

Unittest for socket_protocol

May 15, 2026

Contents

1	Test Information	4
1.1	Testobject Information	4
1.2	Testdata Information	4
1.3	Testsystem Information	4
2	Statistic	4
2.1	Test-Statistic for testrun: Library test	4
2.2	Coverage Statistic	5
3	Tested Requirements for testrun: Library test	6
3.1	Message Object	6
3.1.1	Status (REQ-0001)	6
3.1.2	Service-ID (REQ-0002)	6
3.1.3	Data-ID (REQ-0003)	7
3.1.4	Data (REQ-0004)	8
3.2	Communication	8
3.2.1	A full Message Object including the defined properties and data shall be transfered. (REQ-0005)	8
3.2.2	A checksum shall ensure the correct transmittion (REQ-0006)	9
3.2.3	An authentication between server and client shall be possible including status feedback methods (REQ-0007)	10
3.2.4	An automatic authentication shall available (REQ-0014)	11
3.2.5	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed. (REQ-0008)	12
3.2.6	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts (REQ-0009)	13
3.2.7	Define a channel name for the server and client after connection is established (REQ-0010)	14
3.2.8	The User shall be able to define a new service (REQ-0011)	15
3.2.9	Registration of already registered request Service-ID or response Service-ID shall not be possible (REQ-0012)	16
3.3	Callbacks	17
3.3.1	It shall be possible to register a callback for a specific Service- and Data-ID (REQ-0013)	17
3.3.2	It shall be possible to register a callback for a specific Service-ID and all Data-IDs (REQ-0015)	17

3.3.3	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs (REQ-0016)	18
3.3.4	It shall be possible to register a callback for all incoming messages (REQ-0017)	18
3.3.5	Callback choice, if several callbacks are available (caused by wildcard callbacks) (REQ-0018)	19
3.4	Some additional Information and Passthrough Methods	20
3.4.1	Connection established information (REQ-0020)	20
3.4.2	Is connected information (REQ-0021)	20
3.4.3	Reconnect Method (REQ-0022)	21
3.5	Deprepeated struct protocol	21
3.5.1	A full Message Object including the defined properties and data shall be transfered. (REQ-0023)	21

A Trace for testrun: Library test 23

A.1	Tests with status Info (22)	23
A.1.1	Status (REQ-0001)	23
A.1.2	Service-ID (REQ-0002)	23
A.1.3	Data-ID (REQ-0003)	24
A.1.4	Data (REQ-0004)	25
A.1.5	A full Message Object including the defined properties and data shall be transfered. (REQ-0005)	26
A.1.6	A checksum shall ensure the correct transmission (REQ-0006)	30
A.1.7	An authentication between server and client shall be possible including status feedback methods (REQ-0007)	34
A.1.8	An automatic authentication shall available (REQ-0014)	42
A.1.9	Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed. (REQ-0008)	47
A.1.10	A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts (REQ-0009)	55
A.1.11	Define a channel name for the server and client after connection is established (REQ-0010)	62
A.1.12	The User shall be able to define a new service (REQ-0011)	70
A.1.13	Registration of already registered request Service-ID or response Service-ID shall not be possible (REQ-0012)	74
A.1.14	It shall be possible to register a callback for a specific Service- and Data-ID (REQ-0013)	77
A.1.15	It shall be possible to register a callback for a specific Service-ID and all Data-IDs (REQ-0015)	82
A.1.16	It shall be possible to register a callback for a specific Data-IDs and all Service-IDs (REQ-0016)	86
A.1.17	It shall be possible to register a callback for all incoming messages (REQ-0017)	89

A.1.18	Callback choice, if several callbacks are available (caused by wildcard callbacks) (REQ-0018)	92
A.1.19	Connection established information (REQ-0020)	99
A.1.20	Is connected information (REQ-0021)	105
A.1.21	Reconnect Method (REQ-0022)	107
A.1.22	A full Message Object including the defined properties and data shall be transfered. (REQ-0023)	111
B	Test-Coverage	114
B.1	socket_protocol	114
B.1.1	socket_protocol.__init__.py	115

1 Test Information

1.1 Testobject Information

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

Information	
Name	socket_protocol
State	Released
Version	0.2.0
Dependencies	

1.2 Testdata Information

Information	
Version	0.2.0
Testruns	Library test

1.3 Testsystem Information

System Information	
Architecture	64bit
Distribution	Debian GNU/Linux 13 trixie
Hostname	ahorn
Kernel	6.12.86+deb13-amd64 (#1 SMP PREEMPT_DYNAMIC Debian 6.12.86-1 (2026-05-08))
Machine	x86_64
Path	/home/dirk/work/unittest_collection/socket_protocol/unittest/src
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun: Library test

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	19.279s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
socket_protocol	98.8%	98.9%
socket_protocol.__init__.py	98.8%	

3 Tested Requirements for testrun: Library test

3.1 Message Object

3.1.1 Status (REQ-0001)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:16,816
Finished-Time:	2026-05-15 15:05:16,818
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 (REQ-0002)

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:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:16,818
Finished-Time:	2026-05-15 15:05:16,819
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 (REQ-0003)**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:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:16,819
Finished-Time:	2026-05-15 15:05:16,819
Time-Consumption	0.001s

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

Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Info	Creating a maximum message object: {'data': 'D', 'data_id': 'DID', 'service_id': 'SID', 'status': 'S'}
Success	data_id is part of the message object is correct ('data_id' is in the list or dict).
Success	Content in message object for data_id is correct (Content 'DID' and Type is <class 'str'>).

3.1.4 Data (REQ-0004)

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:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py
(329)
Start-Time: 2026-05-15 15:05:16,820
Finished-Time: 2026-05-15 15:05:16,820
Time-Consumption 0.001s

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. (REQ-0005)

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:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:16,821
Finished-Time:	2026-05-15 15:05:17,574
Time-Consumption	0.753s

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

3.2.2 A checksum shall ensure the correct transmission (REQ-0006)**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:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:17,574

Finished-Time: 2026-05-15 15:05:18,536
 Time-Consumption 0.961s

Testsummary:

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

3.2.3 An authentication between server and client shall be possible including status feedback methods (REQ-0007)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

Testrun:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
 Start-Time: 2026-05-15 15:05:18,536
 Finished-Time: 2026-05-15 15:05:19,505
 Time-Consumption 0.969s

Testsummary:

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

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

3.2.4 An automatic authentication shall available (REQ-0014)

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

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

Testresult

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

Testrun:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
 Start-Time: 2026-05-15 15:05:19,506
 Finished-Time: 2026-05-15 15:05:22,213
 Time-Consumption 2.708s

Testsummary:

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

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

3.2.5 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed. (REQ-0008)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

Testrun:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
 Start-Time: 2026-05-15 15:05:22,214
 Finished-Time: 2026-05-15 15:05:24,189
 Time-Consumption 1.975s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting a Server secret and no Client secret
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on server side is correct (Content {'data_id': 36, 'service_id': 31, 'status': 3, 'data': None} and Type is <class 'socket_protocol.data_storage'>).
Info	Setting no Server secret but a Client secret
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content True and Type is <class 'bool'>).
Success	Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info	Identical secrets set
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).
Success	Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success	Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info	Performing Authentication

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

3.2.6 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts (REQ-0009)

Description

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

Reason for the implementation

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

Fitcriterion

Transmission 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:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
 Start-Time: 2026-05-15 15:05:24,190
 Finished-Time: 2026-05-15 15:05:26,672
 Time-Consumption 2.482s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Identical secrets set
Info	Transferring a message client → server
Success	Returnvalue of Client send Method is correct (Content False and Type is <class 'bool'>).
Success	Received message on server side is correct (Content None and Type is <class 'NoneType'>).
Info	Transferring a message server → client
Success	Returnvalue of Server send Method is correct (Content False and Type is <class 'bool'>).
Success	Received message on client side is correct (Content None and Type is <class 'NoneType'>).
Info	Added msg1 to client whitelist (sid=17, did=34)
Info	Transferring a message client → server

```

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

```

3.2.7 Define a channel name for the server and client after connection is established (REQ-0010)

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:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/___init___py
(329)

Start-Time: 2026-05-15 15:05:26,672
Finished-Time: 2026-05-15 15:05:28,422
Time-Consumption 1.749s

Testsummary:

Info	Setting up communication
Info	Connecting Server and Client
Info	Setting no Channel name for server and client
Success	Channel name of server is correct (Content None and Type is <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 (REQ-0011)

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:
Caller: /home/dirk/work/unititest_collection/socket_protocol/unititest/src/report/__init__.py
(329)
Start-Time: 2026-05-15 15:05:28,422

Finished-Time: 2026-05-15 15:05:29,284
Time-Consumption 0.862s

Testsummary:

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

3.2.9 Registration of already registered request Service-ID or response Service-ID shall not be possible (REQ-0012)

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:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time: 2026-05-15 15:05:29,285
Finished-Time: 2026-05-15 15:05:29,643
Time-Consumption 0.359s

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 (REQ-0013)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:29,644
Finished-Time:	2026-05-15 15:05:30,610
Time-Consumption	0.966s
Testsummary:	
Info	Setting up communication
Info	Connecting Server and Client
Info	Registering a correct working Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).
Info	Overwriting existing Callback using one with faulty (too many) return values
Info	Transferring data
Success	Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 2, 'data': None} and Type is <class 'socket_protocol.data_storage'>).
Info	Removing the registered Callback
Info	Transferring data
Success	Message stored inside callback is correct (Content None and Type is <class 'NoneType'>).
Success	Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 1, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

3.3.2 It shall be possible to register a callback for a specific Service-ID and all Data-IDs (REQ-0015)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:30,611
Finished-Time:	2026-05-15 15:05:31,171
Time-Consumption	0.560s
Testsummary:	
Info	Setting up communication

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

3.3.3 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs (REQ-0016)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:31,172
Finished-Time:	2026-05-15 15:05:31,733
Time-Consumption	0.561s

Testsummary:

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

3.3.4 It shall be possible to register a callback for all incoming messages (REQ-0017)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:31,733
Finished-Time:	2026-05-15 15:05:32,293
Time-Consumption	0.560s

Testsummary:

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

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

3.3.5 Callback choice, if several callbacks are available (caused by wildcard callbacks) (REQ-0018)

Testresult

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

Testrun:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
 Start-Time: 2026-05-15 15:05:32,294
 Finished-Time: 2026-05-15 15:05:33,464
 Time-Consumption 1.171s

Testsummary:

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

3.4 Some additional Information and Passthrough Methods

3.4.1 Connection established information (REQ-0020)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:33,465
Finished-Time:	2026-05-15 15:05:34,274
Time-Consumption	0.809s

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

3.4.2 Is connected information (REQ-0021)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:34,275
Finished-Time:	2026-05-15 15:05:34,633
Time-Consumption	0.358s

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 (REQ-0022)

Testresult

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

Testrun:	
Caller:	/home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time:	2026-05-15 15:05:34,634
Finished-Time:	2026-05-15 15:05:35,342
Time-Consumption	0.708s

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. (REQ-0023)

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:

Caller: /home/dirk/work/unittest_collection/socket_protocol/unittest/src/report/__init__.py (329)
Start-Time: 2026-05-15 15:05:35,342
Finished-Time: 2026-05-15 15:05:36,106
Time-Consumption 0.764s

Testsummary:

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

A Trace for testrun: Library test

A.1 Tests with status Info (22)

A.1.1 Status (REQ-0001)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

Info Creating empty message object: {'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'>)

A.1.2 Service-ID (REQ-0002)

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'>)

A.1.3 Data-ID (REQ-0003)**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'>)

A.1.4 Data (REQ-0004)**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'>)

A.1.5 A full Message Object including the defined properties and data shall be transfered. (REQ-0005)

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

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

Testresult

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

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

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

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

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

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

```

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

```

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

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

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

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

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

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

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

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

Info Transferring a message server → client

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

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

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

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

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

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

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

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

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

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

A.1.6 A checksum shall ensure the correct transmittion (REQ-0006)

Description

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

Reason for the implementation

Ensure correct data transfer.

Fitcriterion

Corrupted message is not in the receive buffer after transmission.

Testresult

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

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

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

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

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

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

```

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

```



```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

```
prot-server: Received message has an invalid checksum. Message will be ignored.
```

```
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↳ in buffer.
```

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

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

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

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

```
Result (Checksum Error -> No message received by server): None (<class 'NoneType'>)
```

```
Expectation (Checksum Error -> No message received by server): result = None (<class
↳ 'NoneType'>)
```

Info Transferring a message server → client

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

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

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

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

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

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

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

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

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

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

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

A.1.7 An authentication between server and client shall be possible including status feedback methods (REQ-0007)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

Info Setting up communication

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

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

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

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

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

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

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

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

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

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

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

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info No secret set

Info Performing Authentication

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

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

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

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

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

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

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

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

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

Info Different secrets set

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

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

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

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

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

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

Info Performing Authentication

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

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

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

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

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

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↪ "'b8874cec8d1160e576815d49db3260af4c7326b3f3641bb9d9a23c27a4c29e8a'"

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 38 38 37

```

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 38 38 37

comm-server: TX -> (64): 34 63 65 63 38 64 31 31 36 30 65 35 37 36 38 31 35 64 34 39 64 62 33
↳ 32 36 30 61 66 34 63 37 33 32 36 62 33 66 33 36 34 31 62 62 39 64 39 61 32 33 63 32 37 61
↳ 34 63 32 39 65 38 61 22 7d 8e 0b

comm-client: RX <- (64): 34 63 65 63 38 64 31 31 36 30 65 35 37 36 38 31 35 64 34 39 64 62 33
↳ 32 36 30 61 66 34 63 37 33 32 36 62 33 66 33 36 34 31 62 62 39 64 39 61 32 33 63 32 37 61
↳ 34 63 32 39 65 38 61 22 7d 8e 0b

comm-server: TX -> (4): f5 7d 3a 3e

comm-client: RX <- (4): f5 7d 3a 3e

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'b8874cec8d1160e576815d49db3260af4c7326b3f3641bb9d9a23c27a4c29e8a'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'5597436a6a5000cd18efbb95d8a6bbffda2bf913aaa832bcc85edc967a6d70cb377c5f5fd554782ffc4151'
↳ 3e1c9676d8a2238d6474d9ecf4065599a7c6c9124'"

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 35 35 39 37

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 35 35 39 37

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

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

comm-client: TX -> (64): 63 35 66 35 66 64 35 35 34 37 38 32 66 66 63 34 31 35 31 33 65 31 63
↳ 39 36 37 36 64 38 61 32 32 33 38 64 36 34 37 34 64 39 65 63 66 34 30 36 35 35 39 39 61 37
↳ 63 36 63 39 31 32 34 22 7d bc 91

comm-server: RX <- (64): 63 35 66 35 66 64 35 35 34 37 38 32 66 66 63 34 31 35 31 33 65 31 63
↳ 39 36 37 36 64 38 61 32 32 33 38 64 36 34 37 34 64 39 65 63 66 34 30 36 35 35 39 39 61 37
↳ 63 36 63 39 31 32 34 22 7d bc 91

comm-client: TX -> (4): a8 9c 3a 3e

comm-server: RX <- (4): a8 9c 3a 3e

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'5597436a6a5000cd18efbb95d8a6bbffda2bf913aaa832bcc85edc967a6d70cb377c5f5fd554782ffc4151'
↳ 3e1c9676d8a2238d6474d9ecf4065599a7c6c9124'"

prot-server: Executing callback __authenticate_check_key__ to process received data

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

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

```

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

```
comm-server: TX -> (7): 7d ea 0a 5c b4 3a 3e
```

```
comm-client: RX <- (7): 7d ea 0a 5c b4 3a 3e
```

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

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

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

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

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

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

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

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

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

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

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

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

Info Identical secrets set

Info Performing Authentication

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

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

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

```
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
```

```
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
```

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

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

```
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'056a123ceed7700f01bbd32f83fd649628d70c4d4bff6b2db7d6648d106b103f'"
```



```

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

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

comm-server: TX -> (64): 31 32 33 63 65 65 64 37 37 30 30 66 30 31 62 62 64 33 32 66 38 33 66
↳ 64 36 34 39 36 32 38 64 37 30 63 34 64 34 62 66 66 36 62 32 64 62 37 64 36 36 34 38 64 31
↳ 30 36 62 31 30 33 66 22 7d 18 ef

comm-client: RX <- (64): 31 32 33 63 65 65 64 37 37 30 30 66 30 31 62 62 64 33 32 66 38 33 66
↳ 64 36 34 39 36 32 38 64 37 30 63 34 64 34 62 66 66 36 62 32 64 62 37 64 36 36 34 38 64 31
↳ 30 36 62 31 30 33 66 22 7d 18 ef

comm-server: TX -> (4): 2c f6 3a 3e
comm-client: RX <- (4): 2c f6 3a 3e

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'056a123ceed7700f01bbd32f83fd649628d70c4d4bff6b2db7d6648d106b103f'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'be242eb5eaf3d8df509de4b816db81353c5331334b7d0aed0545462ff5a86903ced1e79ea2d87f04208c2a3_
↳ 1e3c26b572dbaaa709fb437c968aee0a92934d34a'"

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 62 65 32 34

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 62 65 32 34

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

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

comm-client: TX -> (64): 65 37 39 65 61 32 64 38 37 66 30 34 32 30 38 63 32 61 33 31 65 33 63
↳ 32 36 62 35 37 32 64 62 61 61 61 37 30 39 66 62 34 33 37 63 39 36 38 61 65 65 30 61 39 32
↳ 39 33 34 64 33 34 61 22 7d cf eb

comm-server: RX <- (64): 65 37 39 65 61 32 64 38 37 66 30 34 32 30 38 63 32 61 33 31 65 33 63
↳ 32 36 62 35 37 32 64 62 61 61 61 37 30 39 66 62 34 33 37 63 39 36 38 61 65 65 30 61 39 32
↳ 39 33 34 64 33 34 61 22 7d cf eb

comm-client: TX -> (4): f9 21 3a 3e
comm-server: RX <- (4): f9 21 3a 3e

prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'be242eb5eaf3d8df509de4b816db81353c5331334b7d0aed0545462ff5a86903ced1e79ea2d87f04208c2a3_
↳ 1e3c26b572dbaaa709fb437c968aee0a92934d34a'"

prot-server: Executing callback __authenticate_check_key__ to process received data

```

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

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

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

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

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

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

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

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

```
Result (Return Value of authentication method): True (<class 'bool'>)
```

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

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

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

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

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

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

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

Info Corrupting the authentication mechanism

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

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

Info Performing Authentication

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

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

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

```
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
```

```
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
```

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

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

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

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

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

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

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

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

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

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

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

A.1.8 An automatic authentication shall available (REQ-0014)

Description

An authentication is executed by the client on every connect.

Reason for the implementation

Simplify handling for authentication.

Fitcriterion

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

Testresult

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

Info Setting up communication

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

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

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

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

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

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

```

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

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

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

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

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

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

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

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

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

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

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

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

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

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

```

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Identical secrets set and automatic authentication

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

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

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

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

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

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

Info Connecting Server and Client

```
comm-client: Connection Lost...
```

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

```
comm-server: Connection Lost...
```

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

```
comm-client: Connection established...
```

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

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

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

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

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

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

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

```
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
```

```
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

comm-server: TX -> (6): 30 59 be 2f 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

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

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'ab698c0c3dfcaee495d1e61067450759667c238d4c1e0722ccce2fdfee97d77c'"

comm-client: RX <- (6): 30 59 be 2f 3a 3e

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 61 62 36 39

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 61 62 36 39

comm-server: TX -> (64): 38 63 30 63 33 64 66 63 61 65 65 34 39 35 64 31 65 36 31 30 36 37 34
↳ 35 30 37 35 39 36 36 37 63 32 33 38 64 34 63 31 65 30 37 32 32 63 63 63 65 32 66 64 66 65
↳ 65 39 37 64 37 37 63 22 7d 8a 09

comm-client: RX <- (64): 38 63 30 63 33 64 66 63 61 65 65 34 39 35 64 31 65 36 31 30 36 37 34
↳ 35 30 37 35 39 36 36 37 63 32 33 38 64 34 63 31 65 30 37 32 32 63 63 63 65 32 66 64 66 65
↳ 65 39 37 64 37 37 63 22 7d 8a 09

comm-server: TX -> (4): 58 f0 3a 3e

comm-client: RX <- (4): 58 f0 3a 3e

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'ab698c0c3dfcaee495d1e61067450759667c238d4c1e0722ccce2fdfee97d77c'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'8f1ed3914fed1b3e299f1e061515778d74afb337cd30f7bbde9e2b360b82eaf391724d48c36dcc3ed968b7d'
↳ ec6b905000f811f8d167f1b07591b386a77ef700e'"

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 38 66 31 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 38 66 31 65

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

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

```

```

comm-client: TX -> (64): 34 64 34 38 63 33 36 64 63 63 33 65 64 39 36 38 62 37 64 65 63 36 62
↪ 39 30 35 30 30 30 66 38 31 31 66 38 64 31 36 37 66 31 62 30 37 35 39 31 62 33 38 36 61 37
↪ 37 65 66 37 30 30 65 22 7d b8 49
comm-server: RX <- (64): 34 64 34 38 63 33 36 64 63 63 33 65 64 39 36 38 62 37 64 65 63 36 62
↪ 39 30 35 30 30 30 66 38 31 31 66 38 64 31 36 37 66 31 62 30 37 35 39 31 62 33 38 36 61 37
↪ 37 65 66 37 30 30 65 22 7d b8 49
comm-client: TX -> (4): aa c6 3a 3e
comm-server: RX <- (4): aa c6 3a 3e
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↪ "'8f1ed3914fed1b3e299f1e061515778d74afb337cd30f7bbde9e2b360b82eaf391724d48c36dcc3ed968b7d'
↪ ec6b905000f811f8d167f1b07591b386a77ef700e'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↪ "True"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 74 72 75 65 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 74 72 75 65 7d
comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↪ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback

```

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

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

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

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

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

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

A.1.9 Communication (rx and tx) shall be disabled, if a secret is given but no authentication had been successfully performed. (REQ-0008)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

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

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting a Server secret and no Client secret

Info Transferring a message client → server

```

prot-client: TX -> service: execute request, data_id: 36, status: okay, data:
↪ "'msg3_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 36 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 33 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73
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
prot-server: RX <- service: execute request, data_id: 36, status: okay, data:
↪ "'msg3_data_to_be_transferred'"
prot-server: Authentication is required. Just sending negative response.
prot-server: TX -> service: execute response, data_id: 36, status: authentication required,
↪ data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 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
comm-server: TX -> (8): 6c 7d 5d 78 af a4 3a 3e
comm-client: RX <- (8): 6c 7d 5d 78 af a4 3a 3e
prot-client: RX <- service: execute response, data_id: 36, status: authentication required,
↪ data: "None"

```

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

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

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

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

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

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

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

Info Setting no Server secret but a Client secret

Info Transferring a message server → client

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

```
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

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

```
prot-client: Authentication 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 Transferring a message client → server

prot-client: Authentication is required. TX-Message service: 17, data_id: 34, status: okay,
↪ data: 'msg1_data_to_be_transferred' will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↪ in buffer.

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

Result (Returnvalue of Client send Method): False (<class 'bool'>)

Expectation (Returnvalue of Client send Method): result = False (<class 'bool'>)

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

Result (Received message on server side): None (<class 'NoneType'>)

Expectation (Received message on server side): result = None (<class 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

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

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

Result (Returnvalue of Server send Method): False (<class 'bool'>)

Expectation (Returnvalue of Server send Method): result = False (<class 'bool'>)

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

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

Info Performing Authentication

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

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

```

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

comm-client: TX -> (6): fd 82 a2 a9 3a 3e

comm-server: RX <- (6): fd 82 a2 a9 3a 3e

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

prot-server: Executing callback __authenticate_create_seed__ to process received data

prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'95bae27bbbe2a1f2e0ccd9945b392da0a57f25c5ab6509a5d81534dcd4d75bf1'"

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 39 35 62 61

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 39 35 62 61

comm-server: TX -> (64): 65 32 37 62 62 62 65 32 61 31 66 32 65 30 63 63 64 39 39 34 35 62 33
↳ 39 32 64 61 30 61 35 37 66 32 35 63 35 61 62 36 35 30 39 61 35 64 38 31 35 33 34 64 63 64
↳ 34 64 37 35 62 66 31 22 7d e8 e6

comm-client: RX <- (64): 65 32 37 62 62 62 65 32 61 31 66 32 65 30 63 63 64 39 39 34 35 62 33
↳ 39 32 64 61 30 61 35 37 66 32 35 63 35 61 62 36 35 30 39 61 35 64 38 31 35 33 34 64 63 64
↳ 34 64 37 35 62 66 31 22 7d e8 e6

comm-server: TX -> (4): d3 6a 3a 3e

comm-client: RX <- (4): d3 6a 3a 3e

prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'95bae27bbbe2a1f2e0ccd9945b392da0a57f25c5ab6509a5d81534dcd4d75bf1'"

prot-client: Executing callback __authenticate_create_key__ to process received data

prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'6650df9fe1438ef58080c9487491bac0b24e16abea09f81c8ecbb2f0ab8d2f21a6a9dd1aa50e673f5024399'
↳ a3292c4da6362c16f39c61c6761d45768a06f85ed'"

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 36 35 30

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 36 35 30

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

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

comm-client: TX -> (64): 64 64 31 61 61 35 30 65 36 37 33 66 35 30 32 34 33 39 39 61 33 32 39
↳ 32 63 34 64 61 36 33 36 32 63 31 36 66 33 39 63 36 31 63 36 37 36 31 64 34 35 37 36 38 61
↳ 30 36 66 38 35 65 64 22 7d 9e 5c

```

```

comm-server: RX <- (64): 64 64 31 61 61 35 30 65 36 37 33 66 35 30 32 34 33 39 39 61 33 32 39
↳ 32 63 34 64 61 36 33 36 32 63 31 36 66 33 39 63 36 31 63 36 37 36 31 64 34 35 37 36 38 61
↳ 30 36 66 38 35 65 64 22 7d 9e 5c
comm-client: TX -> (4): 55 cd 3a 3e
comm-server: RX <- (4): 55 cd 3a 3e
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'6650df9fe1438ef58080c9487491bac0b24e16abea09f81c8ecbb2f0ab8d2f21a6a9dd1aa50e673f5024399'
↳ a3292c4da6362c16f39c61c6761d45768a06f85ed'"
prot-server: Executing callback __authenticate_check_key__ to process received data
prot-server: TX -> service: authentication response, data_id: key, status: okay, data:
↳ "True"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 31 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 74 72 75 65 7d
comm-server: TX -> (6): 94 fe 74 32 3a 3e
comm-client: RX <- (6): 94 fe 74 32 3a 3e
prot-client: RX <- service: authentication response, data_id: key, status: okay, data:
↳ "True"
prot-client: Executing callback __authenticate_process_feedback__ to process received data
prot-client: Got positive authentication feedback

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
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
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
prot-server: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

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

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

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

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

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

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

Info Transferring a message server → client

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

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73

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

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

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

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

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

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

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

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

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

A.1.10 A whitelist for communication (rx and tx) shall be available to enable communication for unauthorised counterparts (REQ-0009)

Description

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

Reason for the implementation

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

Fitcriterion

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

Testresult

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

Info Setting up communication

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
```

```
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
```

```
prot-server: Adding Service with Request=read data request and Response=read data response
```

```
prot-server: Adding Service with Request=write data request and Response=write data response
```

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

```
prot-server: Initialisation finished.
```

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

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e

```

```

comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Identical secrets set

Info Transferring a message client → server

```

prot-client: Authentification is required. TX-Message service: 17, data_id: 34, status: okay,
↪ data: 'msg1_data_to_be_transferred' will be ignored.
prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↪ in buffer.

```

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

```
Result (Returnvalue of Client send Method): False (<class 'bool'>)
```

```
Expectation (Returnvalue of Client send Method): result = False (<class 'bool'>)
```

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

```
Result (Received message on server side): None (<class 'NoneType'>)
```

```
Expectation (Received message on server side): result = None (<class 'NoneType'>)
```

Info Transferring a message server → client

```

prot-server: Authentification is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 35) not
↪ in buffer.

```

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

```
Result (Returnvalue of Server send Method): False (<class 'bool'>)
```

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

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

Result (Received message on client side): None (<class 'NoneType'>)

Expectation (Received message on client side): result = None (<class 'NoneType'>)

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

prot-client: Adding Message (service: 17, data_id: 34) to the authentication whitelist

Info Transferring a message client → server

prot-client: TX -> service: 17, data_id: 34, status: okay, data:

↪ "'msg1_data_to_be_transferred'"

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72

↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22

↪ 64 61 74 61 22 3a 3d 20 22 6d 73

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72

↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22

↪ 64 61 74 61 22 3a 3d 20 22 6d 73

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

prot-server: RX <- service: 17, data_id: 34, status: okay, data:

↪ "'msg1_data_to_be_transferred'"

prot-server: Authentication is required. Incoming message will be ignored.

prot-server: TIMEOUT (0.28705533596837945s): Requested data (service_id: 17; data_id: 34) not
↪ in buffer.

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

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

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

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

Result (Received message on server side): None (<class 'NoneType'>)

Expectation (Received message on server side): result = None (<class 'NoneType'>)

Info Transferring a message server → client

prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:

↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.

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

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

```
Result (Returnvalue of Server send Method): False (<class 'bool'>)
```

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

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

```
Result (Received message on client side): None (<class 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<class 'NoneType'>)
```

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

```
prot-server: Adding Message (service: 17, data_id: 34) to the authentication whitelist
```

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
```

```
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
```

```
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
```

```
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
```

```
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
```

```
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
```

```
↳ "'msg1_data_to_be_transferred'"
```

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

```
↳ method
```

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

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

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

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

```
Result (Received message on server side): {'data_id': 34, 'service_id': 17, 'status': 0,
```

```
↳ 'data': 'msg1_data_to_be_transferred'} (<class 'socket_protocol.data_storage'>)
```

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

Info Transferring a message server → client

```
prot-server: Authentication is required. TX-Message service: 17, data_id: 35, status:
↪ service or data unknown, data: 'msg2_data_to_be_transferred' will be ignored.
```

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

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

```
Result (Returnvalue of Server send Method): False (<class 'bool'>)
```

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

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

```
Result (Received message on client side): None (<class 'NoneType'>)
```

```
Expectation (Received message on client side): result = None (<class 'NoneType'>)
```

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

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

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

Info Transferring a message client → server

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 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
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
```

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

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

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

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

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

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

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

Info Transferring a message server → client

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

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 35 2c 20 22 73 65 72 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 34 2c 20 22 64 61 74 61 22 3a 3d 20 22 6d 73

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

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

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

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

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

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

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

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

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

A.1.11 Define a channel name for the server and client after connection is established (REQ-0010)

Description

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

Reason for the implementation

Structured logging by creating logger childs for each channel.

Fitcriterion

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

Testresult

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

Info Setting up communication

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
```

```
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
```

```
prot-server: Adding Service with Request=read data request and Response=read data response
```

```
prot-server: Adding Service with Request=write data request and Response=write data response
```

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



```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d

```

```

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Setting no Channel name for server and client

Success Channel name of server is correct (Content None and Type is <class 'NoneType'>).

Result (Channel name of server): None (<class 'NoneType'>)

Expectation (Channel name of server): result = None (<class 'NoneType'>)

Success Channel name of client is correct (Content None and Type is <class 'NoneType'>).

Result (Channel name of client): None (<class 'NoneType'>)

Expectation (Channel name of client): result = None (<class 'NoneType'>)

Info Setting different Channel names for client and Server

Info Connecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↪ "'client'"

```

comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 63 6c 69 65
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 22 63 6c 69 65
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).

Result (Channel name of server): 'client' (<class 'str'>)

Expectation (Channel name of server): result = 'client' (<class 'str'>)

Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).

Result (Channel name of client): 'client' (<class 'str'>)

Expectation (Channel name of client): result = 'client' (<class 'str'>)

Info Setting identical Channel names for client and server

Info Connecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

```

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'unittest'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 75 6e 69 74
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 75 6e 69 74
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'unittest' and Type is <class 'str'>).

Result (Channel name of server): 'unittest' (<class 'str'>)

Expectation (Channel name of server): result = 'unittest' (<class 'str'>)

Success Channel name of client is correct (Content 'unittest' and Type is <class 'str'>).

Result (Channel name of client): 'unittest' (<class 'str'>)

Expectation (Channel name of client): result = 'unittest' (<class 'str'>)

Info Setting Channel name for client only

Info Connecting Server and Client

```

comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data:
↳ "'client'"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 6c 69 65
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 22 63 6c 69 65
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
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Success Channel name of server is correct (Content 'client' and Type is <class 'str'>).

Result (Channel name of server): 'client' (<class 'str'>)

Expectation (Channel name of server): result = 'client' (<class 'str'>)

Success Channel name of client is correct (Content 'client' and Type is <class 'str'>).

Result (Channel name of client): 'client' (<class 'str'>)

Expectation (Channel name of client): result = 'client' (<class 'str'>)

Info Setting Channel name for server only

Info Connecting Server and Client

comm-client: Connection Lost...

prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-server: Connection Lost...

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

comm-client: Connection established...

comm-client: Cleaning up receive-buffer

prot-client: Cleaning up receive-buffer

prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"

comm-server: Connection established...

comm-server: Cleaning up receive-buffer

prot-server: Cleaning up receive-buffer

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

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

comm-client: TX -> (6): 53 5e 67 0b 3a 3e

comm-server: RX <- (6): 53 5e 67 0b 3a 3e

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

```

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
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'>)

```

A.1.12 The User shall be able to define a new service (REQ-0011)

Description

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

Reason for the implementation

Definition of Request and Response SIDs.

Fitcriterion

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

Testresult

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

Info Setting up communication

```

comm-client: Cleaning up receive-buffer
comm-server: Cleaning up receive-buffer
comm-server: Waiting for incoming connection
prot-server: Cleaning up receive-buffer
prot-server: Adding Service with Request=authentication request and
↳ Response=authentication response
prot-server: Adding Message (service: authentication request, data_id: seed) to the
↳ authentication whitelist
prot-server: Adding Message (service: authentication response, data_id: seed) to the
↳ authentication whitelist

```

```

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

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

prot-server: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-server: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-server: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-server: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

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

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

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

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-server: Adding Service with Request=read data request and Response=read data response
prot-server: Adding Service with Request=write data request and Response=write data response
prot-server: Adding Service with Request=execute request and Response=execute response
prot-server: Initialisation finished.

prot-client: Cleaning up receive-buffer

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

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

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

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

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

prot-client: Adding callback '__authenticate_create_seed__' for SID=0 and DID=0
prot-client: Adding callback '__authenticate_create_key__' for SID=1 and DID=0
prot-client: Adding callback '__authenticate_check_key__' for SID=0 and DID=1
prot-client: Adding callback '__authenticate_process_feedback__' for SID=1 and DID=1
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

prot-client: Adding Service with Request=channel name request and Response=channel name
↳ response

prot-client: Adding Message (service: channel name request, data_id: name) to the
↳ authentication whitelist

prot-client: Adding Message (service: channel name response, data_id: name) to the
↳ authentication whitelist

```



```

prot-client: Adding callback '__channel_name_request__' for SID=8 and DID=0
prot-client: Adding callback '__channel_name_response__' for SID=9 and DID=0
prot-client: Adding Service with Request=read data request and Response=read data response
prot-client: Adding Service with Request=write data request and Response=write data response
prot-client: Adding Service with Request=execute request and Response=execute response
prot-client: Initialisation finished.

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server → client

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↪ "'msg1_data_to_be_transferred'"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↪ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↪ 64 61 74 61 22 3a 3d 20 22 6d 73

```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

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

```
prot-client: TIMEOUT (0.28705533596837945s): Requested data (service_id: 18; data_id: 34) not
↳ in buffer.
```

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

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

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

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

```
Result (Received message on server side): None (<class 'NoneType'>)
```

```
Expectation (Received message on server side): result = None (<class 'NoneType'>)
```

Info Adding service to server instance for the transmit message

```
prot-server: Adding Service with Request=17 and Response=18
```

Info Transferring a message client → server → client

```
prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 37 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 22 6d 73
```

```
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
```

```
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
```

```
prot-server: Incoming message with no registered callback. Sending negative response.
```

```

prot-server: TX -> service: 18, data_id: 34, status: no callback for service, data buffered,
↳ data: "None"

comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c

comm-client: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 33 34 2c 20 22 73 65 72
↳ 76 69 63 65 5f 69 64 22 3a 3d 20 31 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 31 2c 20 22
↳ 64 61 74 61 22 3a 3d 20 6e 75 6c

comm-server: TX -> (8): 6c 7d bd 30 46 9b 3a 3e
comm-client: RX <- (8): 6c 7d bd 30 46 9b 3a 3e

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'>)
```

A.1.13 Registration of already registered request Service-ID or response Service-ID shall not be possible (REQ-0012)

Description

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

Reason for the implementation

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

Fitcriterion

Catch exception for registration of existing request and response SID.

Testresult

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

Info Setting up communication

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

```

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

```

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

```

```
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 (REQ-0013)

Testresult

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

Info Setting up communication

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

```

Info Connecting Server and Client

```
comm-client: Connection established...
```

```

comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0, 'data':
↪ 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33}
↪ (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↪ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Info Overwriting existing Callback using one with faulty (too many) return values

```

prot-server: Overwriting existing callback '__callback__' for service_id (10) and data_id (0)
↪ to '__callback_error__!'

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e

```

```

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
comm-server: TX -> (7): 7d a1 a2 87 f3 3a 3e
comm-client: RX <- (7): 7d a1 a2 87 f3 3a 3e
prot-client: RX <- service: read data response, data_id: 0, status: callback error, data:
↳ "None"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0, 'data':
↳ 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 2, 'data': None} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 2, 'data':
↳ None} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': None, 'data_id': 0, 'service_id':
↳ 11, 'status': 2} (<class 'socket_protocol.data_storage'>)

```

Info Removing the registered Callback

```

prot-server: Deleting existing callback '__callback_error__' for service_id (10) and data_id
↳ (0)!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8

```

```

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
prot-server: RX <- service: read data request, data_id: 0, status: okay, data: "31"
prot-server: Incoming message with no registered callback. Sending negative response.
prot-server: TX -> service: read data response, data_id: 0, status: no callback for service,
↳ data buffered, data: "None"
comm-server: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 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
comm-server: TX -> (7): 7d 88 6a 33 01 3a 3e
comm-client: RX <- (7): 7d 88 6a 33 01 3a 3e
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'>)

A.1.15 It shall be possible to register a callback for a specific Service-ID and all Data-IDs (REQ-0015)

Testresult

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

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

```

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

```

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

prot-server: Adding callback '__callback__' for SID=10 and DID=None

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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'>)

A.1.16 It shall be possible to register a callback for a specific Data-IDs and all Service-IDs (REQ-0016)**Testresult**

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

Info Setting up communication

```

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

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```

prot-server: Adding callback '__callback__' for SID=None and DID=0

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
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'>)

A.1.17 It shall be possible to register a callback for all incoming messages (REQ-0017)

Testresult

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

Info Setting up communication

comm-client: Cleaning up receive-buffer

comm-server: Cleaning up receive-buffer

comm-server: Waiting for incoming connection

prot-server: Cleaning up receive-buffer

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

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

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

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

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

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

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

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

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

prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION

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

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

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

prot-server: Adding callback '__channel_name_request__' for SID=8 and DID=0

prot-server: Adding callback '__channel_name_response__' for SID=9 and DID=0

```

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

```

Info Connecting Server and Client

```

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

```

```

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

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

comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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

comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Registering a correct working Callback

```
prot-server: Adding callback '__callback__' for SID=None and DID=None
```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8

comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8

comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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

```

```
comm-server: TX -> (5): e1 8c bb 3a 3e
```

```
comm-client: RX <- (5): e1 8c bb 3a 3e
```

```
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'>)
```

A.1.18 Callback choice, if several callbacks are available (caused by wildcard callbacks) (REQ-0018)

Testresult

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

Info Setting up communication

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

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

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

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

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

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

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

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

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

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

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

```

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

```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data
```

Info Registering all kind of Callbacks

```
prot-server: Adding callback '__callback3__' for SID=None and DID=None
prot-server: Adding callback '__callback2__' for SID=None and DID=0
prot-server: Adding callback '__callback1__' for SID=10 and DID=None
prot-server: Adding callback '__callback__' for SID=10 and DID=0
```

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
```

```

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): e1 8c bb 3a 3e
comm-client: RX <- (5): e1 8c bb 3a 3e
prot-client: RX <- service: read data response, data_id: 0, status: okay, data: "33"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↪ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0, 'data':
↪ 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↪ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 0, 'data': 33}
↪ (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 33, 'data_id': 0, 'service_id':
↪ 11, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Info Removing Callback for a specific Data- and Service-ID

```

prot-server: Deleting existing callback '__callback__' for service_id (10) and data_id (0)!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"

```



```

comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): 62 51 ca 3a 3e
comm-client: RX <- (5): 62 51 ca 3a 3e
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↳ data: "34"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0, 'data':
↳ 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 34} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 34}
↳ (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 34, 'data_id': 0, 'service_id':
↳ 11, 'status': 6} (<class 'socket_protocol.data_storage'>)

```

Info Removing Callback for a specific Service-ID and all Data-IDs

```

prot-server: Deleting existing callback '__callback1__' for service_id (10) and data_id
↳ (None)!

```

Info Transferring data

```

prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↳ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): 79 60 8b 3a 3e
comm-client: RX <- (5): 79 60 8b 3a 3e
prot-client: RX <- service: read data response, data_id: 0, status: operation not permitted,
↳ data: "35"
prot-client: Message data is stored in buffer and is now ready to be retrieved by receive
↳ method

```

Success Message stored inside callback is correct (Content {'data_id': 0, 'service_id': 10, 'status': 0, 'data': 31} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message stored inside callback): {'data_id': 0, 'service_id': 10, 'status': 0, 'data':
↳ 31} (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message stored inside callback): result = {'data': 31, 'data_id': 0,
↳ 'service_id': 10, 'status': 0} (<class 'socket_protocol.data_storage'>)

```

Success Message received by client is correct (Content {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 35} and Type is <class 'socket_protocol.data_storage'>).

```

Result (Message received by client): {'data_id': 0, 'service_id': 11, 'status': 6, 'data': 35}
↳ (<class 'socket_protocol.data_storage'>)

```

```

Expectation (Message received by client): result = {'data': 35, 'data_id': 0, 'service_id':
↳ 11, 'status': 6} (<class 'socket_protocol.data_storage'>)

```

Info Removing Callback for a specific Data-ID and all Serice-IDs

prot-server: Deleting existing callback '__callback2__' for service_id (None) and data_id (0)!

Info Transferring data

```
prot-client: TX -> service: read data request, data_id: 0, status: okay, data: "31"
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 31 30 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64
↪ 61 74 61 22 3a 3d 20 33 31 7d b8
comm-client: TX -> (5): 5b f5 78 3a 3e
comm-server: RX <- (5): 5b f5 78 3a 3e
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
comm-server: TX -> (5): 96 78 fe 3a 3e
comm-client: RX <- (5): 96 78 fe 3a 3e
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'>)
```

A.1.19 Connection established information (REQ-0020)**Testresult**

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

Info Setting up communication

```

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

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

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

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

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client connection status): True (<class 'bool'>)
```

```
Expectation (Client connection status): result = True (<class 'bool'>)
```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server connection status): True (<class 'bool'>)
```

```
Expectation (Server connection status): result = True (<class 'bool'>)
```

Success Client connection status is correct (Content False and Type is <class 'bool'>).

```
comm-client: Connection Lost...
```

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

```
comm-server: Connection Lost...
```

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

```
Result (Client connection status): False (<class 'bool'>)
```

```
Expectation (Client connection status): result = False (<class 'bool'>)
```

Success Server connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server connection status): False (<class 'bool'>)
```

```
Expectation (Server connection status): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
```

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

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

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

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

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

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

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

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

```
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
```

```
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
```

```
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
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

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

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

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client connection status): True (<class 'bool'>)
```

```
Expectation (Client connection status): result = True (<class 'bool'>)
```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server connection status): True (<class 'bool'>)
```

```
Expectation (Server connection status): result = True (<class 'bool'>)
```

Info Adding secrets to socket_protocol

Success Client connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client connection status): False (<class 'bool'>)
```

```
Expectation (Client connection status): result = False (<class 'bool'>)
```

Success Server connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server connection status): False (<class 'bool'>)
```

```
Expectation (Server connection status): result = False (<class 'bool'>)
```

Info Doing authentication

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

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

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

```
comm-client: TX -> (6): fd 82 a2 a9 3a 3e
```

```
comm-server: RX <- (6): fd 82 a2 a9 3a 3e
```

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

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

```
prot-server: TX -> service: authentication response, data_id: seed, status: okay, data:
↳ "'75dd2114ae67800eb92f4cd9e24aa587a8ba11e69143b4c63b710244e3db562f'"
```

```
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 37 35 64 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 37 35 64 64
```

```
comm-server: TX -> (64): 32 31 31 34 61 65 36 37 38 30 30 65 62 39 32 66 34 63 64 39 65 32 34
↳ 61 61 35 38 37 61 38 62 61 31 31 65 36 39 31 34 33 62 34 63 36 33 62 37 31 30 32 34 34 65
↳ 33 64 62 35 36 32 66 22 7d b5 80
```

```
comm-client: RX <- (64): 32 31 31 34 61 65 36 37 38 30 30 65 62 39 32 66 34 63 64 39 65 32 34
↳ 61 61 35 38 37 61 38 62 61 31 31 65 36 39 31 34 33 62 34 63 36 33 62 37 31 30 32 34 34 65
↳ 33 64 62 35 36 32 66 22 7d b5 80
```

```
comm-server: TX -> (4): af 10 3a 3e
```

```
comm-client: RX <- (4): af 10 3a 3e
```

```
prot-client: RX <- service: authentication response, data_id: seed, status: okay, data:
↳ "'75dd2114ae67800eb92f4cd9e24aa587a8ba11e69143b4c63b710244e3db562f'"
```

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

```
prot-client: TX -> service: authentication request, data_id: key, status: okay, data:
↳ "'c4e2b829620bf5622d81ff6160857c209d4e6e9c1d590ecacd7dbe16143c15026607cac7229e8c7918344d4_
↳ f04c9f67e61b22c3bf19b061a6f5c272f52625152'"
```

```
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 34 65 32
```

```
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 34 65 32
```



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

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

```
comm-client: TX -> (64): 63 61 63 37 32 32 39 65 38 63 37 39 31 38 33 34 34 64 34 66 30 34 63
↳ 39 66 36 37 65 36 31 62 32 32 63 33 62 66 31 39 62 30 36 31 61 36 66 35 63 32 37 32 66 35
↳ 32 36 32 35 31 35 32 22 7d 28 4a
```

```
comm-server: RX <- (64): 63 61 63 37 32 32 39 65 38 63 37 39 31 38 33 34 34 64 34 66 30 34 63
↳ 39 66 36 37 65 36 31 62 32 32 63 33 62 66 31 39 62 30 36 31 61 36 66 35 63 32 37 32 66 35
↳ 32 36 32 35 31 35 32 22 7d 28 4a
```

```
comm-client: TX -> (4): 89 81 3a 3e
```

```
comm-server: RX <- (4): 89 81 3a 3e
```

```
prot-server: RX <- service: authentication request, data_id: key, status: okay, data:
↳ "'c4e2b829620bf5622d81ff6160857c209d4e6e9c1d590ecacd7dbe16143c15026607cac7229e8c7918344d4_'
↳ f04c9f67e61b22c3bf19b061a6f5c272f52625152'"
```

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

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

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

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

```
comm-server: TX -> (6): 94 fe 74 32 3a 3e
```

```
comm-client: RX <- (6): 94 fe 74 32 3a 3e
```

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

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

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

Success Client connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client connection status): True (<class 'bool'>)
```

```
Expectation (Client connection status): result = True (<class 'bool'>)
```

Success Server connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server connection status): True (<class 'bool'>)
```

```
Expectation (Server connection status): result = True (<class 'bool'>)
```

A.1.20 Is connected information (REQ-0021)**Testresult**

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

Info Setting up communication

```

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

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↳ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↳ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
```

```
comm-server: TX -> (6): 30 59 be 2f 3a 3e
```

```
comm-client: RX <- (6): 30 59 be 2f 3a 3e
```

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

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

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): True (<class 'bool'>)
```

```
Expectation (Client Communication instance connection status): result = True (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): True (<class 'bool'>)
```

```
Expectation (Server Communication instance connection status): result = True (<class 'bool'>)
```

Info Disconnecting Server and Client

```
comm-client: Connection Lost...
```

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

```
comm-server: Connection Lost...
```

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

Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): False (<class 'bool'>)
```

```
Expectation (Client Communication instance connection status): result = False (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): False (<class 'bool'>)
```

```
Expectation (Server Communication instance connection status): result = False (<class 'bool'>)
```

A.1.21 Reconnect Method (REQ-0022)

Testresult

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

Info Setting up communication

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

```

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

```

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
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
prot-server: Cleaning up receive-buffer
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
comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e

```

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

Success Client Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): True (<class 'bool'>)
Expectation (Client Communication instance connection status): result = True (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content True and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): True (<class 'bool'>)
Expectation (Server Communication instance connection status): result = True (<class 'bool'>)
```

Info Disconnecting Server and Client

```
comm-client: Connection Lost...
prot-client: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
comm-server: Connection Lost...
prot-server: Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION
```

Success Client Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Client Communication instance connection status): False (<class 'bool'>)
Expectation (Client Communication instance connection status): result = False (<class 'bool'>)
```

Success Server Communication instance connection status is correct (Content False and Type is <class 'bool'>).

```
Result (Server Communication instance connection status): False (<class 'bool'>)
Expectation (Server Communication instance connection status): result = False (<class 'bool'>)
```

Info Connecting Server and Client

```
comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
prot-server: Cleaning up receive-buffer
comm-client: TX -> (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
comm-server: RX <- (64): 3a 3c 7b 22 64 61 74 61 5f 69 64 22 3a 3d 20 30 2c 20 22 73 65 72 76
↪ 69 63 65 5f 69 64 22 3a 3d 20 38 2c 20 22 73 74 61 74 75 73 22 3a 3d 20 30 2c 20 22 64 61
↪ 74 61 22 3a 3d 20 6e 75 6c 6c 7d
```

```

comm-client: TX -> (6): 53 5e 67 0b 3a 3e
comm-server: RX <- (6): 53 5e 67 0b 3a 3e
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
comm-server: TX -> (6): 30 59 be 2f 3a 3e
comm-client: RX <- (6): 30 59 be 2f 3a 3e
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'>)
```

A.1.22 A full Message Object including the defined properties and data shall be transfered. (REQ-0023)

Description

Every Communication shall transfer a complete message with its content.

Reason for the implementation

See Reasons for every single information of the Message Object.

Fitcriterion

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

Testresult

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

Info Setting up communication

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

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

```

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

```

```

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

```

Info Connecting Server and Client

```

comm-client: Connection established...
comm-client: Cleaning up receive-buffer
prot-client: Cleaning up receive-buffer
prot-client: TX -> service: channel name request, data_id: name, status: okay, data: "None"
comm-server: Connection established...
comm-server: Cleaning up receive-buffer
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
prot-server: Cleaning up receive-buffer
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
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
prot-client: RX <- service: channel name response, data_id: name, status: okay, data: "None"
prot-client: Executing callback __channel_name_response__ to process received data

```

Info Transferring a message client → server

```

prot-client: TX -> service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"
comm-client: TX -> (45): 3a 3c 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
comm-server: RX <- (45): 3a 3c 00 00 00 00 00 00 00 11 00 00 00 22 22 6d 73 67 31 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7d 3a 3e
prot-server: RX <- service: 17, data_id: 34, status: okay, data:
↳ "'msg1_data_to_be_transferred'"

```

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

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

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

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

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

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

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

Info Transferring a message server → client

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

```
comm-server: TX -> (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e
```

```
comm-client: RX <- (45): 3a 3c 00 00 00 04 00 00 00 11 00 00 00 23 22 6d 73 67 32 5f 64 61 74
↳ 61 5f 74 6f 5f 62 65 5f 74 72 61 6e 73 66 65 72 65 64 22 7b 3a 3e
```

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

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

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

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

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

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

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

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

B Test-Coverage

B.1 socket_protocol

The line coverage for socket_protocol was 98.8%

The branch coverage for socket_protocol was 98.9%

B.1.1 socket_protocol.__init__.py

The line coverage for socket_protocol.__init__.py was 98.8%

The branch coverage for socket_protocol.__init__.py was 98.9%

```

1  """
2  socket_protocol (Socket Protocol)
3  =====
4
5  **Author:**
6
7  * Dirk Alders <sudo-dirk@mount-mockery.de>
8
9  **Description:**
10
11     This Module supports point to point communication for client-server issues.
12
13  **Submodules:**
14
15  * :class:`socket_protocol.data_storage`
16  * :class:`socket_protocol.pure_json_protocol`
17  * :class:`socket_protocol.struct_json_protocol`
18
19  **Unittest:**
20
21     See also the :download:`unittest <socket_protocol/_testresults_/unittest.pdf>`
22     documentation.
23
24  **Module Documentation:**
25
26  """
27  __VERSION__ = "0.2.0"
28  __DEPENDENCIES__ = []
29
30  import binascii
31  import hashlib
32  import json
33  import logging
34  import os
35  import struct
36  import time
37
38  try:
39     from config import APP_NAME as ROOT_LOGGER_NAME
40 except ImportError:
41     ROOT_LOGGER_NAME = "root"
42 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
43
44
45 __DESCRIPTION__ = """The Module {\\tt %s} is designed for point to point communication for client
46 -server issues.
47 For more Information read the sphinx documentation.""" % __name__.replace("_", "\\_")
48 """The Module Description"""
49
50 SID_AUTH_REQUEST = 0
51 """SID for authentication request"""
52 SID_AUTH_RESPONSE = 1
53 """SID for authentication response"""
54 DID_AUTH_SEED = 0
55 """DID for authentication (seed)"""

```

```

55 DID_AUTH_KEY = 1
56 """DID for authentication (key)"""
57 SID_CHANNEL_NAME_REQUEST = 8
58 """SID for channel name exchange request """
59 SID_CHANNEL_NAME_RESPONSE = 9
60 """SID for channel name exchange response"""
61 DID_CHANNEL_NAME = 0
62 """DID for channel name """
63 SID_READ_REQUEST = 10
64 """SID for a read data request"""
65 SID_READ_RESPONSE = 11
66 """SID for read data response"""
67 SID_WRITE_REQUEST = 20
68 """SID for a write data request"""
69 SID_WRITE_RESPONSE = 21
70 """SID for a write data response"""
71 SID_EXECUTE_REQUEST = 30
72 """SID for a execute request"""
73 SID_EXECUTE_RESPONSE = 31
74 """SID for a execute response"""
75
76 STATUS_OKAY = 0
77 """Status for 'okay'"""
78 STATUS_BUFFERING_UNHANDLED_REQUEST = 1
79 """Status for 'unhandled request'"""
80 STATUS_CALLBACK_ERROR = 2
81 """Status for 'callback errors'"""
82 STATUS_AUTH_REQUIRED = 3
83 """Status for 'authentication is required'"""
84 STATUS_SERVICE_OR_DATA_UNKNOWN = 4
85 """Status for 'service or data unknown'"""
86 STATUS_CHECKSUM_ERROR = 5
87 """Status for 'checksum error'"""
88 STATUS_OPERATION_NOT_PERMITTED = 6
89 """Status for 'operation not permitted'"""
90 STATUS_LOG_LVL = {
91     STATUS_OKAY: logging.INFO,
92     STATUS_BUFFERING_UNHANDLED_REQUEST: logging.WARNING,
93     STATUS_CALLBACK_ERROR: logging.ERROR,
94     STATUS_AUTH_REQUIRED: logging.WARNING,
95     STATUS_SERVICE_OR_DATA_UNKNOWN: logging.ERROR,
96     STATUS_CHECKSUM_ERROR: logging.ERROR,
97     STATUS_OPERATION_NOT_PERMITTED: logging.WARNING,
98 }
99 """Status depending log level for messages"""
100
101 AUTH_STATE_UNTRUSTED_CONNECTION = 0
102 """Authentication Status for an 'Untrusted Connection'"""
103 AUTH_STATE_SEED_REQUESTED = 1
104 """Authentication Status for 'Seed was requested'"""
105 AUTH_STATE_SEED_TRANSFERRED = 2
106 """Authentication Status for 'Seed has been sent'"""
107 AUTH_STATE_KEY_TRANSFERRED = 3
108 """Authentication Status for 'Key has been sent'"""
109 AUTH_STATE_TRUSTED_CONNECTION = 4
110 """Authentication Status for a 'Trusted Connection'"""
111 AUTH_STATE_NAMES = {
112     AUTH_STATE_UNTRUSTED_CONNECTION: "Untrusted Connection",
113     AUTH_STATE_SEED_REQUESTED: "Seed was requested",
114     AUTH_STATE_SEED_TRANSFERRED: "Seed has been sent",
115     AUTH_STATE_KEY_TRANSFERRED: "Key has been sent",
116     AUTH_STATE_TRUSTED_CONNECTION: "Trusted Connection",
117 }

```

```

118 """ Authentication Status names for previous defined authentication states """
119
120
121 class RequestSidExistsError(Exception):
122     pass
123
124
125 class ResponseSidExistsError(Exception):
126     pass
127
128
129 class _callback_storage(dict):
130     DEFAULT_CHANNEL_NAME = "all_others"
131
132     def __init__(self, channel_name, log_prefix):
133         self.init_channel_name(channel_name)
134         self.__log_prefix__ = log_prefix
135         dict.__init__(self)
136
137     def init_channel_name(self, channel_name):
138         if channel_name is None:
139             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + "." + self.
140             DEFAULT_CHANNEL_NAME)
141         else:
142             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + "." +
143             channel_name)
144
145     def get(self, service_id, data_id):
146         if dict.get(self, service_id, {}).get(data_id, None) is not None:
147             return self[service_id][data_id]
148         elif dict.get(self, service_id, {}).get(None, None) is not None:
149             return self[service_id][None]
150         elif dict.get(self, None, {}).get(data_id, None) is not None:
151             return self[None][data_id]
152         elif dict.get(self, None, {}).get(None, None) is not None:
153             return self[None][None]
154         else:
155             return (None, None, None)
156
157     def add(self, service_id, data_id, callback, *args, **kwargs):
158         cb_data = self.get(service_id, data_id)
159         if dict.get(self, service_id, {}).get(data_id, None) is not None:
160             if callback is None:
161                 self.logger.warning(
162                     "%s Deleting existing callback %s for service_id (%s) and data_id (%s)!",
163                     self.__log_prefix__(),
164                     repr(cb_data[0].__name__),
165                     repr(service_id),
166                     repr(data_id),
167                 )
168                 del self[service_id][data_id]
169                 return
170             else:
171                 self.logger.warning(
172                     "%s Overwriting existing callback %s for service_id (%s) and data_id (%s) to
173                     %s!",
174                     self.__log_prefix__(),
175                     repr(cb_data[0].__name__),
176                     repr(service_id),
177                     repr(data_id),
178                     repr(callback.__name__),
179                 )
180         else:
181             self[service_id][data_id] = callback

```

```

178         self.logger.debug(
179             "%s Adding callback %s for SID=%s and DID=%s", self.__log_prefix__(), repr(
180                 callback.__name__), repr(service_id), repr(data_id)
181             )
182         if service_id not in self:
183             self[service_id] = {}
184             self[service_id][data_id] = (callback, args, kwargs)
185
186 class data_storage(dict):
187     """
188     This is a storage object for socket_protocol messages.
189
190     :param status: The message status.
191     :type status: int
192     :param service_id: The Service-ID.
193     :type service_id: int
194     :param data_id: The Data-ID.
195     :type data_id: int
196     :param data: The transfered data.
197     :type data: any
198     """
199
200     KEY_STATUS = "status"
201     KEY_SERVICE_ID = "service_id"
202     KEY_DATA_ID = "data_id"
203     KEY_DATA = "data"
204     ALL_KEYS = [KEY_DATA, KEY_DATA_ID, KEY_SERVICE_ID, KEY_STATUS]
205
206     def __init__(self, *args, **kwargs):
207         dict.__init__(self, *args, **kwargs)
208         for key in self.ALL_KEYS:
209             if key not in self:
210                 self[key] = None
211
212     def get_status(self, default=None):
213         """
214         This Method returns the message status.
215
216         :param default: The default value, if no data is available.
217         """
218         return self.get(self.KEY_STATUS, default)
219
220     def get_service_id(self, default=None):
221         """
222         This Method returns the message Service-ID.
223
224         :param default: The default value, if no data is available.
225         """
226         return self.get(self.KEY_SERVICE_ID, default)
227
228     def get_data_id(self, default=None):
229         """
230         This Method returns the message Data-ID.
231
232         :param default: The default value, if no data is available.
233         """
234         return self.get(self.KEY_DATA_ID, default)
235
236     def get_data(self, default=None):

```

```

237     """
238     This Method returns the message data.
239
240     :param default: The default value, if no data is available.
241     """
242     return self.get(self.KEY_DATA, default)
243
244
245 class pure_json_protocol(object):
246     """
247     This `class` supports to transfer a message and it's data.
248
249     :param comm_instance: A communication instance.
250     :type comm_instance: instance
251     :param secret: An optional secret (e.g. created by ``binascii.hexlify(os.urandom(24))``).
252     :type secret: str
253     :param auto_auth: An optional parameter to enable (True) automatic authentication,
254                       otherwise you need to do it manually, if needed.
255     :type auto_auth: bool
256     :param channel_name: An optional parameter to set a channel name for logging of the
257                           communication.
258     :type channel_name: str
259
260     .. hint::
261
262         * The Service-ID is designed to identify the type of the communication (e.g. :const:`
263         READ_REQUEST`, :const:`WRITE_REQUEST`, :const:`READ_RESPONSE`, :const:`WRITE_RESPONSE`, ...)
264         * The Data-ID is designed to identify the requests / responses using the same Service-ID.
265
266     .. note:: The :class:`comm_instance` needs to have at least the following interface:
267
268         * A Method :func:`comm_instance.init_channel_name` to set the channel name.
269         * A Constant :const:`comm_instance.IS_CLIENT` to identify that the :class:`comm_instance`
270           is a client (True) or a server (False).
271         * A Method :func:`comm_instance.is_connected` to identify if the instance is connected (
272           True) or not (False).
273         * A Method :func:`comm_instance.reconnect` to initiate a reconnect.
274         * A Method :func:`comm_instance.register_callback` to register a data available callback.
275         * A Method :func:`comm_instance.register_connect_callback` to register a connect callback
276           .
277         * A Method :func:`comm_instance.register_disconnect_callback` to register a disconnect
278           callback.
279         * A Method :func:`comm_instance.send` to send data via the :class:`comm_instance`.
280
281     .. note:: The parameter :const:`auto_auth` is only relevant, if a secret is given and the :
282               class:`comm_instance` is a client. The authentication is initiated directly after the
283               connection is established.
284
285     .. note:: The :const:`channel_name`-exchange will be initiated by the client directly after
286               the the connection is established.
287
288         * If a channel_name is given at both communication sides and they are different, the
289           client name is taken over and the server will log a warning message.
290
291     **Example:**
292
293     .. literalinclude:: socket_protocol/_examples_/socket_protocol_example.py
294
295     Will result to the following output:
296
297     .. literalinclude:: socket_protocol/_examples_/socket_protocol_example.log

```



```

287 """
288
289 DEFAULT_CHANNEL_NAME = "all others"
290
291 def __init__(self, comm_instance, secret=None, auto_auth=False, channel_name=None):
292     self.__comm_inst__ = comm_instance
293     self.__secret__ = secret
294     self.__auto_auth__ = auto_auth
295     #
296     self.__auth_whitelist__ = {}
297     self.__sid_response_dict__ = {}
298     self.__sid_name_dict__ = {}
299     self.__did_name_dict__ = {}
300     #
301     self.__callbacks__ = _callback_storage(channel_name, self.__log_prefix__)
302     self.__init_channel_name__(channel_name)
303     #
304     self.__status_name_dict__ = {}
305     self.add_status(STATUS_OKAY, "okay")
306     self.add_status(STATUS_BUFFERING_UNHANDLED_REQUEST, "no callback for service, data
buffered")
307     self.add_status(STATUS_CALLBACK_ERROR, "callback error")
308     self.add_status(STATUS_AUTH_REQUIRED, "authentication required")
309     self.add_status(STATUS_SERVICE_OR_DATA_UNKNOWN, "service or data unknown")
310     self.add_status(STATUS_CHECKSUM_ERROR, "checksum error")
311     self.add_status(STATUS_OPERATION_NOT_PERMITTED, "operation not permitted")
312     #
313     self.__clean_receive_buffer__()
314
315     self.add_service(SID_AUTH_REQUEST, SID_AUTH_RESPONSE, "authentication request", "
authentication response")
316     self.add_data((SID_AUTH_REQUEST, SID_AUTH_RESPONSE), DID_AUTH_SEED, "seed")
317     self.add_data(SID_AUTH_REQUEST, DID_AUTH_KEY, "key")
318     self.add_data(SID_AUTH_RESPONSE, DID_AUTH_KEY, "key")
319     self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_SEED)
320     self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_SEED)
321     self.add_msg_to_auth_whitelist_(SID_AUTH_REQUEST, DID_AUTH_KEY)
322     self.add_msg_to_auth_whitelist_(SID_AUTH_RESPONSE, DID_AUTH_KEY)
323     self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_SEED, self.
__authenticate_create_seed__)
324     self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_SEED, self.
__authenticate_create_key__)
325     self.__callbacks__.add(SID_AUTH_REQUEST, DID_AUTH_KEY, self.__authenticate_check_key__)
326     self.__callbacks__.add(SID_AUTH_RESPONSE, DID_AUTH_KEY, self.
__authenticate_process_feedback__)
327     self.__authentication_state_reset__()
328
329     self.add_service(SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE, "channel name
request", "channel name response")
330     self.add_data((SID_CHANNEL_NAME_REQUEST, SID_CHANNEL_NAME_RESPONSE), DID_CHANNEL_NAME, "
name")
331     self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME)
332     self.add_msg_to_auth_whitelist_(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME)
333     self.__callbacks__.add(SID_CHANNEL_NAME_REQUEST, DID_CHANNEL_NAME, self.
__channel_name_request__)
334     self.__callbacks__.add(SID_CHANNEL_NAME_RESPONSE, DID_CHANNEL_NAME, self.
channel_name_response__)
335
336     self.add_service(SID_READ_REQUEST, SID_READ_RESPONSE, "read data request", "read data
response")
337     self.add_service(SID_WRITE_REQUEST, SID_WRITE_RESPONSE, "write data request", "write data
response")
338     self.add_service(SID_EXECUTE_REQUEST, SID_EXECUTE_RESPONSE, "execute request", "execute
response")

```

```

339
340     self.__seed__ = None
341     self.__comm_inst__.register_callback(self.__data_available_callback__)
342     self.__comm_inst__.register_connect_callback(self.__connection_established__)
343     self.__comm_inst__.register_disconnect_callback(self.__authentication_state_reset__)
344     self.logger.info("%s Initialisation finished.", self.__log_prefix__())
345
346     def __analyse_frame__(self, frame):
347         if frame is not None:
348             return data_storage(json.loads(frame[:-4].decode("utf-8")))
349
350     def __authenticate_check_key__(self, msg):
351         key = msg.get_data()
352         if key == self.__authenticate_salt_and_hash__(self.__seed__):
353             self.__authentication_state__ = AUTH_STATE_TRUSTED_CONNECTION
354             return STATUS_OKAY, True
355         else:
356             self.__authentication_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
357             return STATUS_OKAY, False
358
359     def __authenticate_create_key__(self, msg):
360         self.__authentication_state__ = AUTH_STATE_KEY_TRANSFERRED
361         seed = msg.get_data()
362         key = self.__authenticate_salt_and_hash__(seed)
363         self.send(SID_AUTH_REQUEST, DID_AUTH_KEY, key)
364
365     def __authenticate_create_seed__(self, msg):
366         self.__authentication_state__ = AUTH_STATE_SEED_TRANSFERRED
367         self.__seed__ = binascii.hexlify(os.urandom(32)).decode("utf-8")
368         return STATUS_OKAY, self.__seed__
369
370     def __authenticate_process_feedback__(self, msg):
371         feedback = msg.get_data()
372         if feedback:
373             self.__authentication_state__ = AUTH_STATE_TRUSTED_CONNECTION
374             self.logger.info("%s Got positive authentication feedback", self.__log_prefix__())
375         else:
376             self.__authentication_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
377             self.logger.warning("%s Got negative authentication feedback", self.__log_prefix__())
378         return STATUS_OKAY, None
379
380     def __authenticate_salt_and_hash__(self, seed):
381         return hashlib.sha512(bytes(seed, "utf-8") + self.__secret__).hexdigest()
382
383     def __authentication_state_reset__(self):
384         self.logger.info("%s Resetting authentication state to AUTH_STATE_UNTRUSTED_CONNECTION",
385             self.__log_prefix__())
386         self.__authentication_state__ = AUTH_STATE_UNTRUSTED_CONNECTION
387
388     def __authentication_required__(self, service_id, data_id):
389         return data_id not in self.__auth_whitelist__.get(service_id, [])
390
391     def __buffer_received_data__(self, msg):
392         if not msg.get_service_id() in self.__msg_buffer__:
393             self.__msg_buffer__[msg.get_service_id()] = {}
394         if not msg.get_data_id() in self.__msg_buffer__[msg.get_service_id()]:
395             self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()] = []
396         self.__msg_buffer__[msg.get_service_id()][msg.get_data_id()].append(msg)
397         self.logger.debug("%s Message data is stored in buffer and is now ready to be retrieved
398             by receive method", self.__log_prefix__())
399

```

```

398 def __build_frame__(self, msg):
399     data_frame = json.dumps(self.__mk_msg__(msg.get_status(), msg.get_service_id(), msg.
get_data_id(), msg.get_data()))
400     data_frame = bytes(data_frame, "utf-8")
401     checksum = self.__calc_chksum__(data_frame)
402     return data_frame + checksum
403
404 def __calc_chksum__(self, raw_data):
405     return struct.pack(">I", binascii.crc32(raw_data) & 0xFFFFFFFF)
406
407 @property
408 def __channel_name__(self):
409     cn = self.logger.name.split(".")[1]
410     if cn != self.DEFAULT_CHANNEL_NAME:
411         return cn
412
413 def __channel_name_response__(self, msg):
414     data = msg.get_data()
415     if self.__channel_name__ is None and data is not None:
416         self.__init_channel_name__(data)
417         self.logger.info("%s channel name is now %s", self.__log_prefix__(), repr(self.
__channel_name__))
418     return STATUS_OKAY, None
419
420 def __channel_name_request__(self, msg):
421     data = msg.get_data()
422     if data is None:
423         return STATUS_OKAY, self.__channel_name__
424     else:
425         prev_channel_name = self.__channel_name__
426         self.__init_channel_name__(data)
427         if prev_channel_name is not None and prev_channel_name != data:
428             self.logger.warning(
429                 "%s overwriting user defined channel name from %s to %s", self.__log_prefix__
(), repr(prev_channel_name), repr(data)
430             )
431         elif prev_channel_name is None:
432             self.logger.info("%s channel name is now %s", self.__log_prefix__(), repr(self.
__channel_name__))
433         return STATUS_OKAY, None
434
435 def __check_frame_checksum__(self, frame):
436     try:
437         return self.__calc_chksum__(frame[:-4]) == frame[-4:]
438     except TypeError:
439         logger.exception("Unable to check checksum")
440         return False
441
442 def __clean_receive_buffer__(self):
443     self.logger.debug("%s Cleaning up receive-buffer", self.__log_prefix__())
444     self.__msg_buffer__ = {}
445
446 def __connection_established__(self):
447     self.__clean_receive_buffer__()
448     if self.__comm_inst__.IS_CLIENT:
449         self.send(SID_CHANNEL_NAME_REQUEST, 0, self.__channel_name__)
450     if self.__auto_auth__ and self.__comm_inst__.IS_CLIENT and self.__secret__ is not None:
451         self.authenticate()
452
453 def __log_msg__(self, msg, rx_tx_prefix):
454     self.logger.log(

```

```

455         self.__status_log_lvl__(msg.get_status()),
456         '%s %s %s, %s, data: "%s"' ,
457         self.__log_prefix__(),
458         rx_tx_prefix,
459         self.__get_message_name__(msg.get_service_id(), msg.get_data_id()),
460         self.__get_status_name__(msg.get_status()),
461         repr(msg.get_data()),
462     )
463
464     def __data_available_callback__(self, comm_inst):
465         frame = comm_inst.receive()
466         msg = self.__analyse_frame__(frame)
467         if not self.__check_frame_checksum__(frame):
468             # Wrong Checksum
469             self.logger.log(
470                 self.__status_log_lvl__(STATUS_CHECKSUM_ERROR),
471                 "%s Received message has an invalid checksum. Message will be ignored.",
472                 self.__log_prefix__(),
473             )
474             return # No response needed
475         elif not self.check_authentication_state() and self.__authentication_required__(msg.get_service_id(), msg.get_data_id()):
476             # Authentication required
477             self.__log_msg__(msg, "RX <-")
478             if msg.get_service_id() in self.__sid_response_dict__.keys():
479                 self.logger.log(
480                     self.__status_log_lvl__(STATUS_AUTH_REQUIRED),
481                     "%s Authentication is required. Just sending negative response.",
482                     self.__log_prefix__(),
483                 )
484                 status = STATUS_AUTH_REQUIRED
485                 data = None
486             else:
487                 self.logger.log(
488                     self.__status_log_lvl__(STATUS_AUTH_REQUIRED),
489                     "%s Authentication is required. Incomming message will be ignored.",
490                     self.__log_prefix__(),
491                 )
492                 return # No response needed
493         else:
494             # Valid message
495             self.__log_msg__(msg, "RX <-")
496             callback, args, kwargs = self.__callbacks__.get(msg.get_service_id(), msg.get_data_id
497             ())
498             if msg.get_service_id() in self.__sid_response_dict__.keys():
499                 #
500                 # REQUEST RECEIVED
501                 #
502                 if callback is None:
503                     self.logger.warning("%s Incomming message with no registered callback. Sending negative response.", self.__log_prefix__())
504                     status = STATUS_BUFFERING_UNHANDLED_REQUEST
505                     data = None
506                 else:
507                     self.logger.debug("%s Executing callback %s to process received data", self.__log_prefix__(), callback.__name__)
508                     try:
509                         status, data = callback(msg, *args, **kwargs)
510                     except Exception as e:
511                         self.logger.error(

```

```

511         '{lp} Exception raised. Check callback {callback_name}: "{message}"
and it\'s return values for {msg_info}'.format(
512             lp=self.__log_prefix__(),
513             callback_name=callback.__name__,
514             message=str(e),
515             msg_info=self.__get_message_name__(msg.get_service_id(), msg.
get_data_id()),
516         )
517     )
518     status = STATUS_CALLBACK_ERROR
519     data = None
520     else:
521         #
522         # RESPONSE RECEIVED
523         #
524         if callback is None:
525             self.__buffer_received_data__(msg)
526         else:
527             self.logger.debug("%s Executing callback %s to process received data", self.
__log_prefix__(), callback.__name__)
528             try:
529                 callback(msg, *args, **kwargs)
530             except Exception as e:
531                 self.logger.error(
532                     '{lp} Exception raised. Check callback {callback_name}: "{message}"
for {msg_info}'.format(
533                         lp=self.__log_prefix__(),
534                         callback_name=callback.__name__,
535                         message=str(e),
536                         msg_info=self.__get_message_name__(msg.get_service_id(), msg.
get_data_id()),
537                     )
538                 )
539             return # No response needed
540             self.send(self.__sid_response_dict__[msg.get_service_id()], msg.get_data_id(), data,
status=status)
541
542     def __get_message_name__(self, service_id, data_id):
543         return "service: %s, data_id: %s" % (
544             self.__sid_name_dict__.get(service_id, repr(service_id)),
545             self.__did_name_dict__.get(service_id, {}).get(data_id, repr(data_id)),
546         )
547
548     def __get_status_name__(self, status):
549         return "status: %s" % (self.__status_name_dict__.get(status, "unknown status: %s" % repr(
status)))
550
551     def __init_channel_name__(self, channel_name):
552         self.__comm_inst__.init_channel_name(channel_name)
553         self.__callbacks__.init_channel_name(channel_name)
554         if channel_name is None:
555             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + "." + self.
DEFAULT_CHANNEL_NAME)
556         else:
557             self.logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__ + "." +
channel_name)
558
559     def __log_prefix__(self):
560         return "prot-client:" if self.__comm_inst__.IS_CLIENT else "prot-server:"
561
562     def __mk_msg__(self, status, service_id, data_id, data):
563         return data_storage(

```

```

564         {data_storage.KEY_DATA_ID: data_id, data_storage.KEY_SERVICE_ID: service_id,
565         data_storage.KEY_STATUS: status, data_storage.KEY_DATA: data}
566     )
567
568     def __status_log_lvl__(self, status):
569         return STATUS_LOG_LVL.get(status, logging.CRITICAL)
570
571     def add_data(self, service_id, data_id, name):
572         """
573         Method to add a name for a specific message.
574
575         :param service_id: The Service-ID of the message. See class definitions starting with ``
576         SID_``.
577         :type service_id: int or list of ints
578         :param data_id: The Data-ID of the message.
579         :type data_id: int
580         :param name: The Name for the transfered message.
581         :type name: str
582         """
583         try:
584             iter(service_id)
585         except Exception:
586             service_id = (service_id,)
587
588         for sid in service_id:
589             if sid not in self.__did_name_dict__:
590                 self.__did_name_dict__[sid] = {}
591             self.__did_name_dict__[sid][data_id] = name
592
593     def add_msg_to_auth_whitelist(self, service_id, data_id):
594         """
595         Method to add a specific message to the list, where no authentication is required.
596
597         :param service_id: The Service-ID of the message. See class definitions starting with ``
598         SID_``.
599         :type service_id: int
600         :param data_id: The Data-ID of the message.
601         :type data_id: int
602         """
603         if service_id not in self.__auth_whitelist__:
604             self.__auth_whitelist__[service_id] = []
605         self.__auth_whitelist__[service_id].append(data_id)
606         self.logger.debug(
607             "%s Adding Message (%s) to the authentication whitelist", self.__log_prefix__(),
608             self.__get_message_name__(service_id, data_id)
609         )
610
611     def add_service(self, req_sid, resp_sid, req_name=None, resp_name=None):
612         """
613         Method to add a Service defined by Request- and Response Service-ID.
614
615         :param req_sid: The Request Service-ID.
616         :type req_sid: int
617         :param resp_sid: The Response Service-ID.
618         :type resp_sid: int
619         """
620         if req_sid in self.__sid_response_dict__:
621             self.logger.error(
622                 "%s Service with Request-SID=%d and Response-SID=%d not added, because request

```

```

623         raise RequestSidExistsError("Request for this Service is already registered")
624     elif resp_sid in self.__sid_response_dict__.values():
625         self.logger.error(
626             "%s Service with Request-SID=%d and Response-SID=%d not added, because response
SID is already registered",
627             self.__log_prefix__(),
628             req_sid,
629             resp_sid,
630         )
631     raise ResponseSidExistsError("Response for this Service is already registered")
632 else:
633     self.__sid_response_dict__[req_sid] = resp_sid
634     if req_name is not None:
635         self.__sid_name_dict__[req_sid] = req_name
636     if resp_name is not None:
637         self.__sid_name_dict__[resp_sid] = resp_name
638     self.logger.debug(
639         "%s Adding Service with Request=%s and Response=%s", self.__log_prefix__(),
req_name or repr(req_sid), resp_name or repr(resp_sid)
640     )
641
642 def add_status(self, status, name):
643     """
644     Method to add a name for a status.
645
646     :param status: The Status. See class definitions starting with ``STATUS_``.
647     :type status: int
648     :param name: The Name for the Status.
649     :type name: str
650     """
651     self.__status_name_dict[status] = name
652
653 def authenticate(self, timeout=2):
654     """
655     This method authenticates the client at the server.
656
657     :param timeout: The timeout for the authentication (requesting seed, sending key and
getting authentication_feedback).
658     :type timeout: float
659     :returns: True, if authentication was successfull; False, if not.
660     :rtype: bool
661
662     .. note:: An authentication will only processed, if a secret had been given on
initialisation.
663
664     .. note:: Client and Server needs to use the same secret.
665     """
666     if self.__secret__ is not None:
667         self.__authentication_state__ = AUTH_STATE_SEED_REQUESTED
668         self.send(SID_AUTH_REQUEST, DID_AUTH_SEED, None)
669         cnt = 0
670         while cnt < timeout * 10:
671             time.sleep(0.1)
672             if self.__authentication_state__ == AUTH_STATE_TRUSTED_CONNECTION:
673                 return True
674             elif self.__authentication_state__ == AUTH_STATE_UNTRUSTED_CONNECTION:
675                 break
676             cnt += 1
677         return False
678
679 def check_authentication_state(self):

```

```

680     """
681     This Method return the Authitification State as boolean value.
682
683     :return: True, if authentication state is okay, otherwise False
684     :rtype: bool
685     """
686     return self.__secret__ is None or self.__authentication_state__ ==
AUTH_STATE_TRUSTED_CONNECTION
687
688 def connection_established(self):
689     """
690     This Method returns the Connection state including authentication as a boolean value.
691
692     :return: True, if the connection is established (incl. authentication, if a secret has
been given)
693     :rtype: bool
694     """
695     return self.is_connected() and (self.__secret__ is None or self.
check_authentication_state())
696
697 def is_connected(self):
698     """
699     This Methods returns Connection state of the Communication Instance :func:`comm_instance.
is_connected`.
700
701     :return: True if the :class:`comm_instance` is connected, otherwise False..
702     :rtype: bool
703     """
704     return self.__comm_inst__.is_connected()
705
706 def receive(self, service_id, data_id, timeout=1):
707     """
708     This Method returns a message object for a defined message or None, if this message is
not available after the given timeout.
709
710     :param service_id: The Service-ID for the message. See class definitions starting with ``
SID_``.
711     :type service_id: int
712     :param data_id: The Data-ID for the message.
713     :type data_id: int
714     :param timeout: The timeout for receiving.
715     :type timeout: float
716     :returns: The received data storage object or None, if no data was received.
717     :rtype: data_storage
718     """
719     data = None
720     cnt = 0
721     while data is None and cnt < timeout * 10:
722         try:
723             data = self.__msg_buffer__.get(service_id, {}).get(data_id, []).pop(0)
724         except IndexError:
725             data = None
726             cnt += 1
727             time.sleep(0.1)
728     if data is None and cnt >= timeout * 10:
729         self.logger.warning(
730             "%s TIMEOUT (%ss): Requested data (service_id: %s; data_id: %s) not in buffer.",
731             self.__log_prefix__(),
732             repr(timeout),
733             repr(service_id),
734             repr(data_id),
735         )

```



```

736         return data
737
738     def reconnect(self):
739         """
740         This methods initiates a reconnect by calling :func:`comm_instance.reconnect`.
741         """
742         return self.__comm_inst__.reconnect()
743
744     def register_callback(self, service_id, data_id, callback, *args, **kwargs):
745         """
746         This method registers a callback for the given parameters. Giving ``None`` means, that
747         all Service-IDs or all Data-IDs are used.
748         If a message hitting these parameters has been received, the callback will be executed.
749
750         :param service_id: The Service-ID for the message. See class definitions starting with ``
751         SID_``.
752         :type service_id: int
753         :param data_id: The Data-ID for the message.
754         :type data_id: int
755
756         .. note:: The :func:`callback` is prioritised in the following order:
757
758             * Callbacks with defined Service-ID and Data-ID.
759             * Callbacks with a defined Service-ID and all Data-IDs.
760             * Callbacks with a defined Data-ID and all Service-IDs.
761             * Unspecific Callbacks.
762
763         .. note:: The :func:`callback` is executed with these arguments:
764
765             **Parameters given at the callback call:**
766
767             * The first Arguments is the received message as :class:`data_storage` object.
768             * Further arguments given at registration.
769             * Further keyword arguments given at registration.
770
771             **Return value of the callback:**
772
773             If the Callback is a Request Callback for a registered Service, the return value has
774             to be a tuple or list with
775
776             * :const:`response_status`: The response status (see class definitions starting with
777             :const:`STA_*`).
778             * :const:`response_data`: A JSON iterable object to be used as data for the response.
779
780         .. note:: Only registered services will respond via the callbacks return values with the
781         same data_id.
782         """
783         self.__callbacks__.add(service_id, data_id, callback, *args, **kwargs)
784
785     def send(self, service_id, data_id, data, status=STATUS_OKAY, timeout=2):
786         """
787         This methods sends out a message with the given content.
788
789         :param service_id: The Service-ID for the message. See class definitions starting with ``
790         SERVICE_``.
791         :type service_id: int
792         :param data_id: The Data-ID for the message.
793         :type data_id: int
794         :param data: The data to be transfered. The data needs to be json compatible.
795         :type data: str
796         :param status: The Status for the message. All requests should have ``STATUS_OKAY``.

```

```

791         :type status: int
792         :param timeout: The timeout for sending data (e.g. time to establish new connection).
793         :type timeout: float
794         :return: True if data had been sent, otherwise False.
795         :rtype: bool
796         """
797         if (self.check_authentication_state() or not self.__authentication_required__(
798             service_id, data_id)) or (
799             service_id in self.__sid_response_dict__.values() and status == STATUS_AUTH_REQUIRED
800             and data is None
801         ):
802             msg = data_storage(service_id=service_id, data_id=data_id, data=data, status=status)
803             self.__log_msg__(msg, "TX ->")
804             return self.__comm_inst__.send(self.__build_frame__(msg), timeout=timeout)
805         else:
806             # Authentication required
807             self.logger.warning(
808                 "%s Authentication is required. TX-Message %, %, data: %s will be ignored.",
809                 self.__log_prefix__(),
810                 self.__get_message_name__(service_id, data_id),
811                 self.__get_status_name__(status),
812                 repr(data),
813             )
814             return False
815
816 class struct_json_protocol(pure_json_protocol):
817     """
818     This Class has the same functionality like :class:`pure_json_protocol`. The message length is
819     less than for :class:`pure_json_protocol`, but the functionality and compatibility is
820     reduced.
821     See also parent :py:class:`pure_json_protocol`.
822
823     .. note::
824         This class is depreceated and here for compatibility reasons (to support old clients or
825         servers). Usage of :class:`pure_json_protocol` is recommended.
826     """
827
828     def __init__(self, *args, **kwargs):
829         pure_json_protocol.__init__(self, *args, **kwargs)
830
831     def __analyse_frame__(self, frame):
832         status, service_id, data_id = struct.unpack(">III", frame[0:12])
833         data = json.loads(frame[12:-1].decode("utf-8"))
834         return self.__mk_msg__(status, service_id, data_id, data)
835
836     def __build_frame__(self, msg):
837         frame = struct.pack(">III", msg.get_status(), msg.get_service_id(), msg.get_data_id())
838         frame += bytes(json.dumps(msg.get_data()), "utf-8")
839         frame += self.__calc_chksum__(frame)
840         return frame
841
842     def __calc_chksum__(self, raw_data):
843         chksum = 0
844         for b in raw_data:
845             chksum ^= b
846         return bytes([chksum])
847
848     def __check_frame_checksum__(self, frame):
849         return self.__calc_chksum__(frame[:-1]) == frame[-1:]

```