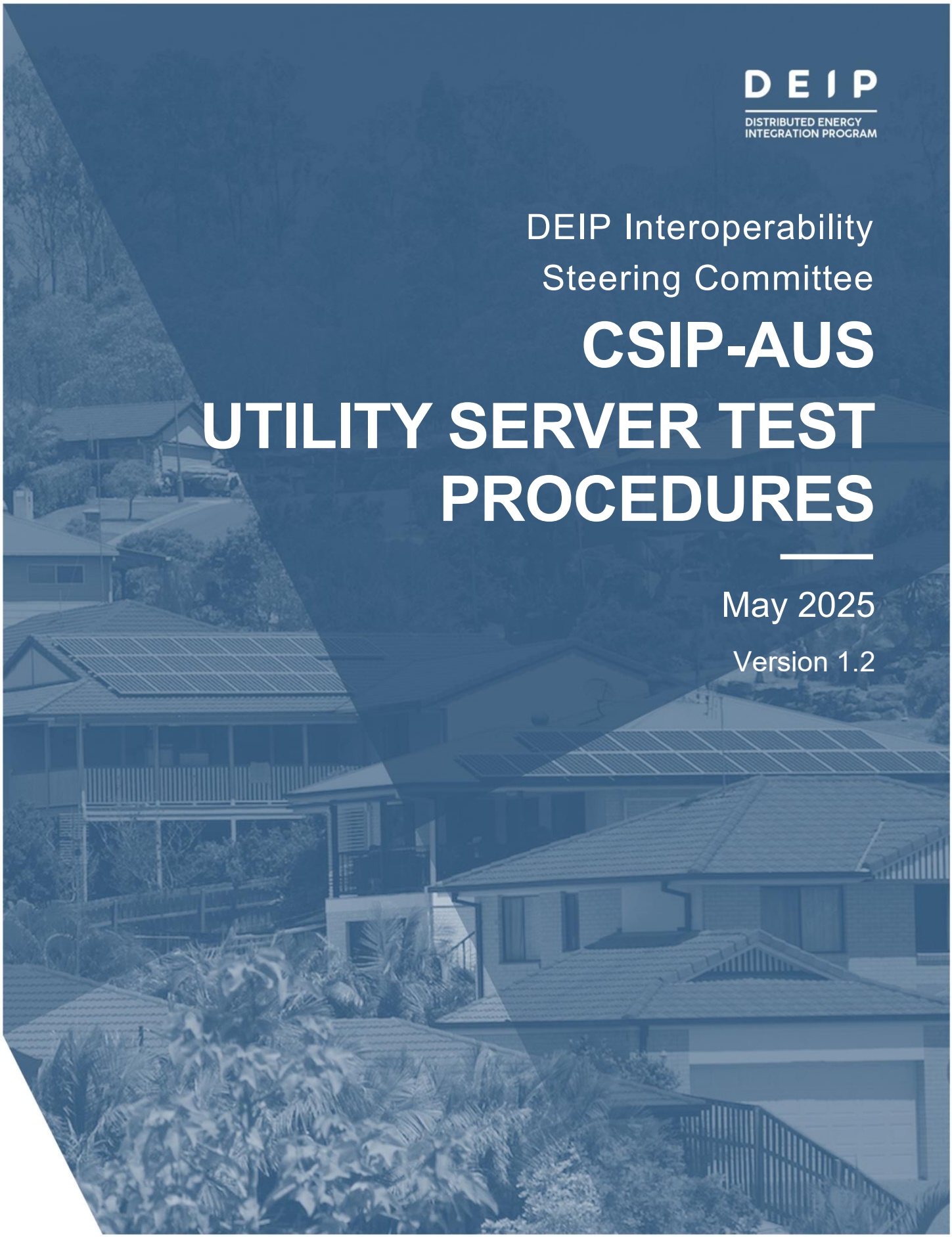

DEIP

DISTRIBUTED ENERGY INTEGRATION PROGRAM

DEIP Interoperability Steering Committee

CSIP-AUS UTILITY SERVER TEST PROCEDURES

May 2025

Version 1.2

About DEIP

The Distributed Energy Integration Program (DEIP) is a collaborative network of government agencies, market bodies, peak industry bodies and consumer associations working together to maximise the value of distributed energy resources (DER) for all Australian consumers.

The DEIP members exchange insights, seek industry consensus, and focus attention on priority activities to provide the necessary pre-policy evidence needed to support informed decision making in the Australian energy transition.

Acknowledgement and Disclaimer

This 'Common Smart Inverter Profile – Australia' and these accompanying Test Procedures were developed by the DER Integration API Technical Working Group. This working group formed in 2019 as a collaboration of Australian energy sector businesses from across the supply chain, including numerous distribution networks, retailers, equipment manufacturers and aggregators.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background

In order to improve the integration of increasing levels of distributed energy resources (DER) in Australian distribution networks, in 2021 the DER Integration API Technical Working Group (DERIAPITWG) released the first version of the Common Smart Inverter Profile – Australia (CSIP-AUS) integration guide. This guide expanded on the CSIP implementation guide based on IEEE 2030.5 / AS 5385 to provide a coherent set of protocols for managing Australia-specific requirements for DER-integration, with a particular focus on the implementation of Dynamic Operating Envelopes. Example use-cases for DOEs are: Flexible Connections including Flexible Exports and or Flexible Imports, and as an Emergency Backstop to control inverters. In the years following CSIP-AUS was supplemented with additional elements, most notably the CSIP-AUS client test procedures released in 2023, with an extension due imminently that will extend this to the verification of DER physical performance.

This document provides a framework and set of procedures to test utility servers against the specifications contained in the CSIP-AUS implementation guide. This version 1.2 is intended to enable verification of conformance against CSIP-AUS version 1.2, and future revisions of CSIP-AUS will be released alongside updates to these procedures to ensure that continuing conformance can be maintained.

1.2 Scope

This document contains test procedures that are designed to assess whether a utility server conforms to the requirements of CSIP-AUS v1.2 (2024).

In this document the term “communications client” is used to describe software designed to receive CSIP-AUS instructions for the purposes of controlling one or more DER. This is equivalent to the use of the term “DER client” in CSIP; the alternative terminology is used here to limit confusion between tests that apply to communications clients versus DER, as described in Section 2.1.

A companion document to both CSIP-AUS and these test procedures (tentatively titled *Common Smart Inverter Profile – Australia – Explanatory Guide*) is intended to be released in the near future, and will provide additional context and guidance to assist implementers and broader industry understand the requirements and functions of CSIP-AUS.

This document consists of testing procedures that a utility server may be tested against by an entity in order to validate conformance with CSIP-AUS. Section 2 describes general requirements for testing, while detailed test procedures are provided in Section 3, and are broken down into subsections testing specific functional areas.

1.3 Acronyms

CSIP-AUS	Common Smart Inverter Profile – Australia
DER	Distributed Energy Resource
DOE	Dynamic Operating Envelope
DRED	Demand Response Enabling Device
EXI	Efficient XML Interchange
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
NTP	Network Time Protocol
OI	Operating Instruction
REST	REpresentational State Transfer
V2G	Vehicle-to-Grid
XML	eXtensible Markup Language

1.4 Definitions

Averaging window – the time period over which an averaging calculation is applied to create time-average of the required measurement.

Comms – shorthand for ‘communications’.

Additional definitions are provided in both CSIP-AUS and the CSIP-AUS Explainer.

2 GENERAL REQUIREMENTS FOR TESTING

2.1 General

This section provides general guidance for the testing of utility servers against the requirements in CSIP-AUS. The testing requirements described in this section apply to all of the tests in Section 3, unless otherwise specified.

2.2 Test Setup

2.2.1 Communications Setup

The testing environment includes a software client that is able to represent both a communications client and one or more DER under control. The tester may elect to do this using either a genuine CSIP-AUS client and actual DER, a test harness that represents a simulated CSIP-AUS client & DER (which will be capable of both subscription/notification and polling functiuoanlity to support obth aggregator and native client types) or a combination of these as needed. A diagram showing the scope of this testing is shown below in Figure 1.

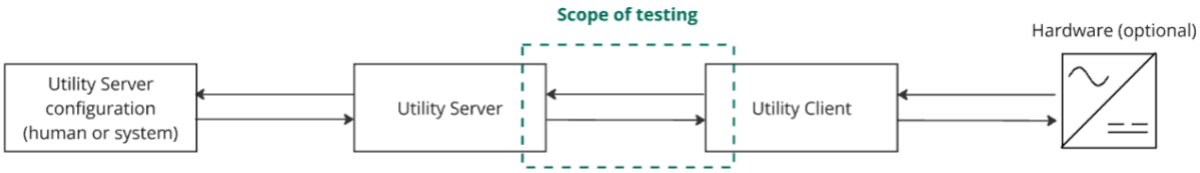

Figure 1 – Scope of utility server testing

Many of the tests in Section 3 include the provision of controls for both generation- and load-type DER. Whether communications client & DER are real or simulated they shall be capable of responding as appropriate to these controls as described in the test. This may mean utilising or simulation a bidirectional DER, or through the use of multiple DER to cover both generation and load requirements.

2.3 Test Configuration

2.3.1 Protocol Requirements

The test client, in addition to the requirements of AS 5385 (IEEE 2030.5), supports the following communications parameters in order to conduct conformances tests for all feasible utility servers:

a) Payloads communicated via HTTPS in XML;
- Resources contain links to their subordinate resources;

b) REST paging techniques are used to process lists from the client that may be incomplete; and
- Resources use the standard AS 5385 (IEEE 2030.5) namespace of urn:ieee:std:2030.5:ns
    a. CSIP-AUS extensions use namespace of https://csipaus.org/ns

It is expected that any physical and electrical commissioning between the utility server, communications client and DER is done prior to commencing tests.

All tests shall validate XML payloads sent by a server during the test against the definitions within the SEP2 and CSIP-AUS schema[1]. This is specifically to ensure that all mandatory fields are supported and payloads are correctly structured, to ensure that production clients do not reject utility server payloads,

[1] CSIP-AUS schema XSD files can be downloaded from https://csipaus.org/ns/

and to verify that ordering of sequences within the payloads matches the requirements of both AS 5385 (IEEE 2030.5) and CSIP-AUS (and particularly that CSIP-AUS extensions occur at the end of any sequences where they are present).

Note: CSIP-AUS schema XSD files can be downloaded from https://csipaus.org/ns/

Where servers are required to create internal resources in response to client interactions (e.g. EndDevice, DER) the creation of resources will take no longer than 10 seconds.

Numerous tests require that the server store data provided by the client or otherwise created during the test, and that the tester is able to verify that this data has been correctly stored. Utility servers shall provide the tester with a means to verify that such storage has occurred; possible (although non-exhaustive) ways of doing do include through the use of a user interface provided by the utility server, through query access to utility server data-stores, and through the use of log files demonstrating storage has occurred.

2.3.2 Supported Functions

2.3.2.1 Core Functions

To conform with CSIP-AUS a utility server shall meet the following sets of CSIP-AUS capabilities:

a) Time
b) Device Capability
c) End Device
d) FSA
e) DER
f) Response
g) Meter/Mirror Meter
h) Log Event
i) Subscription/Notification
j) Security

2.3.2.2 Optional Components

To conform with CSIP-AUS the utility server shall specify any optional functions it supports. A utility server may choose to comply with the following:

- The CSIP-AUS *ConnectionPoint* extension
- *RegistrationLink* and PIN validation

The following function sets are marked as 'Optional' in IEEE 2030.5 & CSIP. For the absence of confusion, all CSIP-AUS servers (and clients) are expected to support these function sets:

a) Response
b) Meter/Mirror Meter

2.3.2.3 Demand Response Components

A server that wishes to claim conformance with only the demand response components of CSIP-AUS (defined in CSIP-AUS Annex C) shall meet the following appliance management controls:

- *opModFixedW*

2.3.2.4 Subscription/Notification Components

The following components are mandatory for utility servers in order to support aggregator-type clients:

a) Subscription/Notification.

2.3.3 Default Configuration

On initial discovery the utility server shall configure the test communications client with the following poll- and post-rates and default controls for the listed Resources in Table 11.

Table 1 – Default communication configurations

Resource	Rate Type	Value
DeviceCapability	pollRate	300 sec
EndDeviceList	pollRate	300 sec
FunctionSetAssignmentsList	pollRate	300 sec
DERProgramList	pollRate	60 sec
DERList (including DERStatus, DERSettings and DERCapability)	pollRate	60 sec
MirrorUsagePoint	postRate	60 sec
DefaultDERControl:opModExpLimW	N/A	0 W
DefaultDERControl:opModImpLimW	N/A	0 W
setGradW	N/A	0.27%/sec

Notes:

- *These values differ from the default values defined in CSIP-AUS and are changed for testing in order to expedite the tests.*
- *The DERList pollRate is also used to determine frequency of posting for subordinate resources.*
- *setGradW is numerically represented as an integer specifying hundredths of a percent per second, so the default value of 0.27%/sec would be represented in payloads as 27.*

Some tests in this document make changes to the state of the utility server, DER or communications client, by changing default values (e.g. *DefaultDERControls*, post/poll-rates and ramp-rates) or by applying active *DERControls*. Such changes are only intended to apply for the test in which they are defined, and shall be reverted back to the values in Table 1 after the conclusion of the test, which may include reverting the client to default values and cancelling or deleting any active *DERControls* that still exist at the end of a given test.

Unless otherwise specified, all DERControls in these tests will be for 5 minutes to begin immediately.

2.3.4 Communication Interactions

Figure 2 and Figure 3 below show the primary communications interactions to provide the functionality required to conform with tests described in this document. Figure 2 relates to the discovery process (tested in Section 3.2.1 and following sections) while Figure 3 describes the commands involved in ongoing communication. Additional examples detailing XML payloads can be found in CSIP Sections 6 and 7, and in CSIP-AUS Annexes B and E.

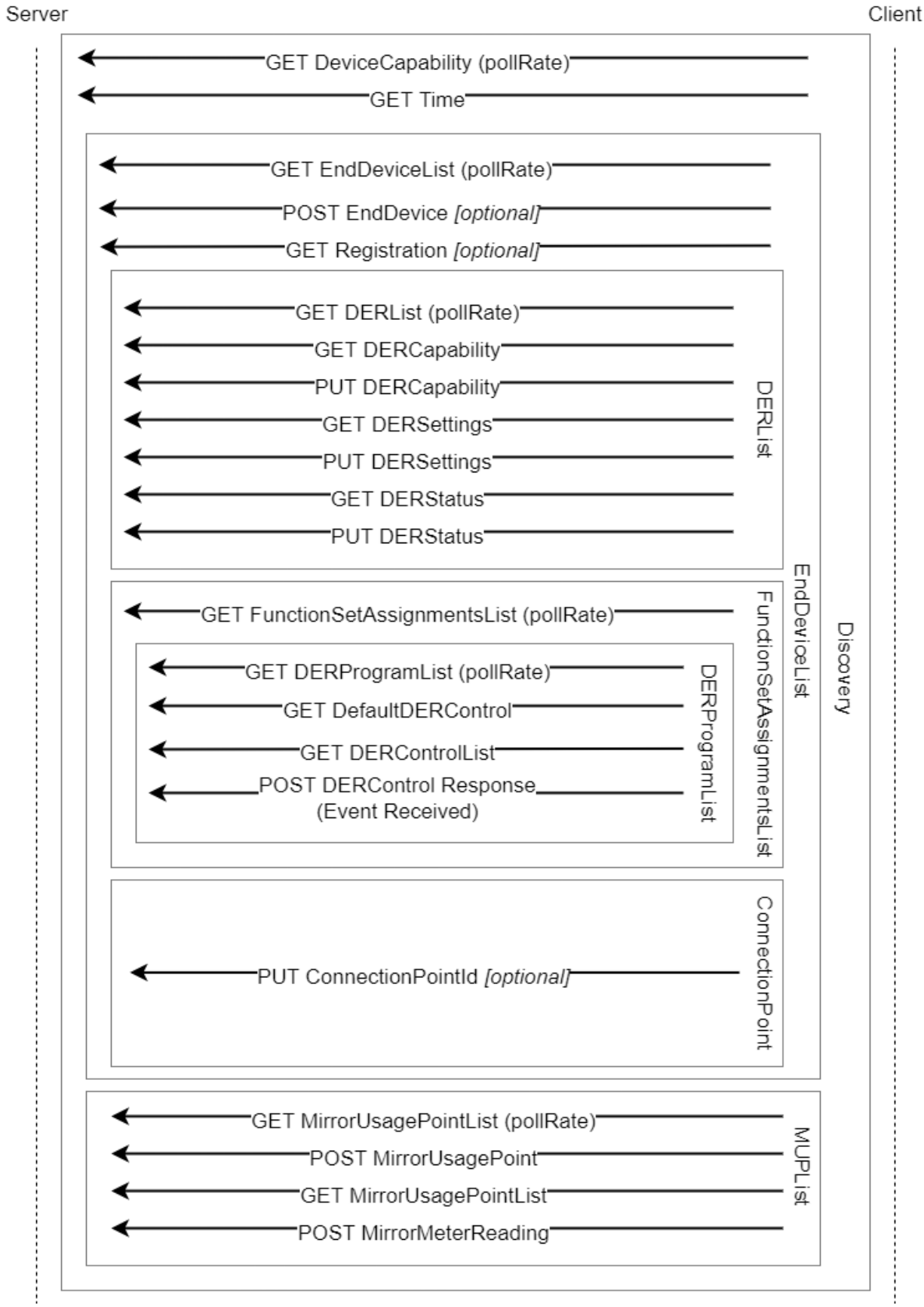

Figure 2 – Client-server interactions – Discovery

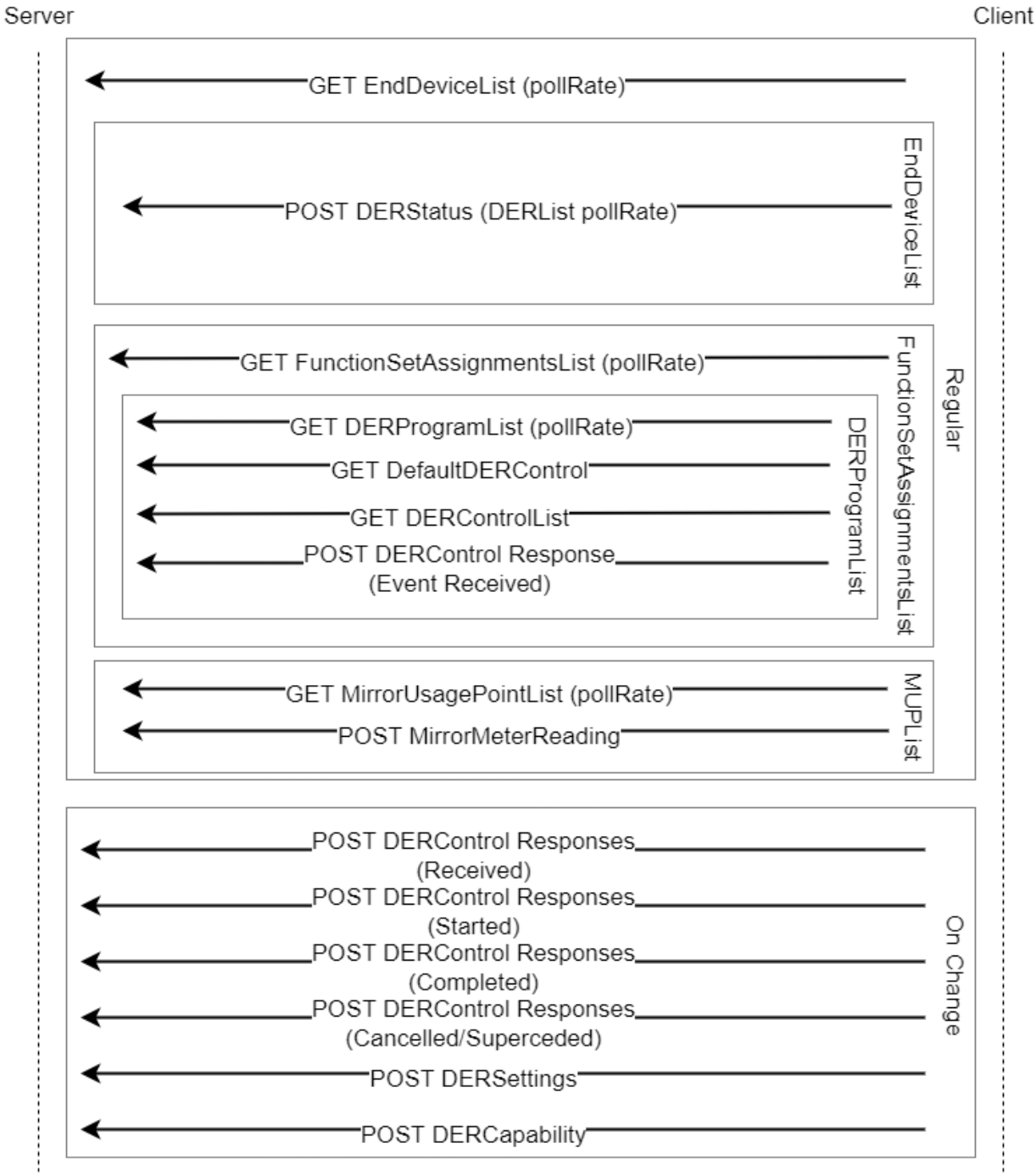

Figure 3 – Client-server interactions – Ongoing

2.4 Test Applicability

Table 2 indicates the applicability of each test to utility servers. Ticks (✓) indicate that a test is applicable to all utility servers, other text indicates that the test may be applicable depending on which functionality the utility server wishes to support. When submitting for testing utility servers shall specify the test designations for capabilities they intend to be tested against.

Table 2 – Matrix of test applicability

Test	Test Name	Test Applicable?
S-ALL-01	Discovery with Out-Of-Band Registration	✓
S-ALL-02	Discovery with In-Band Registration for Direct Clients	✓
S-ALL-03	Discovery with In-Band Registration for Aggregator Clients	✓
S-ALL-04	Discovery Rejecting In-Band Registration	✓
S-OPT-01	Incorrect Connection Point Registration	Servers not supporting in-band registration only
S-OPT-02	Connection Point Registration	Servers supporting ConnectionPoint only
S-OPT-03	Client Registration and PIN Validation	Servers supporting ConnectionPoint only
S-OPT-04	Support for Multiple End Clients	Servers supporting Registration and PIN only
S-ALL-05	Authorisation	✓
S-ALL-06	Mirror Usage Point Creation and Individual Readings	✓
S-ALL-07	Select Edge Cases for Telemetry Reporting	✓
S-ALL-08	Connection Status	✓
S-ALL-09	Operational Mode Status	✓
S-ALL-10	Capabilities and Settings	✓
S-OPT-05	Update Telemetry Post Rates	✓
S-ALL-11	Polled - Active Control – Export Limit	✓
S-ALL-12	Polled - Active Control – Generation Limit	✓
S-ALL-13	Polled - Active Control – Import Limit	✓
S-ALL-14	Polled - Active Control – Load Limit	✓
S-ALL-15	Polled - Active Control – Energise / De-energise	✓
S-ALL-16	Polled - Active Control – Disconnect	✓
S-ALL-17	Polled - Active Control – Control Combinations	✓
S-ALL-18	Polled - Active Control – Ramp Rates	✓
S-ALL-19	Polled - Default Control – Export Limit	✓
S-ALL-20	Polled - Default Control – Generation Limit	✓
S-ALL-21	Polled - Default Control – Import Limit	✓
S-ALL-22	Polled - Default Control – Load Limit	✓
S-ALL-23	Polled - Default Control – Ramp Rate	✓
S-ALL-24	Polled - Default Control – Control Combinations	✓
S-ALL-25	Subscribe	✓
S-ALL-26	Update and Delete Subscription	✓
S-ALL-27	Subscription - Active Control – Export Limit	✓
S-ALL-28	Subscription - Active Control – Generation Limit	✓
S-ALL-29	Subscription - Active Control – Import Limit	✓
S-ALL-30	Subscription - Active Control – Load Limit	✓
S-ALL-31	Subscription - Active Control – Energise / De-energise	✓
S-ALL-32	Subscription - Active Control – Disconnect	✓
S-ALL-33	Subscription - Active Control – Control Combinations	✓
S-ALL-34	Subscription - Active Control – Ramp Rates	✓

S-ALL-35	Subscription - Default Control – Export Limit	✓
S-ALL-36	Subscription - Default Control – Generation Limit	✓
S-ALL-37	Subscription - Default Control – Import Limit	✓
S-ALL-38	Subscription - Default Control – Load Limit	✓
S-ALL-39	Subscription - Default Control – Ramp Rate	✓
S-ALL-40	Variable Export Limit	✓
S-ALL-41	Control Responses	✓
S-ALL-42	Delayed Response to Cancelled Export Control	✓
S-ALL-43	Update Poll Rates	✓
S-ALL-44	Scheduling	✓
S-ALL-45	Randomisation	✓
S-ALL-46	Communication Loss	✓
S-ALL-47	Validation of Scaling Factors	✓
S-ALL-48	Changing Ramp-Rates	✓
S-ALL-49	Multiple DER Programs	✓
S-ALL-50	Extended Operations	✓
S-ALL-51	End Device Duplication	✓
S-ALL-52	LFDI Doesn't Match Certificate	✓
S-ALL-53	Update Connection Point	✓
S-ALL-54	Update Aggregator Certificate	✓
S-ALL-55	Reject Incomplete DER Settings	✓
S-ALL-56	Reject String LFDIs	✓
S-ALL-56	Invalid Control Responses	✓
S-OPT-06	DRED Operational Instruction Response	Servers supporting AS/NZS 4755 demand response only

3 TEST PROCEDURES

3.1 General

This section details the test procedures that a utility server shall complete in order to verify conformance with CSIP-AUS. As detailed in Section 2.4 above, some tests that verify a server's conformance with optional capabilities, and are only applicable to servers that wish to claim conformance with the relevant capability. Such tests have a designation beginning with "S-OPT-".

3.2 Registration

3.2.1 S-ALL-01 Discovery with Out-Of-Band Registration

3.2.1.1 Purpose

This test is intended to validate the utility server's ability to support discovery by clients and to establish basic AS 5385 (IEEE 2030.5) based communications using out-of-band registration of the *EndDevice*.

3.2.1.2 Precondition

If either of the S-ALL-02 or S-ALL-03 tests have been run prior to this test, the server will need to be updated to remove the existing *EndDevice* entry for the client, and the client may need to be reconfigured (e.g. factory-reset) prior to this test to ensure it re-runs the discovery process. The communications client has been preconfigured with the initial /sep2 endpoint of the test utility server. The utility server has been configured with the client's *EndDevice* information out of band.

3.2.1.3 Test procedure

The steps SHALL be performed as described in Table 3. The test results SHALL be compared against the Expected Result column in Table 3, except that steps 4 and 5 can occur in any order. For example, time-synchronisation may be done prior to the client requesting the *EndDeviceList* from the utility server.

Table 3 – Test steps – Discovery wih Out-of-Band Registration

Step No.	Manual or client step	Expected utility server result
1	Client is powered on, and performs HTTP discovery against the pre-configured utility server URI information, commencing with the *DeviceCapability* endpoint, and establish communications with the utility server as appropriate for the client.	Utility server sends the *DeviceCapability* resource including the *TimeLink*, *EndDeviceListLink* and *MirrorUsagePointListLink* resource links to the client.
2	Client receives *DeviceCapability* payload. Client requests resource information from the utility server's *EndDeviceList* endpoint.	Utility server sends the *EndDeviceList* resource containing the client's *EndDevice* information including the *ConnectionPointLink*, *RegistrationLink* and *DERListLink* resource link to the client.
3	Client receives the *EndDeviceList* payload with the required *EndDevice* information including the relevant resources.	Utility server sends the *Time* resource containing the relevant Time information to the client.

	Client accesses the utility server's *Time* endpoint provided in step 1.	
4	Client receives the *Time* payload. Client access the utility server's *DERListLink* endpoint.	Utility server sends the *DERList* resource containing all relevant resources including the *DERCapabilityLink*, *DERStatusLink* and *DERSettingsLink* resource links to the client.

3.2.1.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to respond to a correct HTTP request from the client.
  a) The utility server does not provide the listed resource links to the client.

Notes:

- Clients may respond to the utility server information in steps 3, 4 and 5 of this test in a variety of orders. They may also introduce addition communications in between these, such as the Registration functionality tested in S-ALL-03. Utility servers are expected to respond according to the requirement of IEEE 2030.5 / AS 5385 irrespective of the order these steps occur in.
- After an *EndDevice* has been configured (either in-band or out-of-band) the utility server is responsible for populating the DERList with a child DER to contain the DERCapabilityLink, DERSettlingsLink and DERStatusLink for the *EndDevice*.

3.2.2 S-ALL-02 Discovery with In-Band Registration for Direct Clients

3.2.2.1 Purpose

This test is intended to validate the utility server's ability to support discovery by direct clients and to establish basic AS 5385 (IEEE 2030.5) based communications using in-band registration of the *EndDevice*.

3.2.2.2 Precondition

If either of the S-ALL-01 or S-ALL-03 tests have been run prior to this test, the server will need to be updated to remove the existing *EndDevice* entry for the client, and the client may need to be reconfigured (e.g. factory-reset) prior to this test to ensure it re-runs the discovery process. The communications client has been preconfigured with the initial /sep2 endpoint of the test utility server.

3.2.2.3 Test procedure

The steps SHALL be performed as described in

Table 4. The test results SHALL be compared against the Expected Result column in

Table 4, except that steps 5 and 6 can occur in any order. For example, time-synchronisation may be done prior to the client requesting the *EndDeviceList* from the utility server.

Table 4 – Test steps – Discovery with In-Band Registration for Direct Clients

Step No.	Manual or client step	Expected utility server result
1	Client is powered on, and performs HTTP discovery against the pre-configured utility server URI information, commencing with the *DeviceCapability* endpoint.	Utility server sends the *DeviceCapability* resource including the *TimeLink*, *EndDeviceListLink* and *MirrorUsagePointListLink* resource links to the client.
2	Client receives *DeviceCapability* payload. Client requests resource information from the utility server's *EndDeviceList* endpoint.	Utility server sends the *EndDeviceList* resource, which does not contain an entry for the client's *EndDevice*.
3	Client receives the *EndDeviceList* payload, and posts to the *EndDeviceListLink* including the relevant information for the client's *EndDevice*.	The server responds with the location of the newly-created *EndDevice* resource.
4	The client accesses the utility sever's *EndDeviceList* endpoint.	Utility server sends the updated *EndDeviceList* resource which now contains the relevant *EndDevice* information including the *ConnectionPointLink*, *RegistrationLink* and *DERListLink* resource link to the client.
5	Client receives the updated *EndDeviceList* payload with the required *EndDevice* information including the relevant resources. Client accesses the utility server's *Time* endpoint provided in step 1.	Utility server sends the *Time* resource containing the relevant Time information to the client.
6	Client receives the *Time* payload. Client access the utility server's *DERList* endpoint.	Utility server sends the *DERList* resource containing the *DERCapabilityLink*, *DERStatusLink* and *DERSettingsLink* resource links to the client.

3.2.2.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to respond to a correct HTTP request from the client.
  a) The utility server does not provide the listed resource links to the client.
- The utility server does not populate the *EndDeviceList* with the new *EndDevice*.

Notes:

- Clients may respond to the utility server information in steps 3, 4 and 5 of this test in a variety of orders. They may also introduce addition communications in between these, such as the Connection Point Registration functionality tested in S-OPT-03. Utility servers are expected to respond according to the requirement of IEEE 2030.5 / AS 5385 irrespective of the order these steps occur in.
- To support this test, after an *EndDevice* has been configured (either in-band or out-of-band) the utility server is responsible for populating the DERList with a child DER to contain the DERCapabilityLink, DERSettlingsLink and DERStatusLink for the *EndDevice*. If the server does not perform this internally it may be done by other means, for example via running back-end scripts that perform this function.

3.2.3 S-ALL-03 Discovery with In-Band Registration for Aggregator Clients

3.2.3.1 Purpose

This test is intended to validate the utility server's ability to support discovery by aggregator clients and to establish basic AS 5385 (IEEE 2030.5) based communications using in-band registration of the *EndDevice*.

3.2.3.2 Precondition

If either of the S-ALL-01 or S-ALL-02 tests have been run prior to this test, the server will need to be updated to remove the existing *EndDevice* entry for the client, and the client may need to be reconfigured (e.g. factory-reset) prior to this test to ensure it re-runs the discovery process. The communications client has been preconfigured with the initial /sep2 endpoint of the test utility server.

3.2.3.3 Test procedure

This test is performed identically to test S-ALL-02, the only difference being that is the client SHALL be of an aggregator type. The steps SHALL be performed as described in

Table 4, with variations such that relevant resources may be provided by the server via notification rather than being polled, if subscription has been configured prior to or during the test. The test results SHALL be compared against the Expected Result column in

Table 4, except that steps 5 and 6 can occur in any order. For example, time-synchronisation may be done prior to the client requesting the *EndDeviceList* from the utility server.

3.2.3.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to respond to a correct HTTP request from the client.
  a) The utility server does not provide the listed resource links to the client.
- The utility server does not populate the *EndDeviceList* with the new *EndDevice*.

Notes:

- Clients may respond to the utility server information in steps 3, 4 and 5 of this test in a variety of orders. They may also introduce addition communications in between these, such as the Connection Point Registration functionality tested in S-OPT-03. Utility servers are expected to respond according to the requirement of IEEE 2030.5 / AS 5385 irrespective of the order these steps occur in.

- After an *EndDevice* has been configured (either in-band or out-of-band) the utility server is responsible for populating the DERList with a child DER to contain the DERCapabilityLink, DERSettlingsLink and DERStatusLink for the *EndDevice*.

3.2.4 S-OPT-01 Discovery Rejecting In-Band Registration

3.2.4.1 Purpose

This test is intended to validate the ability of a utility server to reject the in-band registration of clients that are not authorised to do so.

3.2.4.2 Precondition

If the S-ALL-01 test has been run prior to this test, the server will need to be updated to remove the existing *EndDevice* entry for the client, and the client may need to be reconfigured (e.g. factory-reset) prior to this test to ensure it re-runs the discovery process. The communications client has been preconfigured with the initial /sep2 endpoint of the test utility server. The utility server has been configured with the client's *EndDevice* information out of band.

3.2.4.3 Test procedure

The steps SHALL be performed as described in Table 5. The test results SHALL be compared against the Expected Result column in Table 5, except that steps 5 and 6 can occur in any order. For example, time-synchronisation may be done prior to the client requesting the *EndDeviceList* from the utility server.

Table 5 – Test steps – Discovery Rejecting In-Band Registration

Step No.	Manual or client step	Expected utility server result
1	Client is powered on, and performs HTTP discovery against the pre-configured utility server URI information, commencing with the *DeviceCapability* endpoint.	Utility server sends the *DeviceCapability* resource including the *TimeLink*, *EndDeviceListLink* and *MirrorUsagePointListLink* resource links to the client.
2	Client receives *DeviceCapability* payload. Client requests resource information from the utility server's *EndDeviceList* endpoint.	Utility server sends the *EndDeviceList* resource, which contains an entry for the client's *EndDevice*.
3	Client receives the *EndDeviceList* payload, and posts to the *EndDeviceListLink* including the relevant information for the client's *EndDevice*.	Utility server responds to the attempt to create a new *EndDevice* with with a 4XX HTTP response code.

3.2.4.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not reject the attempt to create a new *EndDevice*.

3.2.5 S-OPT-02 Incorrect Connection Point Registration

3.2.5.1 Purpose

This test is intended to validate a utility server's ability to detect and reject registration of incorrect *ConnectionPoints*. This test only applies utility servers that intend to support the *ConnectionPoint* extension and that support white-listing or similar mechanisms for validating *ConnectionPoint* registrations.

3.2.5.2 Precondition

Prior to test commencement the utility server will be configured such that the *connectionPointId* provided by the client in the test represents an invalid *ConnectionPoint*.

3.2.5.3 Test procedure

The steps SHALL be performed as described in Table 6. The test results SHALL be compared against the Expected Result column in Table 6.

Table 6 – Test steps – Incorrect Connection Point Registration

Step No.	Manual or client step	Expected utility server result
1	After having completed the discovery process (as detailed in tests S-ALL-01 and/or S-ALL-02) including receiving the *ConnectionPointLink* resource links from the utility server, the client sends a *connectionPointId* that the server has been configured to recognise as invalid..	The utility server returns an *Error* payload to the client with a *reasonCode* of 1 indicating an invalid request value.

3.2.5.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does provide the error response with *reasonCode* = 1 to the client.

3.2.6 S-OPT-03 Connection Point Registration

3.2.6.1 Purpose

This test is intended to validate a utility server's support for the *ConnectionPoint* extension.

3.2.6.2 Test procedure

This test only applies utility servers that intend to support the *ConnectionPoint* extension.

The steps SHALL be performed as described in Table 7. The test results SHALL be compared against the Expected Result column in Table 7.

Table 7 – Test steps – Connection Point Registration

Step No.	Manual or client step	Expected utility server result
1	After having completed the discovery process (as detailed in tests S-ALL-01 and/or S-ALL-02) including receiving the *ConnectionPointLink* resource links from the utility server, the client sends a valid *connectionPointId* to the utility server's *ConnectionPoint* endpoint.	The utility server returns a HTTP 201 response confirming that the *ConnectionPoint* has been correctly received.

3.2.6.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does provide the appropriate 201 response to the client.

3.2.7 S-OPT-04 Client Registration and PIN Validation

3.2.7.1 Purpose

This test is intended to validate the utility server's ability to provide the *Registration* payload to the client for PIN validation.

3.2.7.2 Precondition

The client will have completed the Discovery process via one (or both) of tests S-ALL-01 and S-ALL-02. Both the utility server and the client will have been pre-configured with a matching 6-digit PIN as defined in AS 5385 (IEEE 2030.5) clause 6.3.5.6.

3.2.7.3 Test procedure

This test only applies utility servers that intend to support the Registration package with PIN validation as defined in AS 5385 (IEEE 2030.5). The steps SHALL be performed as described in Table 8. The test results SHALL be compared against the Expected Result column in Table 8.

Table 8 – Test steps – Site Registration

Step No.	Manual or client step	Expected utility server result
1	The client accesses the *RegistrationLink* endpoint.	The utility server provides the *Registration* payload to the client, including the PIN code for the client.

3.2.7.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to provide the correct *Registration* payload to the client..

3.2.8 S-ALL-04 Support for Multiple End Clients

3.2.8.1 Purpose

This test is intended to validate that the utility server is able to control and receive telemetry from more than one client.

3.2.8.2 Precondition

Two clients CA and CB are created, and both perform Discovery and Registration with the utility server by completing test S-ALL-02, with both clients having a single EndDevice under management. Both clients are either simulating or managing DERs that are exporting at the connection point at least 50% of their rated active power. The utility server is configured to generate at least 5 sequential, non-overlapping scheduled controls for the EndDevices under management by the clients with a range of export and/or import limits, each with 1min duration.

3.2.8.3 Test Procedure

The steps SHALL be performed as described in Table 9. The test results SHALL be compared against the Expected Result column in Table 9.

Table 9 – Test steps – Support for Multiple End Clients

Step No.	Manual or client step	Expected utility server result
1	Client CA navigates through *EndDevice*, *FunctionSetAssignments* and *DERPrograms* to retrieve the *DERControlList* for the *EndDevice* under management.	The utility server responds with an *DERControlList* payload with the correct schedule of active controls.
2	Client CA requests resource information from the utility server's *MirrorUsagePointList* endpoint.	Utility server sends the *MirrorUsagePointList* resource that should not yet contain an entry for the client's *MirrorUsagePoint*.
3	Client CA posts a valid *MirrorUsagePoint* payload to the *MirrorUsagePointList* endpoint on the utility server including MirrorMeterReading definitions that relate to DER active power and DER voltage.	For each *MirrorUsagePoint* payload posted the server returns a HTTP payload confirming that the *MirrorUsagePoints* have been created and the URI of the resource.
4	Client CA posts *MirrorMeterReadings* or *MirrorMeterReadingList* payloads to the *MirrorUsagePoint* endpoint created in step 3 at the configured interval (the testing default is 60 seconds) for a minimum of	The utility server receives the provided data and retains it internally.

	four intervals, for each of DER active power and DER voltage.	
5	Client CB navigates through *EndDevice*, *FunctionSetAssignments* and *DERPrograms* to retrieve the *DERControlList* for the *EndDevice* under management.	The utility server responds with an *EndDeviceList* payload with the correct schedule of active controls.
6	Client CB requests resource information from the utility server's *MirrorUsagePointList* endpoint.	Utility server sends the *MirrorUsagePointList* resource that should not yet contain an entry for the client's *MirrorUsagePoint*.
7	Client CB posts a valid *MirrorUsagePoint* payload to the *MirrorUsagePointList* endpoint on the utility server for DER active power and DER voltage.	For each MirrorUsagePoint payload posted the server returns a HTTP payload confirming that the *MirrorUsagePoints* have been created and the URI of its endpoint.
8	Client CB posts *MirrorMeterReading* or *MirrorMeterReadingList* payloads to the *MirrorUsagePointList* endpoint at the configured interval (the testing default is 60 seconds) for a minimum of four sets of MirrorMeterReadings, for both DER active power and DER voltage.	The utility server receives the provided data and retains it internally.

3.2.8.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to fully support both clients' ability to access active controls and post telemetry.

3.2.9 S-ALL-05 Authorisation

3.2.9.1 Purpose

This test is intended to validate that the utility server restricts clients' access to resources that the client is not authorised to access.

3.2.9.2 Precondition

The client has completed at least one of the Discovery tests described in S-ALL-01, S-ALL-02 and S-ALL-03 including registration of its *EndDevice*. The utility server shall have at least one other *EndDevice* configured within it. Both *EndDevices* have created *MirrorUsagePoints* to enable the provision of telemetry to the server (e.g. by completing the MirrorUsagePoint creation steps in test S-ALL-05.

3.2.9.3 Test Procedure

The steps SHALL be performed as described in Table 10. The test results SHALL be compared against the Expected Result column in Table 10.

Table 10 – Test steps – Authorisation

Step No.	Manual or client step	Expected utility server result
1	The client attempts to access the *EndDeviceListLink* endpoint for an *EndDevice* that is not the one managed by the client.	The utility server receives and rejects the request by returning a HTTP response code of 401 (unauthorized) or 404 (resource does not exist), or by returning an *EndDeviceList* without the *EndDevice* that the client was not authorised to access.
2	The client attempts to access the *DERControlListLink* endpoint for an *EndDevice* that is not the one managed by the client.	The utility server receives and rejects the request by returning a HTTP response code of 401 (unauthorized) or 404 (resource does not exist), or by returning an *DERControlList* without the *DERControl* that the client was not authorised to access.
3	The client attempts to post to the *DERStatusLink* endpoint for an *EndDevice* that is not the one managed by the client.	The utility server receives and rejects the request by returning a HTTP response code of 401 (unauthorized) or 404 (resource does not exist), or by returning an *EndDeviceList* without the *EndDevice* that the client was not authorised to access.
4	The client attempts to access the *EndDeviceListLink* endpoint for the *EndDevice* that is managed by the client.	Utility server sends the *EndDeviceList* resource which contains the relevant *EndDevice* information to the client.
5	The utility server is configured to disallow access by the client to the *EndDevice* resource that it was previously managing. The client attempts to access the *EndDeviceListLink* endpoint for the *EndDevice* that was previously managed by the client.	The utility server receives and rejects the request by returning a HTTP response code of 401 (unauthorized) or 404 (resource does not exist), or by returning an *EndDeviceList* without the *EndDevice* that the client was not authorised to access.
6	The client attempts to post to the *MirrorUsagePointLink* endpoint with a MirrorMeterReading payload that specifies the *mRID* that was created by the *EndDevice* that is not the one managed by the client.	The utility server receives and rejects the request by returning a HTTP response code of 401 (unauthorized) or 404 (resource does not exist).

3.2.9.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to reject the access attempts.
- The utlity server returns resources that the client should not have authorisation to access.

Notes:

- CSIP provides additional tests to verify authorisation capabilities for utility servers that can build on the minimalist requirements checked in this test.

3.3 Monitoring

This section is intended to validate a utility server's ability to receive telemetry and other operating data from clients relating to the DER under the clients' management, including by creating all resources necessary to support receiving this data.

3.3.1 S-ALL-06 Mirror Usage Point Creation and Individual Readings

3.3.1.1 Purpose

This test is intended to validate the utility server's ability to correctly create and configure *MirrorUsagePoints* necessary for the following telemetry data:

a) Site active power
b) Site reactive power
c) DER active power
d) DER reactive power
e) Frequency
f) Site voltage
g) DER voltage

3.3.1.2 Precondition

The client has completed the Discovery process as defined in tests S-ALL-01 or S-ALL-02, and the client is either simulating or managing a DER that is generating at least 50% of its rated active power. If a real DER is used a test load is electrically connected in parallel with the DER, and set to consume 25% of the DER's rated active power.

3.3.1.3 Test procedure

The steps SHALL be performed as described in Table 11. The test results SHALL be compared against the Expected Result column in Table 11.

Table 11 – Test steps – Mirror Usage Point Creation and Individual Readings

Step No.	Manual or client step	Expected utility server result
1	Client requests resource information from the utility server's *MirrorUsagePointList* endpoint.	Utility server sends the *MirrorUsagePointList* resource that should not yet contain an entry for the client's *MirrorUsagePoint*.
2	The client posts a valid *MirrorUsagePoint* payload to the *MirrorUsagePointList* endpoint	For each MirrorUsagePoint payload posted the server returns a HTTP payload

	on the utility server for each of the telemetry datatypes listed in 3.3.1.1.	confirming that the *MirrorUsagePoints* have been created and the URI of its endpoint.
3	The client again requests resource information from the utility server's *MirrorUsagePointList* endpoint.	Utility server sends the *MirrorUsagePointList* resource that now contains an entry for each of the client's *MirrorUsagePoints created in step 2,* including relevant information such as the postRate required for *MirrorMeterReading* posts.
4	The client posts *MirrorMeterReading* or *MirrorMeterReadingList* payloads to the *MirrorUsagePoint* endpoint at the configured interval (the testing default is 60 seconds) for a minimum of four sets of MirrorMeterReadings, including all of the required and claimed data points as described above in clause 4.3.1.1.	The utility server receives the provided data and retains it internally, and returns the location endpoint of the *MirrorMeterReading*.
5	The client accesses the location of the resource returned in the response header in step 4 with request parameters that are configured to retrieve the most recent *MirrorMeterReading* telemetry that was posted in step 4.	The utility server responds with a payload which includes the *MirrorMeterReadings* posted in step 4.

3.3.1.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not configure and provide resource information and endpoints relating to all of the *MirrorUsagePoints* provided by the client.
- The utility server fails to receive, retain and then return telemetry data reported by the client.

3.3.2 S-ALL-07 Select Edge Cases for Telemetry Reporting

3.3.2.1 Purpose

This test is intended to validate the utility server's ability to correctly respond to edge-cases that can occur during client communications.

3.3.2.2 Precondition

The client has completed the MirrorUsagePoint creation process as defined in test S-ALL-06, and the client is either simulating or managing a DER that is generating at least 50% of its rated active power. If a real DER is used a test load is electrically connected in parallel with the DER, and set to consume 25% of

the DER's rated active power. The utility server has been configured with the Private Enterprise Number (PEN) of the client.

3.3.2.3 Test procedure

The steps SHALL be performed as described in Table 12. The test results SHALL be compared against the Expected Result column in Table 12.

Table 12 – Test steps – Edge Cases for Telemetry Reporting

Step No.	Manual or client step	Expected utility server result
1	The client posts a *MirrorUsagePoint* with the same mRID for one previously created to the utility server's *MirrorUsagePointList* endpoint. The newly-created *MirrorUsagePoint* shall have a different value for the *roleFlags* attribute to that previously created.	The utility server receives the updated *MirrorUsagePoint* and updates the original *MirrorUsagePoint* with the new *roleFlags* setting, updating the *serviceCategoryKind* and *status* fields accordingly.
2	The client posts a *MirrorMeterReading* with the same mRID for one previously created to the utility server's *MirrorUsagePoint* endpoint. The newly-created *MirrorMeterReading*'s *Reading* component shall have a different value for the *value* attribute to that previously created.	The utility server receives the updated *MirrorMeterReading* and updates the original *MirrorMeterReading* with the new *value* setting, updating the *lastUpdateTime*, *nextUpdateTime*, *Reading* values and *MirrorReadingSet* accordingly. Other attributes for the *MirrorMeterReading* shall remain unchanged.
3	The client posts a valid *MirrorMeterReading* with the same mRID for one previously created to the utility server's *MirrorUsagePoint* endpoint.	The utility server receives the request but does not create a new *MirrorMeterReading*; rather it will overwrite the old *MirrorMeterReading*.

3.3.2.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server creates a new *MirrorUsagePoint* rather than updating the existing *MirrorUsagePoint*.
- The utility server creates a new *MirrorMeterReading* rather than updating the existing *MirrorMeterReading*.
  a) The utility server does not reject an inval

3.3.3 S-ALL-08 Connection Status

3.3.3.1 Purpose

This test is intended to validate the utility sever's ability to receive DER generation connection status from a client.

3.3.3.2 Precondition

The client is either simulating or managing a DER that is generating at least 50% of its rated active power.

3.3.3.3 Test Procedure

The steps SHALL be performed as described in Table 13. The test results SHALL be compared against the Expected Result column in Table 13.

Table 13 – Test steps – Connection Status

Step No.	Manual or client step	Expected utility server result
1	Action is taken to change the DERStatus to 'disconnected'. Where a client manages multiple DER all DER are disconnected / reconnected as part of this test. The client PUTs to the utility server with bit 0 of *genConnectStatus* = 0.	The utility server receives and retains the updated *DERStatus* payload.
2	Action is taken to reconnect the DER to the grid supply. The client PUTs to the utility server with bit 0 of *genConnectStatus* = 1.	The utility server receives and retains the updated *DERStatus* payload.
3	Action is taken to change the DERStatus to 'unavailable'. Where a client manages multiple DER all DER are disconnected / reconnected as part of this test. The client PUTs to the utility server with bit 1 of *genConnectStatus* = 0.	The utility server receives and retains the updated *DERStatus* payload.
4	Action is taken to change the DER status to 'available'. The client PUTs to the utility server with bit 1 of *genConnectStatus* = 1.	The utility server receives and retains the updated *DERStatus* payload.
5	Action is taken to change the DERStatus to 'non-operating'. Where a client manages multiple DER all DER are disconnected / reconnected as part of this test. The client PUTs to the utility server with bit 2 of *genConnectStatus* = 0.	The utility server receives and retains the updated *DERStatus* payload.
6	Action is taken to change the DER status to 'operating'. The client PUTs to the utility server with bit 2 of *genConnectStatus* = 1.	The utility server receives and retains the updated *DERStatus* payload.

3.3.3.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to receive a valid *DERStatus* payload.

3.3.4 S-ALL-09 Operational Mode Status

3.3.4.1 Purpose

This test is intended to validate the utility server's ability to receive DER operational mode status from a client.

3.3.4.2 Precondition

The client is either simulating or managing a DER that is generating at least 50% of its rated active power.

3.3.4.3 Test Procedure

The steps SHALL be performed as described in Table 14. The test results SHALL be compared against the Expected Result column in Table 14.

Table 14 – Test steps – Operational Mode Status

Step No.	Manual or client step	Expected utility server result
1	Action is taken to change the OperationalModeStatus to cease offline. The client posts *OperationalModeStatus* = 1 to the utility server.	The utility server receives and retains the updated *DERStatus* payload.
2	Action is taken to change the OperationalModeStatus to cease operating. The client posts *OperationalModeStatus* = 2 to the utility server.	The utility server receives and retains the updated *DERStatus* payload.

3.3.4.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to receive a valid *DERStatus* payload.

3.3.5 S-ALL-10 Capabilities and Settings

3.3.5.1 Purpose

This test is intended to validate the utility server's ability to receive DER capabilities and settings from a client.

3.3.5.2 Test Procedure

The steps SHALL be performed as described in Table 15. The test results SHALL be compared against the Expected Result column in Table 15.

Table 15 – Test steps – Capabilities and Settings

Step No.	Manual or client step	Expected utility server result
1	The client completes the discovery process as detailed in test 3.2.1, including receiving the *DERCapability* and *DERSettings* resource links from the utility server. The client puts *DERCapability* and *DERSettings* payloads to the utility server on-connection.	The utility server receives and retains the updated *DERCapability* and *DERSettings* payloads.
2	The client PUTs new values for both *DERCapability* and *DERSettings* payloads to the utility server, reflecting the addition of an additional DER managed under the EndDevice.	The utility server receives and retains the new *DERCapability* and *DERSettings* payloads responding with a HTTP code of 2XX to confirm acceptance. None of the original DERCapability or DERSettings values are retained.

3.3.5.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to receive a valid *DERCapability* or *DERSettings* payload.

3.3.6 S-OPT-05 Update Telemetry Post Rates

3.3.6.1 Purpose

This test is intended to validate the utility server's ability to change posting rates for the Metering Mirror function set.

3.3.6.2 Precondition

The utility server shall be configured to change all default *MirrorUsagePoint* post-rates for the test client from 60s to 300s.

3.3.6.3 Test Procedure

The steps SHALL be performed as described in Table 16. The test results SHALL be compared against the Expected Result column in Table 16.

Table 16 – Test steps – Update Telemetry Post Rates

Step No.	Manual or client step	Expected utility server result

1	The client shall poll the *MirrorUsagePoint* endpoint.	The utility server responds with a *MirrorUsagePoint* payload with the *postRate* of all *MirrorUsagePoint* specified as 300s.

3.3.6.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The *MirrorUsagePoint* post rate retrieved by the client is not the expected value.

Notes:

i) Procurers of a utility server may wish to perform this test by varying the *postRate* of a specific resource (e.g. a given *MirrorUsagePoint* rather than all *MirrorUsagePoints*).

3.4 Control

3.4.1 Polled

3.4.1.1 S-ALL-11 Active Control – Export Limit

3.4.1.1.1 Purpose

This test is intended to validate the utility server's ability to provide active export controls using *opModExpLimW* to a client.

3.4.1.1.2 Precondition

The client is either simulating or managing a DER.

3.4.1.1.3 Test Procedure

The steps SHALL be performed as described in Table 17. The test results SHALL be compared against the Expected Result column in Table 17.

Table 17 – Test steps – Polled Active Controls, Export Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModExpLimW* = 0W and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new export limit.

3.4.1.1.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DERControlList* payload to the client.
  a) The server incorrectly specifies the details of the export limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the export limit.

3.4.1.2 S-ALL-12 Active Control – Generation Limit

3.4.1.2.1 Purpose

This test is intended to validate the utility server's ability to provide active generation controls using *opModGenLimW* to a client.

3.4.1.2.2 Precondition

The client is either simulating or managing a DER

3.4.1.2.3 Test Procedure

The steps SHALL be performed as described in Table 18. The test results SHALL be compared against the Expected Result column in Table 18.

Table 18 – Test steps – Polled Active Controls, Generation Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModGenLimW* = 0W and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new generation limit.

3.4.1.2.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DERControlList* payload to the client.
  a) The server incorrectly specifies the details of the generation limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the generation limit.

3.4.1.3 S-ALL-13 Active Control – Import Limit

3.4.1.3.1 Purpose

This test is intended to validate the utility server's ability to provide import controls using *opModImpLimW* to a client.

3.4.1.3.2 Precondition

The client is either simulating or managing a DER.

3.4.1.3.3 Test Procedure

The steps SHALL be performed as described in Table 19. The test results SHALL be compared against the Expected Result column in Table 19.

Table 19 – Test steps – Polled Active Controls, Import Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModImpLimW* = 0W and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new import limit.

3.4.1.3.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DERControlList* payload to the client.
  a) The server incorrectly specifies the details of the import limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the import limit.

3.4.1.4 S-ALL-14 Active Control – Load Limit

3.4.1.4.1 Purpose

This test is intended to validate the utility server's ability to provide load controls using *opModLoadLimW* to a client.

3.4.1.4.2 Precondition

The client is either simulating or managing a DER.

3.4.1.4.3 Test Procedure

The steps SHALL be performed as described in Table 20. The test results SHALL be compared against the Expected Result column in Table 20.

Table 20 – Test steps – Polled Active Controls, Load Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with	The utility server responds with the *DERControlList* payload containing the new load limit.

	opModLoadLimW = 0W and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	

3.4.1.4.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DERControlList* payload to the client.
  a) The server incorrectly specifies the details of the load limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the load limit.

3.4.1.5 S-ALL-15 Active Control – Energise / De-energise

3.4.1.5.1 Purpose

This test is intended to validate the utility server's ability to provide commands to energise & de-energise DER using *opModEnergize* to a client.

3.4.1.5.2 Precondition

The client has *opModEnergize* set to *true*.

3.4.1.5.3 Test Procedure

The steps SHALL be performed as described in Table 21. The test results SHALL be compared against the Expected Result column in Table 21.

Table 21 – Test steps – Polled Active Controls, Energize/De-energize

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModEnergize* = *false* and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new de-energise control.
2	The utility server is configured with an active control with *opModEnergize* = *true* and	The utility server responds with the *DERControlList* payload containing the new re-energise control.

	without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate	

3.4.1.5.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DERControlList* payload to the client.
  a) The server incorrectly specifies the details of the control.

3.4.1.6 S-ALL-16 Active Control – Disconnect

3.4.1.6.1 Purpose

This test is intended to validate the utility server's ability to provide commands to connect & disconnect DER using *opModConnect* to a client.

3.4.1.6.2 Precondition

The client has *opModConnect* set to *true*.

3.4.1.6.3 Test Procedure

The steps SHALL be performed as described in Table 22. The test results SHALL be compared against the Expected Result column in Table 22.

Table 22 – Test steps – Polled Active Controls, Disconnect

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModConnect* = *false* and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new de-energise control.
2	The utility server is configured with an active control with *opModConnect* = *true* and without including the rampTms attribute.	The utility server responds with the *DERControlList* payload containing the new reconnect control.

	The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate	

3.4.1.6.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DERControlList* payload to the client.
  a) The server incorrectly specifies the details of the control.

3.4.1.7 S-ALL-17 Active Control – Control Combinations

3.4.1.7.1 Purpose

This test is intended to validate the utility server's ability to provide combinations of controls to a client.

3.4.1.7.2 Test Procedure

The steps SHALL be performed as described in Table 23. The test results SHALL be compared against the Expected Result column in Table 23.

Table 23 – Test steps – Polled Active Controls, Control Combinations

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with: • *opModExpLimW* = 50% of the DER active power rating • *opModGenLimW* = 0W • *opModImpLimW* = 50% of the DER active power rating • *opModLoadLimW* = 0W • *opModEnergize* = false • *opModConnect* = false and without including the rampTms attribute. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new control, specifying all of the individual control types.

3.4.1.7.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The server does not provide the *DERControlList* payload to the client.
b) The server incorrectly specifies the details of the control.

3.4.1.8 S-ALL-18 Active Control – Ramp Rates

3.4.1.8.1 Purpose

This test is intended to validate the utility server's ability to provide active control ramp-rates to a client.

3.4.1.8.2 Precondition

The client is either simulating or managing a DER.

3.4.1.8.3 Test Procedure

The steps SHALL be performed as described in Table 24. The test results SHALL be compared against the Expected Result column in Table 24.

Table 24 – Test steps – Polled Active Controls, Ramp Rates

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModExpLimW* = 0W and *opModImpLimW* = 0W with rampTms = 30 seconds. The client polls the utility server's *DERControlListLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DERControlList* payload containing the new export limit and ramp-rate.

3.4.1.8.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The server does not provide the *DERControlList* payload including the active control ramp-rate to the client.

3.4.1.9 S-ALL-19 Default Control – Export Limit

3.4.1.9.1 Purpose

This test is intended to validate the utility server's ability to provide a *DefaultDERControl* export limit to a client.

3.4.1.9.2 Precondition

The client is either simulating or managing a DER.

3.4.1.9.3 Test Procedure

The steps SHALL be performed as described in Table 25. The test results SHALL be compared against the Expected Result column in Table 25.

Table 25 – Test steps – Polled Default Controls, Export Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModExpLimW* = 30% of the DER's rated active power. The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DefaultDERControl* payload containing the new export limit.

3.4.1.9.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DefaultDERControl* payload to the client.
  a) The server incorrectly specifies the details of the export limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the export limit.

3.4.1.10 S-ALL-20 Default Control – Generation Limit

3.4.1.10.1 Purpose

This test is intended to validate the utility server's ability to provide a *DefaultDERControl* generation limit to a client.

3.4.1.10.2 Precondition

The client is either simulating or managing a DER.

3.4.1.10.3 Test Procedure

The steps SHALL be performed as described in Table 26. The test results SHALL be compared against the Expected Result column in Table 26.

Table 26 – Test steps – Polled Default Controls, Generation Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModGenLimW* = 30% of the DER's rated active power.	The utility server responds with the *DefaultDERControl* payload containing the new generation limit.

	The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	

3.4.1.10.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DefaultDERControl* payload to the client.
  a) The server incorrectly specifies the details of the generation limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the generation limit.

3.4.1.11 S-ALL-21 Default Control – Import Limit

3.4.1.11.1 Purpose

This test is intended to validate the utility server's ability to provide a *DefaultDERControl* import limit to a client.

3.4.1.11.2 Precondition

The client is either simulating or managing a DER.

3.4.1.11.3 Test Procedure

The steps SHALL be performed as described in Table 27. The test results SHALL be compared against the Expected Result column in Table 27.

Table 27 – Test steps – Polled Default Controls, Import Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModImpLimW* = 30% of the DER's rated active power. The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DefaultDERControl* payload containing the new import limit.

3.4.1.11.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DefaultDERControl* payload to the client.
  a) The server incorrectly specifies the details of the import limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the import limit.

3.4.1.12 S-ALL-22 Default Control – Load Limit

3.4.1.12.1 Purpose

This test is intended to validate the utility server's ability to provide a *DefaultDERControl* load limit to a client.

3.4.1.12.2 Precondition

The client is either simulating or managing a DER.

3.4.1.12.3 Test Procedure

The steps SHALL be performed as described in Table 28. The test results SHALL be compared against the Expected Result column in Table 28.

Table 28 – Test steps – Polled Default Controls, Load Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModLoadLimW* = 30% of the DER's rated active power. The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DefaultDERControl* payload containing the new load limit.

3.4.1.12.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DefaultDERControl* payload to the client.
  a) The server incorrectly specifies the details of the load limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the load limit.

3.4.1.13 S-ALL-23 Default Controls – Ramp Rate

3.4.1.13.1 Purpose

This test is intended to validate the utility server's ability to update ramp-rate values with a *DefaultDERControl*.

3.4.1.13.2 Test Procedure

The steps SHALL be performed as described in Table 29. The test results SHALL be compared against the Expected Result column in Table 29.

Table 29 – Test steps – Polled Default Controls, Ramp Rate

Step No.	Manual or client step	Expected utility server result

1	The utility server is configured with a default control with *setGradW* = 1% per second. The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DefaultDERControl* payload containing the new ramp-rate.

3.4.1.13.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The server does not provide the *DefaultDERControl* payload to the client.
  a) The server incorrectly specifies the details of the ramp-rate.

3.4.1.14 S-ALL-24 Default Control – Control Combinations

3.4.1.14.1 Purpose

This test is intended to validate the utility server's ability to provide combinations of default controls to a client.

3.4.1.14.2 Test Procedure

The steps SHALL be performed as described in Table 30. The test results SHALL be compared against the Expected Result column in Table 30.

Table 30 – Test steps – Polled Default Controls, Control Combinations

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with: • *opModExpLimW* = 50% of the DER active power rating • *opModGenLimW* = 0W • *opModImpLimW* = 50% of the DER active power rating • *opModLoadLimW* = 0W The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	The utility server responds with the *DefaultDERControl* payload containing the new control, specifying all of the individual control types.

3.4.1.14.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The server does not provide the *DefaultDERControlList* payload to the client.
b) The server incorrectly specifies the details of the default control.

3.4.2 Subscription / Notification

3.4.2.1 S-ALL-25 Subscribe

3.4.2.1.1 Purpose

This test is intended to validate test the utility server's ability to support subscription for updates to the each of following:

- *EndDeviceList*
- *FunctionSetAssignmentsList*
- *DERProgramList*
- *DERControlList*
- *DefaultDERControl*

3.4.2.1.2 Precondition

The client has completed the Discovery process as defined in either test S-ALL-01 or test S-ALL-02.

3.4.2.1.3 Test Procedure

The steps SHALL be performed as described in Table 31. The test results SHALL be compared against the Expected Result column in Table 31.

Table 31 – Test steps – Subscribe

Step No.	Manual or client step	Expected utility server result
1	An *EndDeviceList* subscription request is triggered within the client system (either manually or as an automated response to completing the Discovery process). The client requests the creation a Subscription resource for the *EndDeviceList* resource, specifying the *NotificationURI* for the server to send notifications to, and posts this to the appropriate endpoint on the utility server.	The utility server responds with a 2XX HTTP response confirming creation of the subscription.
2	The utility server is configured to update the *pollRate* for an *EndDevice* associated with the client under test to 30 seconds.	The utility server sends a notification for the updated *EndDeviceList* to the relevant endpoint provided by the client.
3	An *FunctionSetAssignmentsList* subscription request is triggered within the client system. The client creates a Subscription resource for the utility server's *FunctionSetAssignmentsListLink* endpoint,	The utility server responds with a 2XX HTTP response confirming acceptance of the subscription.

	specifying the *NotificationURI* for the server to send notifications to, and posts this to the appropriate endpoint on the utility server.	
4	The utility server is configured to assign a *FunctionSetAssignment* to the same *EndDevice* updated in Step 2.	The utility server sends a notification for the updated *FunctionSetAssignmentsList* including this *FunctionSetAssignment* to the relevant endpoint provided by the client.
5	An *DERProgramList* subscription request is triggered within the client. The client creates a Subscription resource for the utility server's *DERProgramListLink* endpoint, specifying the *NotificationURI* for the server to send notifications to, and posts this to the appropriate endpoint on the utility server.	The utility server responds with a 2XX HTTP response confirming acceptance of the subscription.
6	The utility server is configured to assign a new *DERProgramList* to the same *FunctionSetAssignment* assigned in Step 4.	The utility server sends a notification for the updated *DERProgramList* to the relevant endpoint provided by the client.
7	An *DERControlList* subscription request is triggered within the client. The client creates a Subscription resource for the utility server's *DERControlListLink* endpoint, specifying the *NotificationURI* for the server to send notifications to, and posts this to the appropriate endpoint on the utility server.	The utility server responds with a 2XX HTTP response confirming acceptance of the subscription.
8	The utility server is configured to add a new *DERControl* to the same *EndDevice* updated in Step 2.	The utility server sends a notification for the updated *DERControlList* to the relevant endpoint provided by the client.
9	An *DefaultDERControl* subscription request is triggered within the client. The client creates a Subscription resource for the utility server's *DefaultDERControlLink* endpoint, specifying the *NotificationURI* for the server to send notifications to, and posts this to the appropriate endpoint on the utility server.	The utility server responds with a 2XX HTTP response confirming acceptance of the subscription.
10	The utility server is configured to update the value of the *DefaultDERControl* to the same *EndDevice* updated in Step 2.	The utility server sends a notification for the updated *DefaultDERControl* to the relevant endpoint provided by the client.

3.4.2.1.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

i) The utility server rejects a valid subscription request.

- The utility server fails to post a notification when required to do so.

Notes:

A. HTTP acknowledgment is used in this test to verify receipt of several notification types. Such acknowledgement typically occurs within a few seconds of the respective notification being sent, however can be significantly slower depending on network traffic and client operations.

3.4.2.2 S-ALL-26 Update and Delete Subscription

3.4.2.2.1 Purpose

This test is intended to validate test the utility server's ability to support a client deleting an existing subscription.

3.4.2.2.2 Precondition

The client has completed the Discovery process as defined in either test S-ALL-01 or test S-ALL-02 and the Subscription process as defined in test S-ALL-25.

3.4.2.2.3 Test Procedure

The steps SHALL be performed as described in Table 32. The test results SHALL be compared against the Expected Result column in Table 32.

Table 32 – Test steps – Delete Subscription

Step No.	Manual or client step	Expected utility server result
1	The client renews the subscription previously created by submitting a POST request to the relevant SubscriptionList with the same Resource.	The utility server responds with a 2XX HTTP response confirming the update to the subscription resource.
2	The client makes a DELETE requesrt against the subscription that was created in step 1.	The utility server responds with a 2XX HTTP response confirming deletion of the subscription.

3.4.2.2.4 Post-test

- The client resubscribes to the *DERControlList* as described in test S-ALL-25.

3.4.2.2.5 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server rejects a valid update- or delete-subscription request.

Notes:

A. HTTP acknowledgment is used in this test to verify receipt of several notification types. Such acknowledgement typically occurs within a few seconds of the respective notification being sent, however can be significantly slower depending on network traffic and client operations.

3.4.2.3 S-ALL-27 Active Controls – Export Limit

3.4.2.3.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed active controls using *opModExpLimW*.

3.4.2.3.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.3.3 Test Procedure

The steps SHALL be performed as described in Table 33. The test results SHALL be compared against the Expected Result column in Table 33.

Table 33 – Test steps – Subscription Active Controls, Export Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModExpLimW* = 0W and without including the rampTms attribute.	The utility server posts the new export limit to the client's *Notification* endpoint configured during the subscription process.

3.4.2.3.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the export limit to the client's *DERControlList* endpoint.
  a) The server incorrectly specifies the details of the export limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the export limit.

3.4.2.4 S-ALL-28 Active Controls – Generation Limit

3.4.2.4.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed active controls using *opModGenLimW*.

3.4.2.4.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.4.3 Test Procedure

The steps SHALL be performed as described in Table 34. The test results SHALL be compared against the Expected Result column in Table 34.

Table 34 – Test steps – Subscription Active Controls, Generation Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModGenLimW* = 0W and without including the rampTms attribute.	The utility server posts the new generation limit to the client's *DERControlList* endpoint configured during the subscription process.

3.4.2.4.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the generation limit to the client's *DERControlList* endpoint.
  a) The server incorrectly specifies the details of the generation limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the generation limit.

3.4.2.5 S-ALL-29 Active Controls – Import Limit

3.4.2.5.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed active controls using *opModImpLimW*.

3.4.2.5.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.5.3 Test Procedure

The steps SHALL be performed as described in Table 35. The test results SHALL be compared against the Expected Result column in Table 35.

Table 35 – Test steps – Subscription Active Controls, Import Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModImpLimW* = 0W and without including the rampTms attribute.	The utility server posts the new import limit to the client's *DERControlList* endpoint configured during the subscription process.

3.4.2.5.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the import limit to the client's *DERControlList* endpoint.
  a) The server incorrectly specifies the details of the import limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the import limit.

3.4.2.6 S-ALL-30 Active Controls – Load Limit

3.4.2.6.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed active controls using *opModLoadLimW*.

3.4.2.6.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.6.3 Test Procedure

The steps SHALL be performed as described in Table 36. The test results SHALL be compared against the Expected Result column in Table 36.

Table 36 – Test steps – Subscription Active Controls, Load Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModLoadLimW* = 0W and without including the rampTms attribute.	The utility server posts the new load limit to the client's *DERControlList* endpoint configured during the subscription process.

3.4.2.6.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the load limit to the client's *DERControlList* endpoint.
  a) The server incorrectly specifies the details of the load limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the load limit.

3.4.2.7 S-ALL-31 Active Controls – Energise

3.4.2.7.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed active controls using *opModEnergize*.

3.4.2.7.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25 and has *opModEnergize* set to *true*.

3.4.2.7.3 Test Procedure

The steps SHALL be performed as described in Table 37. The test results SHALL be compared against the Expected Result column in Table 37.

Table 37 – Test steps – Subscription Active Controls, Energise

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModEnergize* = *false* and without including the rampTms attribute.	The utility server posts the new load limit to the client's *DERControlList* endpoint configured during the subscription process.

3.4.2.7.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the de-energise control to the client's *DERControlList* endpoint.
  a) The server incorrectly specifies the details of the control.

3.4.2.8 S-ALL-32 Active Controls – Disconnect

3.4.2.8.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed active controls using *opModConnect*.

3.4.2.8.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25 and has *opModConnect* set to *true*.

3.4.2.8.3 Test Procedure

The steps SHALL be performed as described in Table 38. The test results SHALL be compared against the Expected Result column in Table 38.

Table 38 – Test steps – Subscription Active Controls, Disconnect

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModConnect* = *false* and without including the rampTms attribute.	The utility server posts the new load limit to the client's *DERControlList* endpoint configured during the subscription process.

3.4.2.8.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the disconnect control to the client's *DERControlList* endpoint.
  a) The server incorrectly specifies the details of the control.

3.4.2.9 S-ALL-33 Active Control – Control Combinations

3.4.2.9.1 Purpose

This test is intended to validate the utility server's ability to provide combinations of controls to a client.

3.4.2.9.2 Precondition

The client has *opModConnect* set to *true*. The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.9.3 Test Procedure

The steps SHALL be performed as described in Table 39. The test results SHALL be compared against the Expected Result column in Table 39.

Table 39 – Test steps – Polled Active Controls, Control Combinations

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with: • *opModExpLimW* = 50% of the DER active power rating • *opModGenLimW* = 0W • *opModImpLimW* = 50% of the DER active power rating • *opModLoadLimW* = 0W • *opModEnergize* = false • *opModConnect* = false and without including the rampTms attribute.	The utility server posts the new load limit to the client's *DERControlList* endpoint configured during the subscription process, specifying all of the individual control types.

3.4.2.9.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The server does not provide the *DERControlList* payload to the client.
b) The server incorrectly specifies the details of the control.

3.4.2.10 S-ALL-34 Active Control – Ramp Rates

3.4.2.10.1 Purpose

This test is intended to validate the utility server's ability to provide active control ramp-rates to a client.

3.4.2.10.2 Precondition

The client is either simulating or managing a DER. The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.10.3 Test Procedure

The steps SHALL be performed as described in Table 40. The test results SHALL be compared against the Expected Result column in Table 40.

Table 40 – Test steps – Polled Active Controls, Ramp Rates

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModExpLimW* = 0W and *opModImpLimW* = 0W with rampTms = 30 seconds.	The utility server posts the the *DERControlList* payload containing the new export limit and ramp-rate to the client's DERControlList endpoint configured during the subscription process, specifying all of the individual control types.

3.4.2.10.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The server does not provide the *DERControlList* payload including the active control ramp-rate to the client.

3.4.2.11 S-ALL-35 Default Controls – Export Limit

3.4.2.11.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed default controls using *opModExpLimW*.

3.4.2.11.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.11.3 Test Procedure

The steps SHALL be performed as described in Table 41. The test results SHALL be compared against the Expected Result column in Table 41.

Table 41 – Test steps – Subscription Default Controls, Export Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModExpLimW* = 0W.	The utility server posts the new export limit to the client's *DefaultDERControl* endpoint configured during the subscription process.

3.4.2.11.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:The utility server does not post the export limit to the client's *DefaultDERControl* endpoint.

a) The utility server does not post the export limit to the client's *DefaultDERControl* endpoint.
b) The server incorrectly specifies the details of the export limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the export limit.

3.4.2.12 S-ALL-36 Default Controls – Generation Limit

3.4.2.12.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed default controls using *opModGenLimW*.

3.4.2.12.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.12.3 Test Procedure

The steps SHALL be performed as described in Table 42. The test results SHALL be compared against the Expected Result column in Table 42.

Table 42 – Test steps – Subscription Default Controls, Generation Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModGenLimW* = 0W.	The utility server posts the new generation limit to the client's *DefaultDERControl* endpoint configured during the subscription process.

3.4.2.12.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the generation limit to the client's *DefaultDERControl* endpoint.
  a) The server incorrectly specifies the details of the generation limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the generation limit.

3.4.2.13 S-ALL-37 Default Controls – Import Limit

3.4.2.13.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed default controls using *opModImpLimW*.

3.4.2.13.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.13.3 Test Procedure

The steps SHALL be performed as described in Table 43. The test results SHALL be compared against the Expected Result column in Table 43.

Table 43 – Test steps – Subscription Default Controls, Import Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModImpLimW* = 0W.	The utility server posts the new import limit to the client's *DefaultDERControl* endpoint configured during the subscription process.

3.4.2.13.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the import limit to the client's *DefaultDERControl* endpoint.
  a) The server incorrectly specifies the details of the import limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the import limit.

3.4.2.14 S-ALL-38 Default Controls – Load Limit

3.4.2.14.1 Purpose

This test is intended to validate the utility server's ability to notify a client of subscribed default controls using *opModLoadLimW*.

3.4.2.14.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.14.3 Test Procedure

The steps SHALL be performed as described in Table 44. The test results SHALL be compared against the Expected Result column in Table 44.

Table 44 – Test steps – Subscription Default Controls, Load Limit

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *opModILoadLimW* = 0W.	The utility server posts the new load limit to the client's *DefaultDERControl* endpoint configured during the subscription process.

3.4.2.14.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not post the load limit to the client's *DefaultDERControl* endpoint.

a) The server incorrectly specifies the details of the load limit, including by correctly matching the *value* and *multiplier* elements to the intended value of the load limit.

3.4.2.15 S-ALL-39 Default Controls – Ramp Rate

3.4.2.15.1 Purpose

This test is intended to validate the utility server's ability to notify a client of updates to subscribed ramp-rate values on *DefaultDERControl*.

3.4.2.15.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.15.3 Test Procedure

The steps SHALL be performed as described in Table 45. The test results SHALL be compared against the Expected Result column in Table 45.

Table 45 – Test steps – Subscription Default Controls, Ramp Rate

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a default control with *setGradW* = 1%.	The utility server posts a default control with the new ramp-rate to the client's *DefaultDERControl* endpoint configured during the subscription process.

3.4.2.15.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

The utility server does not post the updated default control to the client's *DefaultDERControl* endpoint.

a) The server incorrectly specifies the details of the ramp-rate.

3.4.2.16 S-ALL-40 Default Control – Control Combinations

3.4.2.16.1 Purpose

This test is intended to validate the utility server's ability to provide combinations of default controls to a client.

3.4.2.16.2 Precondition

The client has subscribed to the DERControlList resource as defined in test S-ALL-25.

3.4.2.16.3 Test Procedure

The steps SHALL be performed as described in Table 46. The test results SHALL be compared against the Expected Result column in Table 46.

Table 46 – Test steps – Polled Default Controls, Control Combinations

Step No.	Manual or client step	Expected utility server result

1	The utility server is configured with a default control with: • *opModExpLimW* = 50% of the DER active power rating • *opModGenLimW* = 0W • *opModImpLimW* = 50% of the DER active power rating • *opModLoadLimW* = 0W The client polls the utility server's *DefaultDERControlLink* endpoint at the expected interval based on the default poll-rate.	The utility server posts the new control to the client's *DefaultDERControl* endpoint configured during the subscription process, specifying all of the individual control types.

3.4.2.16.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The server does not provide the *DefaultDERControlList* payload to the client.
b) The server incorrectly specifies the details of the default control.

3.4.3 S-ALL-41 Control Responses

3.4.3.1 Purpose

This test is intended to validate the utility server's ability to receive control responses from a client for the following control events:

a) Event received
b) Event started
c) Event completed
d) The event has been cancelled
e) The event has been superseded

3.4.3.2 Test Procedure

The steps SHALL be performed as described in Table 47. The test results SHALL be compared against the Expected Result column in Table 47.

Table 47 – Test steps – Control Responses

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power, *responseRequired* =	The utility server receives and retains the 'received' control response. The server updates the control's EventStatus to

	03 and *replyTo* set to the response endpoint on the test utility server. On the next communication of *DERControlList* to the client (either by notification or polling depending on client type) it receives the active control and posts back control response status = 1 (received).	'Scheduled', which shall be reflected in future DERControlList polls/notifications.
2	The DER starts the active control, and the client posts back control response status = 2 (started).	The utility server receives and retains the 'started' control response. The server updates the control's EventStatus to 'Active', which shall be reflected in future DERControlList polls/notifications.
3	The DER completes the active control, and the client posts back control response status = 3 (completed).	The utility server receives and retains the 'completed' control response.
4	The utility server is configured with an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power, *responseRequired* = 03 and *replyTo* set to the response endpoint on the test utility server, to begin in 5 minutes. The client posts back control response status = 1 (received) and after 5 minutes commences the control and posts back control response status = 2 (started).	The utility server receives and retains the 'received' and 'started' control responses. The server updates the control's EventStatus to 'Active, which shall be reflected in future DERControlList polls/notifications.
5	While the control created in step 4 is in operation, the utility server is configured to cancel it. On the next communication of *DERControlList* to the client (either by notification or polling depending on client type) it receives the cancellation, ends the active control and posts back control response status = 6 (cancelled).	The utility server receives and retains the 'cancelled' control response. The server updates the control's EventStatus to 'Cancelled', which shall be reflected in future DERControlList polls/notifications.
6	The utility server is configured with an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power, *responseRequired* = 03 and *replyTo* set to the response	The utility server receives and retains the 'received' control response. The server updates the control's EventStatus to

	endpoint on the test utility server, to begin in 10 minutes. The client posts back control response status = 1 (received).	'Scheduled', which shall be reflected in future DERControlList polls/notifications.
7	Prior to the start of the control created in step 6, the utility server is configured to cancel the control. On next poll/notification of the *DERProgramList* the client receives the cancellation and posts back control response status = 6 (cancelled).	The utility server receives and retains the 'cancelled' control response. The server updates the control's EventStatus to 'Cancelled', which shall be reflected in future DERControlList polls/notifications.
8	The utility server configures an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power to begin in 10 minutes. The client posts back control response status = 1 (received).	The utility server receives and retains the 'received' control response. The server updates the control's EventStatus to 'Scheduled', which shall be reflected in future DERControlList polls/notifications.
9	Prior to the commencement of the previous control, the utility server is configured to create a second active control with the same primacy and an overlapping control window. On the next communication of *DERProgramList* to the client (either by notification or polling depending on client type) it receives the new active control and posts back control response status = 7 (superseded).	The utility server receives and retains the 'superseded' control response. The server updates the control's EventStatus to 'Superseded', which shall be reflected in future DERControlList polls/notifications.

3.4.3.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not correctly receive and retain any of the control responses communicated by the client.

3.4.4 S-ALL-42 Delayed Response to Cancelled Export Control

3.4.4.1 Purpose

This test is intended to validate a utility server's robustness to client behaviour in response to cancelled active export limit controls.

3.4.4.2 Precondition

The client is either simulating or managing a DER that is exporting at the connection point at least 50% of its rated active power. The utility server has a current active control in operation with *opModExpLimW* set to 200% of the DER's rated active power, and a default DER control in-place with *opModExpLimW* set to 30% of the DER's rated active power.

3.4.4.3 Test Procedure

The steps SHALL be performed as described in Table 48. The test results SHALL be compared against the Expected Result column in Table 48.

Table 48 – Test steps – Delayed Response to Cancelled Export Control

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured to cancel all active controls.	On the next communication of *DERControls* to the client (either by notification or polling depending on client type) the utility server will update the cancelled active control to include *EventStatus* = *2* (Cancelled).
2	The client does not immediately provide a *DERControlResponse* to the utility server. The client repolls the utility server's *DERControlListLink* endpoint at the respective poll-rate.	The payload provided by the utility server continues to include *EventStatus* = *2* (Cancelled) for the cancelled active control.
3	The client posts a *DERControlResponse* to the utility server's *DERControlResponse* for the cancelled active control with *status* = *6* (cancelled).	On the next communication of *DERControls* to the client (either by notification or polling depending on client type) the utility server shall update the *currentStatus* of the *DERControl* to 2 (Cancelled).

3.4.4.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not retain the cancelled active control until it receives the 'cancelled' control response from the client.

3.4.5 S-ALL-43 Update Poll Rates

3.4.5.1 Purpose

This test is intended to validate the utility server's ability to configure new polling rates for the following resources and communicate them to a client.

- *DeviceCapability*

- *EndDeviceList*
- *FunctionSetAssignmentsList*
- *DERProgramList*
- *DERList*

3.4.5.2 Test Procedure

The steps SHALL be performed as described in Table 49. The test results SHALL be compared against the Expected Result column in Table 49.

Table 49 – Test steps – Poll Rates

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured to update the client's *DeviceCapability* poll-rate from 300s to 60s.	On the next communication of *DeviceCapability* to the client the payload will include pollRate = 60.
2	The utility server is configured to update the client's *EndDeviceList* poll-rate from 300s to 60s.	On the next communication of *EndDeviceList* to the client (either by notification or polling depending on client type) the payload will include pollRate = 60.
3	The utility server is configured to update the client's *FunctionSetAssignments* poll-rate from 300s to 60s.	On the next communication of *FunctionSetAssignmentsList* to the client (either by notification or polling depending on client type) the payload will include pollRate = 60.
4	The utility server is configured to update the client's *DERProgramList* poll-rate from to 60s to 120s.	On the next communication of *DERProgramList* to the client (either by notification or polling depending on client type) the payload will include pollRate = 120.
5	The utility server is configured to update the client's *DERList* poll-rate from 60s to 120s.	On the next communication of *DERList* to the client the payload will include pollRate = 120.
6	The utility server is configured to update the client's *MirrorUsagePoint* poll-rate from 60s to 120s.	On the next communication of *MirrorUsagePointList* to the client (either by notification or polling depending on client type) the payload will include pollRate = 120.

3.4.5.3 Post-test

a) The utility server updates all poll rates back to their default values as described in Table 1.

3.4.5.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not include the *pollRate* elements in any of the payloads specified.
- The utility server does not update any of the the *pollRates* to the intended values.

Notes:

- Utility server vendors may wish to have this test performed by varying the *pollRate* of a specific resource (e.g. a given *EndDevice* rather than all *EndDevices*) in order to demonstrate that they support this functionality.

3.4.6 S-ALL-44 Scheduling

3.4.6.1 Purpose

This test is intended to validate the utility server's ability to communicate a schedule of controls to a client, including pagination.

3.4.6.2 Test Procedure

The steps SHALL be performed as described in Table 50. The test results SHALL be compared against the Expected Result column in Table 50.

Table 50 – Test steps – Scheduling

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured to generate a schedule of 24 sequential, non-overlapping controls with a range of export and/or import limits, each with 1min duration.	On the next communication of *DERControls* to the client (either by notification or polling depending on client type) the payload will be a *DERControlList* including all 24 configured *DERControls*.
2	The client polls the utility server's *DERControlListLink* endpoint to retrieve only the earliest 10 controls, by setting s=0 and l=10 in the URI parameters.	On the next communication of *DERControls* to the client (either by notification or polling depending on client type) the payload will be a *DERControlList* including only the most recent *DERControls* configured in the utility server.

3.4.6.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The utility server does provide the schedule of controls as requested by the client.

3.4.7 S-ALL-45 Randomisation

3.4.7.1 Purpose

This test is intended to validate the utility server's ability to specify randomization to clients by using the *randomizeStart* attribute.

3.4.7.2 Test Procedure

The steps SHALL be performed as described in Table 51. The test results SHALL be compared against the Expected Result column in Table 51.

Table 51 – Test steps – Randomisation

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a schedule of five sequential, non-overlapping active controls, each with the following attributes: • Duration = 120s • *randomizeStart* = 60s • *opModExpLimW* = 200% of the DER's rated active power	On the next communication of *DERControls* to the client (either by notification or polling depending on client type) the payload will be a *DERControlList* including all 5 configured *DERControls*. Each control will include *randomizeStart = 60*.

3.4.7.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not provide the *randomizeStart* parameters in the controls.
- The utility server provides incorrect values for any of the *randomizeStart* parameters in the controls.

3.4.8 S-ALL-46 Communication Loss

3.4.8.1 Purpose

This test is intended to validate the utility server's ability to appropriate respond to a loss of communications with a subscription / notification client.

3.4.8.2 Precondition

A DER is under the management of the client and is importing or exporting at the connection point at least 50% of its rated active power. The utility has a current active control in operation, with *opModImpLimW* / *opModExpLimW* set to 200% of the DER's rated active power.

3.4.8.3 Test Procedure

The steps SHALL be performed as described in Table 52. The test results SHALL be compared against the Expected Result column in Table 52.

Table 52 – Test steps – Communications Loss

Step No.	Manual or client step	Expected utility server result

1	Communications between the utility server and client are lost. The utility server is configured with a *DERControl* with *opModExpLimW* / *opModImpLimW* = 0.	The utility server attempts to notify the client of the new control, but is unable to do so due to the lack of comms.
2	Communications between the utility server and client are reestablished 5 minute after the expiry of the current schedule.	The utility server notifies the client with the new active control or removes the Subscription.

3.4.8.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not correctly notify the client of the control after communications are reestablished.

Notes:

- The tester is responsible for determining an appropriate way to create a loss of communications between the comms client and utility server. Examples may include disabling of physical networking (e.g. unplugging of Ethernet cable or turning off of WiFi router), shutting down the test utility server, etc.

3.4.9 S-ALL-47 Validation of Scaling Factors

3.4.9.1 Purpose

This test is intended to validate the utility server's ability to communicate and interpret scaling factors in controls.

3.4.9.2 Precondition

The client is either simulating or managing a DER that is exporting at the connection point at least 100kW.

3.4.9.3 Test Procedure

The steps SHALL be performed as described in Table 53. The test results SHALL be compared against the Expected Result column in Table 53.

Table 53 – Test steps – Validation of Scaling Factors

Step No.	Manual or client step	Expected utility server result
1	The utility server configures an active control with *opModExpLimW* = 100,000W. On the next poll of the utility server the client receives and	The payload of the *DERControl* received by the client has a *multiplier* of at least 1 due to the size of the limit.

	starts the active control by limited export power to no more than 100kW.	
2	The client provides telemetry to the utility server reflecting the active power exported. The payload of the *MirrorMeterReadings* shall have a *multiplier* value of at least 1 due to the size of the power flows.	The utility server receives and retains the telemetry data.
3	The client provides telemetry to the utility server reflecting the active power exported. The payload of the *MirrorMeterReadings* shall have a *multiplier* value of 0 and a value of 100,000.	The utility server receives and rejects the telemetry data by returning a 4XX HTTP response code.

3.4.9.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The utility server constructs an incorrect value for the export limit, either by constructing a limit that does not match the intended limit of 100kW or by specifying a *multiplier* of 0 and instead overflowing the *value* element's 16-bit capability.
b) The utility server does not reject the incorrectly-structured value in step 3.

Notes:

i) Due to the scales of power limits and telemetry involved in this test, it is likely to be efficient for this to be enacted by simulating a client and DER rather than using a real DER.

3.4.10 S-ALL-48 Changing Ramp-Rates

3.4.10.1 Purpose

This test is intended to validate that a utility server is able to correctly change a client's ramp-rate settings.

3.4.10.2 Precondition

The client is either simulating or managing a DER.

3.4.10.3 Test procedure

The steps SHALL be performed as described in Table 54. The test results SHALL be compared against the Expected Result column in Table 54.

Table 54 – Test steps – Changing Ramp-Rates

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with a *DefaultDERControl* with *setGradW* = 1% per second.	On the next communication of *DERProgramList* to the client (either by notification or polling depending on client type) the utility server provide the new *DefaultDERControl* with a new value for *setGradW*.

3.4.10.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not correctly specify the new value of setGradW in the updated default control.

3.4.11 S-ALL-49 Multiple DER Programs

3.4.11.1 Purpose

This test is intended to validate that a utility server properly supports multiple *FunctionSetAssignments* and *DERPrograms*.

3.4.11.2 Precondition

A DER capable of generating active power is under the management of the client and is capable of exporting at the connection point at least 50% of its rated active power. The utility server has configured two *FunctionSetAssignments* FSA1 and FSA2 to apply to the client, and each *FunctionSetAssignment* contains one *DERProgram* configured under it (DERP1 and DERP2 respectively). DERP1 has a primacy of 1 and DERP2 has a primacy of 2.

3.4.11.3 Test procedure

The steps SHALL be performed as described in Table 55. The test results SHALL be compared against the Expected Result column in Table 55.

Table 55 – Test steps for Physical Response – Primacy Validation for Generators

Step No.	Manual or client step	Expected utility server result
1	The utility server is configured with DERP1 having a default DER control in-place with *opModExpLimW* set to 30% of the DER's rated active power. The utility server configures an active control on DERP2 with *opModExpLimW* = 200% of the DER's rated active power. The client polls the utility server's *DERProgramList* end-point.	The utility server responds to the client polling providing the DERProgramList including the details for both DERP1 and DERP2.

3.4.11.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The utility server does not provide the correct details for both DERPrograms.

3.5 Ongoing Operations

3.5.1 S-ALL-50 Extended Operations

3.5.1.1 Purpose

This test is intended to validate the utility server's ability to maintain compliant communications and operations with a client over an extended period.

3.5.1.2 Test Procedure

The steps SHALL be performed as described in Table 56. The test results SHALL be compared against the Expected Result column in Table 56.

Table 56 – Test steps – Extended Operations

Step No.	Manual or client step	Expected utility server result
1	A connection between the utility server and client is established and maintained for 3 consecutive days (72 hours). This will include regularly requesting DERControls and providing DERControlResponses, and submitting MirrorMeterReadings, all at the expected post & poll rates (default to 1 minute).	The utility server and client maintain communications for this period.

3.5.1.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- Communications are not maintained throughout the specified period. Short periods of intermittent communication are acceptable provided they do not exceed 60 seconds.

Notes:

- As some DER clients are not always online due to reliance on a variable power source (e.g. sunlight) they may lose maintain communications when the source is unavailable (e.g. outside daylight hours). As such this test is best performed using either a simulated client or a DER with a reliable power source.

3.6 Edge Cases

3.6.1 General

Tests in this section are intended to ensurer that the utility server under test is robust to a range of edge cases that may occur during operations.

3.6.2 S-ALL-51 End Device Duplication

3.6.2.1 Purpose

This test is intended to validate that the utility server rejects attempts by a client to register a second EndDevice with the same LFDI.

3.6.2.2 Precondition

The client has completed at least one of the Discovery tests described in S-ALL-01, S-ALL-02 and S-ALL-03 including registration of its *EndDevice*.

3.6.2.3 Test Procedure

The steps SHALL be performed as described in Table 57. The test results SHALL be compared against the Expected Result column in Table 57.

Table 57 – Test steps – End Device Duplication

Step No.	Manual or client step	Expected utility server result
1	The client attempts to post to the utility server's *EndDeviceListLink* endpoint for an *EndDevice* with the same LDFI as previously configured.	The utility server receives and rejects the request by returning a HTTP response code of 4XX.

3.6.2.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to reject the duplication.

3.6.3 S-ALL-52 LFDI Doesn't Match Certificate

3.6.3.1 Purpose

This test is intended to validate that the utility server rejects the creation of a direct-client *EndDevice* whose LFDI does not the client's certificate.

3.6.3.2 Precondition

The client has not yet completed Discovery against the utility server.

3.6.3.3 Test Procedure

The steps SHALL be performed as described in Table 58. The test results SHALL be compared against the Expected Result column in Table 58.

Table 58 – Test steps – LFDI Doesn't Match Certificate

Step No.	Manual or client step	Expected utility server result
1	The client attempts to post to the *EndDevice* endpoint for an *EndDevice* that is not the one managed by the client.	The utility server receives and rejects the request by returning a HTTP response code of 4XX.

3.6.3.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to reject the access attempt.

3.6.4 S-ALL-53 Update Connection Point

3.6.4.1 Purpose

This test is intended to validate that the utility server allows a client to update a previously-specified *ConnectionPointId*.

3.6.4.2 Precondition

The client has completed at least one of the Discovery tests described in S-ALL-01, S-ALL-02 and S-ALL-03 including registration of its *EndDevice*, and has registered a ConnectionPoint via test S-OPT-03.

3.6.4.3 Test Procedure

The steps SHALL be performed as described in Table 59. The test results SHALL be compared against the Expected Result column in Table 59.

Table 59 – Test steps – Update Connection Point

Step No.	Manual or client step	Expected utility server result
1	The client overwrites the existing *ConnectionPointId* but putting a new value to the utility server's *ConnectionPointLink* endpoint.	The utility server returns a HTTP 2XX response confirming that the *ConnectionPoint* has been correctly received.
2	The client acesses the utility server's *ConnectionPointLink* endpoint to request the value of *ConnectionPointId* for the *EndDevice*.	The utility server returns a *ConnectionPoint* payload with the new value of *ConnectionPointId*.

3.6.4.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to update the *ConnectionPointId* to the new value.

3.6.5 S-ALL-54 Update Aggregator Certificate

3.6.5.1 Purpose

This test is intended to validate that the utility server allows an aggregator client to update its certificate and continue to function.

3.6.5.2 Precondition

The aggregator client has completed at least one of the Discovery tests described in S-ALL-01, S-ALL-02 and S-ALL-03 and has had its *EndDevice* registered out-of-band.

3.6.5.3 Test Procedure

The steps SHALL be performed as described in Table 60. The test results SHALL be compared against the Expected Result column in Table 60.

Table 60 – Test steps – Update Aggregator Certificate

Step No.	Manual or client step	Expected utility server result
1	The aggregator client is reissued a new certificate that is permitted to communiate with the utility server (e.g. is within the trust chain of the server's certificate). The utility server is configured to associate LFDI of the new certificate with the client. The client attempts to access the *EndDeviceListLink* endpoint for the *EndDevice* that is managed by the client.	Utility server sends the *EndDeviceList* resource which contains the relevant *EndDevice* information to the client.

3.6.5.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

a) The utility server does not provide access to the aggregator client utilising the new certificate.

3.6.6 S-ALL-55 Reject Incomplete DER Settings

3.6.6.1 Purpose

This test is intended to validate that the utility server correctly rejects a *DERSettings* payload without an *updatedTime* attribute.

3.6.6.2 Precondition

The client will have completed the Discovery process via one (or both) of tests S-ALL-01 and S-ALL-02.

3.6.6.3 Test Procedure

The steps SHALL be performed as described in Table 61. The test results SHALL be compared against the Expected Result column in Table 61.

Table 61 – Test steps – Reject Incomplete DER Settings

Step No.	Manual or client step	Expected utility server result
1	The client is manually updated to vary the value for *setMaxW*. The client puts an updated *DERSettings* payload to the utility server. This payload will not include the mandatory *updatedTime* attribute.	The utility server receives and rejects the updated *DERSettings* payload by responding with a 4XX HTTP code.
2	The client is manually updated to vary the value for *setMaxW* again. The client puts an updated *DERSettings* payload to the utility server. This payload will include the mandatory *updatedTime* attribute, however will construct the *modesEnabled* attribute as an integer rather than a bitmap.	The utility server receives and rejects the updated *DERSettings* payload by responding with a 4XX HTTP code.

3.6.6.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility does not correctly reject any of the invalid *DERSettings* payloads.
- The utility server incorrectly updates its internal representation of the *EndDevice*'s *DERSettings* with the new values

3.6.7 S-ALL-56 Reject String LFDIs

3.6.7.1 Purpose

This test is intended to validate that the utility server correctly rejects an LFDI provided by an aggregator client that is constructed as a string rather than a hexadecimal.

3.6.7.2 Precondition

The client has not completed any of the Discovery tests described in S-ALL-01, S-ALL-02 and S-ALL-03 (or has been factory reset to ensure that Discovery can be performed fresh).

3.6.7.3 Test Procedure

The steps SHALL be performed as described in Table 62. The test results SHALL be compared against the Expected Result column in Table 62.

Table 62 – Test steps – Reject String LFDIs

Step No.	Manual or client step	Expected utility server result
1	Client is powered on, and performs HTTP discovery against the pre-configured utility server URI information, commencing with the *DeviceCapability* endpoint.	Utility server sends the *DeviceCapability* resource including the *TimeLink*, *EndDeviceListLink* and *MirrorUsagePointListLink* resource links to the client.
2	Client receives *DeviceCapability* payload. Client requests resource information from the utility server's *EndDeviceList* endpoint.	Utility server sends the *EndDeviceList* resource, which does not contain an entry for the client's *EndDevice*.
3	Client receives the *EndDeviceList* payload, and posts to the *EndDeviceListLink* including the relevant information for the client's *EndDevice*. The client formats the LFDI as a string that uses non-hexidecimal characters.	The utility server detects and rejects the anomolous LFDI by returning a 400 HTTP response.

3.6.7.4 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to correctly reject the incorrectly-formatted LFDI within the *EndDevice* payload.

3.6.8 S-ALL-57 Invalid Control Responses

3.6.8.1 Purpose

This test is intended to validate the utility server's ability to receive and respond to invalid control responses from a client.

3.6.8.1.1 Precondition

If the client supports Subscription/Notification, the client has subscribed to the DERControlList resource as defined in test S-ALL-23.

3.6.8.2 Test Procedure

The steps SHALL be performed as described in Table 63. The test results SHALL be compared against the Expected Result column in Table 63.

Table 63 – Test steps – Invalid Control Responses

Step No.	Manual or client step	Expected utility server result

1	The utility server is configured with an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power, *responseRequired* = 03 and *replyTo* set to the response endpoint on the test utility server. On the next communication of *DERControlList* to the client (either by notification or polling depending on client type) it receives the active control and posts back control response = 1 (received). The control response will include an mRID for a *DERControl* that does not exist.	The utility server receives and rejects the control response by sending a 4XX HTTP response.
2	The utility server is configured with an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power, *responseRequired* = 03 and *replyTo* set to the response endpoint on the test utility server. On the next communication of *DERControlList* to the client (either by notification or polling depending on client type) it receives the active control and posts back control response = 1 (received). The control response will include an LFDI for an *EndDevice* that does not exist.	The utility server receives and rejects the control response by sending a 4XX HTTP response.
3	The utility server is configured with an active control with *opModExpLimW* / *opModImpLimW* = 200% of the DER's rated active power, *responseRequired* = 03 and *replyTo* set to the response endpoint on the test utility server. On the next communication of *DERControlList* to the client (either by notification or polling depending on client type) it receives the active control and posts back control response = 15 (reserved).	The utility server receives and rejects the control response by sending a 4XX HTTP response.

3.6.8.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server does not correctly reject any of the invalid control responses communicated by the client.

3.7 Demand Response Test Procedures

3.7.1 General

Tests in this section are applicable to utility servers that wish to manage demand response systems aligned with the AS/NZS 4755 framework alongside the DER controls described in the main body of CSIP-AUS.

3.7.2 S-OPT-06 DRED Operational Instruction Response

3.7.2.1 Purpose

This test is intended to validate the utility server's ability to serve operational instructions to a demand response client.

3.7.2.2 Test Procedure

The steps SHALL be performed as described in Table 64. The test results SHALL be compared against the Expected Result column in Table 64.

Table 64 – Test steps for Communications Client – DRED OI Response

Step No.	Manual or DER step	Expected utility server result
1	The utility server is configured with an active control with an immediate start and *opModConnect* = false (representing OI0).	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModConnect* = false.
2	The utility server is configured with an active control with an immediate start, a duration of five minutes and *opModConnect* = true to cease the previous DRM.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately with a duration of 5 minutes and with *opModConnect* = true.
3	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = -75% (representing OI3) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = -75%.
4	After the end of the previous control the utility server is configured with an active control	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server

	with *opModFixedW* = -50% (representing OI2) to start no more than 5 minutes in the future.	includes an active control configured to start immediately and with *opModFixedW* = -50%.
5	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = -0.01% (representing OI1) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = -0.01%.
6	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = -100% (representing OI4) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = -100%.
7	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = 75% (representing OI7) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = 75%.
8	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = 50% (representing OI6) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = 50%.
9	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = 0% (representing OI5) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = 0%.
10	After the end of the previous control the utility server is configured with an active control with *opModFixedW* = 100% (representing OI8) to start no more than 5 minutes in the future.	On the next polling of the *DERControlListLink* by the client the payload provided by the utility server includes an active control configured to start immediately and with *opModFixedW* = 100%.

3.7.2.3 Criteria

If during testing the utility server experiences one of the following events these are considered non-conformances with this requirement:

- The utility server fails to correctly construct an operating instruction and serve it to the client.