):
        if dict.get(self, service_id, {}).get(data_id, None) is not None:
            return self[service_id][data_id]
        elif dict.get(self, service_id, {}).get(None, None) is not None:
            return self[service_id][None]
        elif dict.get(self, None, {}).get(data_id, None) is not None:
            return self[None][data_id]
        elif dict.get(self, None, {}).get(None, None) is not None:
            return self[None][None]
        else:
            return (None, None, None)

    def add(self, service_id, data_id, callback, *args, **kwargs):
        cb_data = self.get(service_id, data_id)
        if dict.get(self, service_id, {}).get(data_id, None) is not None:
            if callback is None:
                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))
                del(self[service_id][data_id])
                return
            else:
                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__))
        else:
            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))
        if service_id not in self:
            self[service_id] = {}
        self[service_id][data_id] = (callback, args, kwargs)


class data_storage(dict):
    """
    This is a storage object for socket_protocol messages.

    :param status: The message status.
    :type status: int
    :param service_id: The Service-ID.
    :type service_id: int
    :param data_id: The Data-ID.
    :type data_id: int
    :param data: The transfered data.
    :type data: any
    """

    KEY_STATUS = 'status'
    KEY_SERVICE_ID = 'service_id'
    KEY_DATA_ID = 'data_id'
    KEY_DATA = 'data'
    ALL_KEYS = [KEY_DATA, KEY_DATA_ID, KEY_SERVICE_ID, KEY_STATUS]

```

```python
    def __init__(self, *args, **kwargs):
        dict.__init__(self, *args, **kwargs)
        for key in self.ALL_KEYS:
            if key not in self:
                self[key] = None

    def get_status(self, default=None):
        """
        This Method returns the message status.

        :param default: The default value, if no data is available.
        """
        return self.get(self.KEY_STATUS, default)

    def get_service_id(self, default=None):
        """
        This Method returns the message Service-ID.

        :param default: The default value, if no data is available.
        """
        return self.get(self.KEY_SERVICE_ID, default)

    def get_data_id(self, default=None):
        """
        This Method returns the message Data-ID.

        :param default: The default value, if no data is available.
        """
        return self.get(self.KEY_DATA_ID, default)

    def get_data(self, default=None):
        """
        This Method returns the message data.

        :param default: The default value, if no data is available.
        """
        return self.get(self.KEY_DATA, default)


class pure_json_protocol(object):
    """
    This `class` supports to transfer a message and it's data.

    :param comm_instance: A communication instance.
    :type comm_instance: instance
    :param secret: An optinal secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
    :type secret: str
    :param auto_auth: An optional parameter to enable (True) automatic authentification, otherwise you need to do it manually, if needed.
    :type auto_auth: bool
    :param channel_name: An optional parameter to set a channel name for logging of the communication.
    :type channel_name: str

    .. hint::

        * The Service-ID is designed to identify the type of the communication (e.g. :const:`READ_REQUEST`, :const:`WRITE_REQUEST`, :const:`READ_RESPONSE`, :const:`WRITE_RESPONSE`, ...)
        * The Data-ID is designed to identify the requests / responses using the same Service_ID.

    .. note:: The :class:`comm_instance` needs to have at least the following interface:

```

```
        * A Method :func:`comm_instance.init_channel_name` to set the channel name.
        * A Constant :const:`comm_instance.IS_CLIENT` to identify that the :class:`comm_instance` is a client (True) or a server (False).
        * A Method :func:`comm_instance.is_connected` to identify if the instance is connected (True) or not (False).
        * A Method :func:`comm_instance.reconnect` to initiate a reconnect.
        * A Method :func:`comm_instance.register_callback` to register a data available callback.
        * A Method :func:`comm_instance.register_connect_callback` to register a connect callback.
        * A Method :func:`comm_instance.register_disconnect_callback` to register a disconnect callback.
        * A Method :func:`comm_instance.send` to send data via the :class:`comm_instance`.

    .. note::  The parameter :const:`auto_auth` is only relevant, if a secret is given and the :class:`comm_instance` is a client. The authentification is initiated directly after the connection is established.

    .. note::  The :const:`channel_name`-exchange will be initiated by the client directly after the the connection is established.

        * 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.

    **Example:**

    .. literalinclude:: socket_protocol/_examples_/socket_protocol_client.py

    and

    .. literalinclude:: socket_protocol/_examples_/socket_protocol_server.py

    Will result to the following output:

    .. literalinclude:: socket_protocol/_examples_/socket_protocol_client.log
    """
    DEFAULT_CHANNEL_NAME = 'all_others'

    def __init__(self, comm_instance, secret=None, auto_auth=False, channel_name=None):
        self.__comm_inst__ = comm_instance
        self.__secret__ = secret
        self.__auto_auth__ = auto_auth
        #
        self.__auth_whitelist__ = {}
        self.__sid_response_dict__ = {}
        self.__sid_name_dict__ = {}
        self.__did_name_dict__ = {}
        #
        self.__status_name_dict = {}
        self.add_status(STATUS_OKAY, 'okay')
        self.add_status(STATUS_BUFFERING_UNHANDLED_REQUEST, 'no callback for service, data buffered')
        self.add_status(STATUS_CALLBACK_ERROR, 'callback error')
        self.add_status(STATUS_AUTH_REQUIRED, 'authentification required')
        self.add_status(STATUS_SERVICE_OR_DATA_UNKNOWN, 'service or data unknown')
        self.add_status(STATUS_CHECKSUM_ERROR, 'checksum error')
        self.add_status(STATUS_OPERATION_NOT_PERMITTED, 'operation not permitted')
        #
        self.__callbacks__ = _callback_storage(channel_name, self.__log_prefix__)
        self.__init_channel_name__(channel_name)
        #
        self.__clean_receive_buffer__()

```

```
        self.add_service(SID_AUTH_REQUEST, SID_AUTH_RESPONSE, 'authentification request', 'authentification response')
        self.add_data((SID_AUTH_REQUEST, SID_AUTH_RESPONSE), DID_AUTH_SEED, 'seed')
        self.add_data(SID_AUTH_REQUEST, DID_AUTH_KEY, 'key')
        self.add_data(SID_AUTH_RESPONSE, DID_AUTH_KEY, 'key')
        self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_SEED)
        self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_SEED)
        self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_KEY)
        self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_KEY)
        self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_SEED, self.__authentificate_create_seed__)
        self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_SEED, self.__authentificate_create_key__)
        self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_KEY, self.__authentificate_check_key__)
        self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_KEY, self.__authentificate_process_feedback__)
        self.__authentification_state_reset__()

        self.add_service(SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE, 'channel name request', 'channel name response')
        self.add_data((SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE), DID_CHANNEL_NAME, 'name')
        self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME)
        self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME)
        self.__callbacks__.add(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME, self.__channel_name_request__)
        self.__callbacks__.add(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME, self.__channel_name_response__)

        self.add_service(SID_READ_REQUEST, SID_READ_RESPONSE, 'read data request', 'read data response')
        self.add_service(SID_WRITE_REQUEST, SID_WRITE_RESPONSE, 'write data request', 'write data response')
        self.add_service(SID_EXECUTE_REQUEST, SID_EXECUTE_RESPONSE, 'execute request', 'execute response')

        self.__seed__ = None
        self.__comm_inst__.register_callback(self.__data_available_callback__)
        self.__comm_inst__.register_connect_callback(self.__connection_established__)
        self.__comm_inst__.register_disconnect_callback(self.__authentification_state_reset__)
        logger.info('%s Initialisation finished.', self.__log_prefix__())

    def __analyse_frame__(self, frame):
        if sys.version_info >= (3, 0):
            return data_storage(json.loads(frame[:-4].decode('utf-8')))
        else:
            return data_storage(json.loads(frame[:-4]))

    def __authentificate_check_key__(self, msg):
        key = msg.get_data()
        if key == self.__authentificate_salt_and_hash__(self.__seed__):
            self.__authentification_state__ = AUTH_STATE_TRUSTED_CONNECTION
            return STATUS_OKAY, True
        else:
            self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
            return STATUS_OKAY, False

    def __authentificate_create_key__(self, msg):
        self.__authentification_state__ = AUTH_STATE_KEY_TRANSFERRED
        seed = msg.get_data()
        key = self.__authentificate_salt_and_hash__(seed)
        self.send(SID_AUTH_REQUEST, DID_AUTH_KEY, key)
```

```python

    def __authentificate_create_seed__(self, msg):
        self.__authentification_state__ = AUTH_STATE_SEED_TRANSFERRED
        if sys.version_info >= (3, 0):
            self.__seed__ = binascii.hexlify(os.urandom(32)).decode('utf-8')
        else:
            self.__seed__ = binascii.hexlify(os.urandom(32))
        return STATUS_OKAY, self.__seed__

    def __authentificate_process_feedback__(self, msg):
        feedback = msg.get_data()
        if feedback:
            self.__authentification_state__ = AUTH_STATE_TRUSTED_CONNECTION
            self.logger.info("%s Got positive authentification feedback", self.__log_prefix__())
        else:
            self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
            self.logger.warning("%s Got negative authentification feedback", self.__log_prefix__())
        return STATUS_OKAY, None

    def __authentificate_salt_and_hash__(self, seed):
        if sys.version_info >= (3, 0):
            return hashlib.sha512(bytes(seed, 'utf-8') + self.__secret__).hexdigest()
        else:
            return hashlib.sha512(seed.encode('utf-8') + self.__secret__.encode('utf-8')).hexdigest()

    def __authentification_state_reset__(self):
        self.logger.info("%s Resetting authentification state to AUTH_STATE_UNTRUSTED_CONNECTION", self.__log_prefix__())
        self.__authentification_state__ = AUTH_STATE_UNTRUSTED_CONNECTION

    def __authentification_required__(self, service_id, data_id):
        return data_id not in self.__auth_whitelist__.get(service_id, [])

    def __buffer_received_data__(self, msg):
        if not msg.get_service_id() in self.__msg_buffer__:
            self.__msg_buffer__[msg.get_service_id()] = {}
        if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
            self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
        self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
        self.logger.debug("%s Message data is stored in buffer and is now ready to be retrieved by receive method", self.__log_prefix__())

    def __build_frame__(self, msg):
        data_frame = json.dumps(self.__mk_msg__(msg.get_status(), msg.get_service_id(), msg.get_data_id(), msg.get_data()))
        if sys.version_info >= (3, 0):
            data_frame = bytes(data_frame, 'utf-8')
        checksum = self.__calc_chksum__(data_frame)
        return data_frame + checksum

    def __calc_chksum__(self, raw_data):
        return struct.pack('>I', binascii.crc32(raw_data) & 0xffffffff)

    @property
    def __channel_name__(self):
        cn = self.logger.name.split('.')[-1]
        if cn != self.DEFAULT_CHANNEL_NAME:
            return cn

```

```
    def __channel_name_response__(self, msg):
        data = msg.get_data()
        if self.__channel_name__ is None and data is not None:
            self.__init_channel_name__(data)
            self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.__channel_name__))
        return STATUS_OKAY, None

    def __channel_name_request__(self, msg):
        data = msg.get_data()
        if data is None:
            return STATUS_OKAY, self.__channel_name__
        else:
            prev_channel_name = self.__channel_name__
            self.__init_channel_name__(data)
            if prev_channel_name is not None and prev_channel_name != data:
                self.logger.warning('%s overwriting user defined channel name from %s to %s', self.__log_prefix__(), repr(prev_channel_name), repr(data))
            elif prev_channel_name is None:
                self.logger.info('%s channel name is now %s', self.__log_prefix__(), repr(self.__channel_name__))
            return STATUS_OKAY, None

    def __check_frame_checksum__(self, frame):
        return self.__calc_chksum__(frame[:-4]) == frame[-4:]

    def __clean_receive_buffer__(self):
        self.logger.debug("%s Cleaning up receive-buffer", self.__log_prefix__())
        self.__msg_buffer__ = {}

    def __connection_established__(self):
        self.__clean_receive_buffer__()
        if self.__comm_inst__.IS_CLIENT:
            self.send(SID_CHANNEL_NAME_REQUEST, 0, self.__channel_name__)
        if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
            self.authentificate()

    def __log_msg__(self, msg, rx_tx_prefix):
        self.logger.log(
            self.__status_log_lvl__(msg.get_status()),
            '%s %s %s, %s, data: "%s"',
            self.__log_prefix__(),
            rx_tx_prefix,
            self.__get_message_name__(msg.get_service_id(), msg.get_data_id()),
            self.__get_status_name__(msg.get_status()),
            repr(msg.get_data())
        )

    def __data_available_callback__(self, comm_inst):
        frame = comm_inst.receive()
        msg = self.__analyse_frame__(frame)
        if not self.__check_frame_checksum__(frame):
            # Wrong Checksum
            self.logger.log(self.__status_log_lvl__(STATUS_CHECKSUM_ERROR), "%s Received message has an invalid checksum. Message will be ignored.", self.__log_prefix__())
            return  # No response needed
        elif not self.check_authentification_state() and self.__authentification_required__(msg.get_service_id(), msg.get_data_id()):
            # Authentification required
```

```
            self.__log_msg__(msg, 'RX <-')
            if msg.get_service_id() in self.__sid_response_dict__.keys():
                self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED), "%s Authentification is required. Just sending negative response.", self.__log_prefix__())
                status = STATUS_AUTH_REQUIRED
                data = None
            else:
                self.logger.log(self.__status_log_lvl__(STATUS_AUTH_REQUIRED), "%s Authentification is required. Incomming message will be ignored.", self.__log_prefix__())
                return  # No response needed
        else:
            # Valid message
            self.__log_msg__(msg, 'RX <-')
            callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id())
            if msg.get_service_id() in self.__sid_response_dict__.keys():
                #
                # REQUEST RECEIVED
                #
                if callback is None:
                    self.logger.warning("%s Incomming message with no registered callback. Sending negative response.", self.__log_prefix__())
                    status = STATUS_BUFFERING_UNHANDLED_REQUEST
                    data = None
                else:
                    self.logger.debug("%s Executing callback %s to process received data", self.__log_prefix__(), callback.__name__)
                    try:
                        status, data = callback(msg, *args, **kwargs)
                    except Exception as e:
                        logger.error('{lp} Exception raised. Check callback {callback_name}: "{message}" and it\'s return values for {msg_info}'.format(lp=self.__log_prefix__(), callback_name=callback.__name__, message=str(e), msg_info=self.__get_message_name__(msg.get_service_id(), msg.get_data_id())))
                        status = STATUS_CALLBACK_ERROR
                        data = None
            else:
                #
                # RESPONSE RECEIVED
                #
                if callback is None:
                    self.__buffer_received_data__(msg)
                else:
                    self.logger.debug("%s Executing callback %s to process received data", self.__log_prefix__(), callback.__name__)
                    try:
                        callback(msg, *args, **kwargs)
                    except Exception as e:
                        logger.error('{lp} Exception raised. Check callback {callback_name}: "{message}" for {msg_info}'.format(lp=self.__log_prefix__(), callback_name=callback.__name__, message=str(e), msg_info=self.__get_message_name__(msg.get_service_id(), msg.get_data_id())))
                return  # No response needed
        self.send(self.__sid_response_dict__[msg.get_service_id()], msg.get_data_id(), data, status=status)

    def __get_message_name__(self, service_id, data_id):
        return 'service: %s, data_id: %s' % (
            self.__sid_name_dict__.get(service_id, repr(service_id)),
            self.__did_name_dict__.get(service_id, {}).get(data_id, repr(data_id)),
        )
```

```python
    def __get_status_name__(self, status):
        return 'status: %s' % (self.__status_name_dict.get(status, 'unknown status: %s' % repr(status)))

    def __init_channel_name__(self, channel_name):
        self.__comm_inst__.init_channel_name(channel_name)
        self.__callbacks__.init_channel_name(channel_name)
        if channel_name is None:
            self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + self.DEFAULT_CHANNEL_NAME)
        else:
            self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + '.' + channel_name)

    def __log_prefix__(self):
        return 'prot-client:' if self.__comm_inst__.IS_CLIENT else 'prot-server:'

    def __mk_msg__(self, status, service_id, data_id, data):
        return data_storage({data_storage.KEY_DATA_ID: data_id, data_storage.KEY_SERVICE_ID: service_id, data_storage.KEY_STATUS: status, data_storage.KEY_DATA: data})

    def __status_log_lvl__(self, status):
        return STATUS_LOG_LVL.get(status, logging.CRITICAL)

    def add_data(self, service_id, data_id, name):
        """
        Method to add a name for a specific message.

        :param service_id: The Service-ID of the message. See class definitions starting with ``SID_``.
        :type service_id: int or list of ints
        :param data_id: The Data-ID of the message.
        :type data_id: int
        :param name: The Name for the transfered message.
        :type name: str
        """
        try:
            iter(service_id)
        except Exception:
            service_id = (service_id, )

        for sid in service_id:
            if sid not in self.__did_name_dict__:
                self.__did_name_dict__[sid] = {}
            self.__did_name_dict__[sid][data_id] = name

    def add_msg_to_auth_whitelist_(self, service_id, data_id):
        """
        Method to add a specific message to the list, where no authentification is required.

        :param service_id: The Service-ID of the message. See class definitions starting with ``SID_``.
        :type service_id: int
        :param data_id: The Data-ID of the message.
        :type data_id: int
        """
        if service_id not in self.__auth_whitelist__:
            self.__auth_whitelist__[service_id] = []
        self.__auth_whitelist__[service_id].append(data_id)
        logger.debug('%s Adding Message (%s) to the authentification whitelist', self.__log_prefix__(), self.__get_message_name__(service_id, data_id))

```

```
    def add_service(self, req_sid, resp_sid, req_name=None, resp_name=None):
        """
        Method to add a Service defined by Request- and Response Serivce-ID.

        :param req_sid: The Request Service-ID.
        :type req_sid: int
        :param resp_sid: The Response Service-ID.
        :type resp_sid: int
        """
        if req_sid in self.__sid_response_dict__:
            logger.error('%s Service with Request-SID=%d and Response-SID=%d not added, because request SID is already registered', self.__log_prefix__(), req_sid, resp_sid)
            raise RequestSidExistsError("Request for this Service is already registered")
        elif resp_sid in self.__sid_response_dict__.values():
            logger.error('%s Service with Request-SID=%d and Response-SID=%d not added, because response SID is already registered', self.__log_prefix__(), req_sid, resp_sid)
            raise ResponseSidExistsError("Response for this Service is already registered")
        else:
            self.__sid_response_dict__[req_sid] = resp_sid
            if req_name is not None:
                self.__sid_name_dict__[req_sid] = req_name
            if resp_name is not None:
                self.__sid_name_dict__[resp_sid] = resp_name
            logger.debug('%s Adding Service with Request=%s and Response=%s', self.__log_prefix__(), req_name or repr(req_sid), resp_name or repr(resp_sid))

    def add_status(self, status, name):
        """
        Method to add a name for a status.

        :param status: The Status. See class definitions starting with ``STATUS_``.
        :type status: int
        :param name: The Name for the Status.
        :type name: str
        """
        self.__status_name_dict[status] = name

    def authentificate(self, timeout=2):
        """
        This method authetificates the client at the server.

        :param timeout: The timeout for the authentification (requesting seed, sending key and getting authentification_feedback).
        :type timeout: float
        :returns: True, if authentification was successfull; False, if not.
        :rtype: bool

        .. note:: An authentification will only processed, if a secret had been given on initialisation.

        .. note:: Client and Server needs to use the same secret.
        """
        if self.__secret__ is not None:
            self.__authentification_state__ = AUTH_STATE_SEED_REQUESTED
            self.send(SID_AUTH_REQUEST, DID_AUTH_SEED, None)
            cnt = 0
            while cnt < timeout * 10:
                time.sleep(0.1)
                if self.__authentification_state__ == AUTH_STATE_TRUSTED_CONNECTION:
                    return True
                elif self.__authentification_state__ == AUTH_STATE_UNTRUSTED_CONNECTION:
                    break
                cnt += 1
        return False
```

```

    def check_authentification_state(self):
        """
        This Method return the Authitification State as boolean value.

        :return: True, if authentification state is okay, otherwise False
        :rtype: bool
        """
        return self.__secret__ is None or self.__authentification_state__ == AUTH_STATE_TRUSTED_CONNECTION

    def connection_established(self):
        """
        This Method returns the Connection state including authentification as a boolean value.

        :return: True, if the connection is established (incl. authentification, if a secret has been given)
        :rtype: bool
        """
        return self.is_connected() and (self.__secret__ is None or self.check_authentification_state())

    def is_connected(self):
        """
        This Methods returns Connection state of the Communication Instance :func:`comm_instance.is_connected`.

        :return: True if the :class:`comm_instance` is connected, otherwise False..
        :rtype: bool
        """
        return self.__comm_inst__.is_connected()

    def receive(self, service_id, data_id, timeout=1):
        """
        This Method returns a message object for a defined message or None, if this message is not available after the given timout.

        :param service_id: The Service-ID for the message. See class definitions starting with ``SID_``.
        :type service_id: int
        :param data_id: The Data-ID for the message.
        :type data_id: int
        :param timeout: The timeout for receiving.
        :type timeout: float
        :returns: The received data storage object or None, if no data was received.
        :rtype: data_storage
        """
        data = None
        cnt = 0
        while data is None and cnt < timeout * 10:
            try:
                data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
            except IndexError:
                data = None
            cnt += 1
            time.sleep(0.1)
        if data is None and cnt >= timeout * 10:
            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))
        return data

    def reconnect(self):
```

```
        """
        This methods initiates a reconnect by calling :func:`comm_instance.reconnect`.
        """
        return self.__comm_inst__.reconnect()

    def register_callback(self, service_id, data_id, callback, *args, **kwargs):
        """
        This method registers a callback for the given parameters. Giving ``None`` means, that all Service-IDs or all Data-IDs are used.
        If a message hitting these parameters has been received, the callback will be executed.

        :param service_id: The Service-ID for the message. See class definitions starting with ``SID_``.
        :type service_id: int
        :param data_id: The Data-ID for the message.
        :type data_id: int

        .. note:: The :func:`callback` is priorised in the following order:

            * Callbacks with defined Service-ID and Data-ID.
            * Callbacks with a defined Service-ID and all Data-IDs.
            * Callbacks with a defined Data-ID and all Service-IDs.
            * Unspecific Callbacks.

        .. note:: The :func:`callback` is executed with these arguments:

            **Parameters given at the callback call:**

            * The first Arguments is the received message as :class:`data_storage` object.
            * Further arguments given at registration.
            * Further keyword arguments given at registration.

            **Return value of the callback:**

            If the Callback is a Request Callback for a registered Service, the return value has to be a tuple or list with

            * :const:`response_status`: The response status (see class definitions starting with :const:`STA_*`.
            * :const:`response_data`: A JSON iterable object to be used as data for the response.

        .. note:: Only registered services will respond via the callbacks return values with the same data_id.
        """
        self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)

    def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2):
        """
        This methods sends out a message with the given content.

        :param service_id: The Service-ID for the message. See class definitions starting with ``SERVICE_``.
        :type service_id: int
        :param data_id: The Data-ID for the message.
        :type data_id: int
        :param data: The data to be transfered. The data needs to be json compatible.
        :type data: str
        :param status: The Status for the message. All requests should have ``STATUS_OKAY``.
        :type status: int
        :param timeout: The timeout for sending data (e.g. time to establish new connection).
        :type timeout: float
```

```python
        :return: True if data had been sent, otherwise False.
        :rtype: bool
        """
        if (self.check_authentification_state() or not self.__authentification_required__(service_id, data_id)) or (service_id in self.__sid_response_dict__.values() and status == STATUS_AUTH_REQUIRED and data is None):
            msg = data_storage(service_id=service_id, data_id=data_id, data=data, status=status)
            self.__log_msg__(msg, 'TX ->')
            return self.__comm_inst__.send(self.__build_frame__(msg), timeout=timeout)
        else:
            # Authentification required
            self.logger.warning("%s Authentification 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))
            return False


class struct_json_protocol(pure_json_protocol):
    """
    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.
    See also parent :py:class:`pure_json_protocol`.

    .. note::
        This class is depreceated and here for compatibility reasons (to support old clients or servers). Usage of :class:`pure_json_protocol` is recommended.
    """
    def __init__(self, *args, **kwargs):
        pure_json_protocol.__init__(self, *args, **kwargs)

    def __analyse_frame__(self, frame):
        status, service_id, data_id = struct.unpack('>III', frame[0:12])
        if sys.version_info >= (3, 0):
            data = json.loads(frame[12:-1].decode('utf-8'))
        else:
            data = json.loads(frame[12:-1])
        return self.__mk_msg__(status, service_id, data_id, data)

    def __build_frame__(self, msg):
        frame = struct.pack('>III', msg.get_status(), msg.get_service_id(), msg.get_data_id())
        if sys.version_info >= (3, 0):
            frame += bytes(json.dumps(msg.get_data()), 'utf-8')
            frame += self.__calc_chksum__(frame)
        else:
            frame += json.dumps(msg.get_data())
            frame += self.__calc_chksum__(frame)
        return frame

    def __calc_chksum__(self, raw_data):
        chksum = 0
        for b in raw_data:
            if sys.version_info >= (3, 0):
                chksum ^= b
            else:
                chksum ^= ord(b)
        if sys.version_info >= (3, 0):
            return bytes([chksum])
        else:
            return chr(chksum)

    def __check_frame_checksum__(self, frame):
        return self.__calc_chksum__(frame[:-1]) == frame[-1:]
```