



## **OCA OCPP 2.0.1**

### **Part 6 - Test case document Charging Station Management System for OCTT**

[Release\_2023-05] Draft Thu May 25 13:03:02 2023 +0200

# Table of Contents

1. Introduction.....	2
1.1. About this document .....	2
1.2. Conventions .....	2
1.3. General pre/post conditions & tool validations.....	2
2. A Security .....	4
3. B Provisioning.....	21
4. C Authorization.....	40
5. D Local Authorization List Management.....	55
6. E Transactions .....	59
7. F Remote Control.....	88
8. G Availability.....	104
9. H Reservation.....	114
10. I Tariff and Cost.....	114
11. J MeterValues .....	117
12. K SmartCharging .....	127
13. L Firmware Management .....	154
14. M ISO IEC 15118 CertificateManagement .....	173
15. N Diagnostics.....	182
16. O Display Message .....	203
17. P DataTransfer.....	220
18. Reusable states .....	222

Copyright © 2010 - 2023 Open Charge Alliance. All rights reserved.

#### Copyright and Restrictions on Circulation and Use

This OCPP 2.0.1 Test Cases Draft document (set) is protected by copyright held by the Open Charge Alliance, and is made available only for the purpose of reviewing this draft by OCA participants / members during the review period, to make comments/proposals for the purpose of improving the specification and by users of the OCTT for the purpose of running the OCTT 2.0.1 software with their implementation(s) / system.

All other rights are reserved; in particular any party receiving this document (set) is explicitly prohibited from distributing it in any form.

Following the release of any subsequent version (whether draft or final) of the OCPP 2.0.1 (Part 6) Test cases specification set by OCA, receiving parties are prohibited from using the content of this draft for any purpose, including particularly as a basis for protocol or test tool implementations, unless specific written permission by OCA is given or a separate license for this is acquired.

#### Version History

Version	Reviewed by	Modified by	Description
1.0 draft	N/a	Open Charge Alliance	Draft version

# 1. Introduction

## 1.1. About this document

This document is created to describe a set of valid test cases for OCPP 2.0.1. These test cases can be executed using the OCPP Compliance Testing Tool (OCTT) for OCPP 2.0.1. The scenarios in the tool are described in detail including the expected behaviour of the System Under Test (SUT). This document is divided in chapters, each describing an OCPP functional block as can be found in the official OCPP specification. These are:

- A. Security
- B. Provisioning
- C. Authorization
- D. Local Authorization List Management
- E. Transactions
- F. Remote Control
- G. Availability
- H. Reservation
- I. Tariff and Cost
- J. Meter Values
- K. Smart Charging
- L. Firmware Management
- M. ISO 15118 Certificate Management
- N. Diagnostics
- O. Display Message
- P. Data Transfer

The scenarios in this document are also part of the OCA certification process of OCPP. Please refer to OCPP 2.0.1 Part 5 - Certification Profiles for more information about the relation between certification profiles and the test scenarios in this document.

## 1.2. Conventions

The following conventions / rules apply to all test cases, unless explicitly mentioned otherwise. These will not be mentioned separately at every test case.

- The OCPP specification is always leading.
- This document does not specify which tests need to be passed for certification, this will be specified in a separate document.
- All messages shall comply with the OCPP 2.0.1 schemas from the OCPP specification.
- The messages are to be sent as mentioned in the scenario details.
- Validations will be mentioned and grouped per step.
- Messages, datatypes and configuration variables will convey to the following formatting rules:
  - Datatypes, messages and configuration variables are displayed bold.
  - Values are displayed italic.

## 1.3. General pre/post conditions & tool validations

General conditions/validations are overruled by testcase specific conditions/validations, unless specifically stated otherwise.

### General pre conditions:

The following pre conditions apply to all test cases, unless explicitly mentioned otherwise.

- The Configuration variable **TxCtrlr.TxStartPoint** is *"EVConnected,Authorized"*

- 
- The Configuration variable **TxCtrlr.TxStopPoint** is *"EVConnected"*
  - The Configuration variable **AuthCtrlr.AuthEnabled** is *true*
  - The Configuration variable **AuthCtrlr.AuthorizeRemoteStart** is *false*
  - The Configuration variable **AdditionalRootCertificateCheck** is *false*
  - The Configuration variable **AllowNewSessionsPendingFirmwareUpdate** is *false*
  - The Configuration variable **AlignedDataSendDuringIdle** is *false*
  - The Configuration variable **CentralContractValidationAllowed** is *true*

**General tool rules/validations:**

- The list of ChargingSchedulePeriod elements in a chargingSchedule SHALL be ordered by increasing values of ChargingSchedulePeriod.startPeriod. This means the list is in chronological order.
- The CSMS SHALL NOT set phaseToUse in a SetChargingProfileRequest when numberPhases is other than 1.

DRAFT

## 2. A Security

Table 1. Test Case Id: TC\_A\_01\_CSMS

Test case name	Basic Authentication - Valid username/password combination	
Test case Id	TC_A_01_CSMS	
Use case Id(s)	A00, B01	
Requirement(s)	A00.FR.204, B01.FR.02	
System under test	CSMS	
Description	The Charging Station uses Basic authentication to authenticate itself to the CSMS, when using security profile 1 or 2.	
Purpose	To verify whether the CSMS is able to validate the (valid) Basic authentication credentials provided by the Charging Station at the connection request.	
Prerequisite(s)	The CSMS supports security profile 1 and/or 2	
Before (Preparations)	<b>Configuration State:</b> The CSMS must have a password configured that equals the configured BasicAuthPassword at the OCTT.	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a HTTP upgrade request with an Authorization header, containing a username/password combination.  <u>Note(s):</u> - The Authorization header is formatted as follows: <i>AUTHORIZATION: Basic &lt;Base64 encoded(&lt;Configured ChargingStationId&gt;:&lt;Configured BasicAuthPassword&gt;)&gt;</i>	2. The CSMS validates the username/password combination AND upgrades the connection to a (secured) WebSocket connection.
	3. The OCTT sends a <b>BootNotificationRequest</b>	4. The CSMS responds with a <b>BootNotificationResponse</b>
	5. The OCTT notifies the CSMS about the current state of all connectors.	6. The CSMS responds accordingly.
Tool validations	* Step 4: Message: <b>BootNotificationResponse</b> - <b>status</b> must be <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 2. Test Case Id: TC\_A\_02\_CSMS

<b>Test case name</b>	<b>Basic Authentication - Username does not equal ChargingStationId</b>	
<b>Test case Id</b>	TC_A_02_CSMS	
<b>Use case Id(s)</b>	A00	
<b>Requirement(s)</b>	A00.FR.204	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station uses Basic authentication to authenticate itself to the CSMS, when using security profile 1 or 2.	
<b>Purpose</b>	To verify whether the CSMS is able to validate the (invalid) Basic authentication credentials provided by the Charging Station at the connection request.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 1 and/or 2	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. The OCTT sends a HTTP upgrade request with an Authorization header, containing a username/password combination.</p> <p><u>Note(s):</u> - The Authorization header is formatted as follows: AUTHORIZATION: Basic &lt;Base64 encoded(&lt;Configured ChargingStationId&gt; + Invalid:&lt;Configured basicAuthPassword&gt;)&gt;</p>	<p>2. The CSMS validates the username/password combination AND rejects the connection upgrade request.</p>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 3. Test Case Id: TC\_A\_03\_CSMS

<b>Test case name</b>	<b>Basic Authentication - Invalid password</b>	
<b>Test case Id</b>	TC_A_03_CSMS	
<b>Use case Id(s)</b>	A00	
<b>Requirement(s)</b>	A00.FR.204	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station uses Basic authentication to authenticate itself to the CSMS, when using security profile 1 or 2.	
<b>Purpose</b>	To verify whether the CSMS is able to validate the (invalid) Basic authentication credentials provided by the Charging Station at the connection request.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 1 and/or 2	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. The OCTT sends a HTTP upgrade request with an Authorization header, containing a username/password combination.</p> <p><u>Note(s):</u></p> <p>- The Authorization header is formatted as follows:  <b>AUTHORIZATION: Basic</b> &lt;Base64 encoded(&lt;Configured ChargingStationId&gt;:&lt;randomly chosen identifierString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by identifierString)&gt;)&gt;</p>	<p>2. The CSMS validates the username/password combination AND rejects the connection upgrade request.</p>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	



Table 4. Test Case Id: TC\_A\_04\_CSMS

<b>Test case name</b>	<b>TLS - server-side certificate - Valid certificate</b>
<b>Test case Id</b>	TC_A_04_CSMS
<b>Use case Id(s)</b>	A00
<b>Requirement(s)</b>	A00.FR.306,A00.FR.307,A00.FR.312,A00.FR.318,A00.FR.321,A00.FR.502,A00.FR.503,A00.FR.507,A00.FR.508,A00.FR.510
<b>System under test</b>	CSMS
<b>Description</b>	The CSMS uses a server-side certificate to identify itself to the Charging Station, when using security profile 2 or 3.
<b>Purpose</b>	To verify whether the CSMS is able to provide a valid server certificate and setup a secured WebSocket connection.
<b>Prerequisite(s)</b>	The CSMS supports security profile 2 and/or 3
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a
	<b>Memory State:</b> N/a
	<b>Reusable State(s):</b> N/a

Test case name	TLS - server-side certificate - Valid certificate	
Main (Test scenario)	Charging Station	CSMS
	1. The OCTT terminates the connection and initiates a TLS handshake and sends a Client Hello to the CSMS.	2. The CSMS responds with a Server Hello With the <Configured server certificate>
	3. The OCTT performs the following actions: Send client certificate Client Key Exchange Certificate verify Change Cipher Spec Finished  <u>Note(s):</u> - The client certificate is only sent when the CSMS uses security profile 3.	4. The CSMS performs the following actions: Change Cipher Spec Finished
	5. The OCTT sends a HTTP upgrade request to the CSMS  <u>Note(s):</u> - The HTTP request only contains a username/password combination when the CSMS uses security profile 2.	6. The CSMS upgrades the connection to a (secured) WebSocket connection.
	7. The OCTT sends a <b>BootNotificationRequest</b> with <b>reason</b> <i>PowerUp</i> <b>chargingStation.model</b> <Configured model> <b>chargingStation.vendorName</b> <Configured vendorName>	8. The CSMS responds with a <b>BootNotificationResponse</b>
	9. The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	10. The CSMS responds accordingly.

Test case name	TLS - server-side certificate - Valid certificate
Tool validations	<p>* Step 3:</p> <p>The OCTT validates the following before finishing the TLS handshake:</p> <ul style="list-style-type: none"> <li>- The CSMS must use TLS version 1.2 or above</li> </ul> <p>At least the following set of cipher suites must be supported:</p> <p>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</p> <p>AND</p> <p>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</p> <p>AND</p> <p>TLS_RSA_WITH_AES_128_GCM_SHA256</p> <p>AND</p> <p>TLS_RSA_WITH_AES_256_GCM_SHA384</p> <ul style="list-style-type: none"> <li>- When using RSA or DSA the key must be at least 2048 bits long.</li> </ul> <p>and when using elliptic curve cryptography the key must be at least 224 bits long.</p> <ul style="list-style-type: none"> <li>- The received server side certificate must be transmitted in the X.509 format encoded in Privacy-Enhanced Mail (PEM) format.</li> <li>- The certificate must include a serial number.</li> <li>- The subject field of the certificate must contain a commonName RDN which consists of the FQDN of the endpoint of the server.</li> </ul> <p><i>NOTE: If one of the above validations fails, the OCTT can still proceed with the next steps of the testcase (if it is able to), but the testcase will FAIL and the OCTT reports why it failed.</i></p> <p>* Step 8:</p> <p>Message: <b>BootNotificationResponse</b> with <b>status Accepted</b></p>
	<p><b>Post scenario validations:</b> N/a</p>

Table 5. Test Case Id: TC\_A\_06\_CSMS

<b>Test case name</b>	<b>TLS - server-side certificate - TLS version too low</b>	
<b>Test case Id</b>	TC_A_06_CSMS	
<b>Use case Id(s)</b>	A00	
<b>Requirement(s)</b>	A00.FR.314,A00.FR.315,A00.FR.409,A00.FR.416,A00.FR.417,A00.FR.418	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS uses a server-side certificate to identify itself to the Charging Station, when using security profile 2 or 3.	
<b>Purpose</b>	To verify whether the CSMS is able to terminate the connection when it notices the used TLS version is lower than 1.2.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 2 and/or 3	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT terminates the connection and initiates a TLS handshake with a TLS version lower than 1.2 and sends a Client Hello to the CSMS.	2. The CSMS notices that the TLS version is lower than 1.2 and terminates the connection.
	3. The OCTT initiates a TLS handshake with TLS version 1.2 or higher and sends a Client Hello to the CSMS.	4. The CSMS responds with a Server Hello With the <Configured server certificate>
	5. The OCTT performs the following actions: Send client certificate Client Key Exchange Certificate verify Change Cipher Spec Finished  <u>Note(s):</u> - The client certificate is only sent when the CSMS uses security profile 3.	6. The CSMS performs the following actions: Change Cipher Spec Finished
	7. The OCTT sends a HTTP upgrade request to the CSMS  <u>Note(s):</u> - The HTTP request only contains a username/password combination when the CSMS uses security profile 2.	8. The CSMS upgrades the connection to a (secured) WebSocket connection.
	9. The OCTT sends a <b>BootNotificationRequest</b> with <b>reason</b> PowerUp <b>chargingStation.model</b> <Configured model> <b>chargingStation.vendorName</b> <Configured vendorName>	10. The CSMS responds with a <b>BootNotificationResponse</b>

Test case name	TLS - server-side certificate - TLS version too low	
	<p><b>11.</b> The OCTT notifies the CSMS about the current state of all connectors.</p> <p>Message: <b>StatusNotificationRequest</b></p> <ul style="list-style-type: none"> <li>- <b>connectorStatus</b> <i>Available</i></li> </ul> <p>Message: <b>NotifyEventRequest</b></p> <ul style="list-style-type: none"> <li>- <b>trigger</b> <i>Delta</i></li> <li>- <b>actualValue</b> <i>"Available"</i></li> <li>- <b>component.name</b> <i>"Connector"</i></li> <li>- <b>variable.name</b> <i>"AvailabilityState"</i></li> </ul>	<p><b>12.</b> The CSMS responds accordingly.</p>
	<p><b>13</b> The OCTT sends a <b>SecurityEventNotificationRequest</b> With <b>type</b> <i>InvalidTLSVersion</i></p>	<p><b>14</b> The CSMS responds with a <b>SecurityEventNotificationResponse</b></p>
Tool validations	<p>* Step 10:</p> <p>Message: <b>BootNotificationResponse</b></p> <ul style="list-style-type: none"> <li>- <b>status</b> <i>Accepted</i></li> </ul>	
	<p><b>Post scenario validations:</b></p> <p>N/a</p>	

Table 6. Test Case Id: TC\_A\_07\_CSMS

Test case name	TLS - Client-side certificate - valid certificate	
Test case Id	TC_A_07_CSMS	
Use case Id(s)	A00	
Requirement(s)	A00.FR.409,A00.FR.410,A00.FR.415,A00.FR.416,A00.FR.421	
System under test	CSMS	
Description	The Charging Station uses a client-side certificate to identify itself to the CSMS, when using security profile 3.	
Purpose	To verify whether the CSMS is able to receive a client certificate provided by a Charging Station and setup a secured WebSocket connection.	
Prerequisite(s)	The CSMS supports security profile 3	
Before (Preparations)	Configuration State:	
	N/a	
	Memory State:	
	N/a	
	Reusable State(s):	
	N/a	
Main (Test scenario)	Charging Station	CSMS
	1. The OCTT terminates the connection and initiates a TLS handshake and sends a Client Hello to the CSMS.	2. The CSMS responds with a Server Hello With the <Configured server certificate>
	3. The OCTT performs the following actions: Send <Configured client certificate> Client Key Exchange Certificate verify Change Cipher Spec Finished	4. The CSMS performs the following actions: Change Cipher Spec Finished
	5. The OCTT sends a HTTP upgrade request to the CSMS	6. The CSMS upgrades the connection to a (secured) WebSocket connection.
	7. The OCTT sends a <b>BootNotificationRequest</b> with <b>reason</b> <i>PowerUp</i> <b>chargingStation.model</b> <Configured model> <b>chargingStation.vendorName</b> <Configured vendorName>	8. The CSMS responds with a <b>BootNotificationResponse</b>
	9. The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	10. The CSMS responds accordingly.

Test case name	TLS - Client-side certificate - valid certificate
Tool validations	<p>* Step 3:</p> <p>The OCTT validates the following before finishing the TLS handshake:</p> <ul style="list-style-type: none"><li>- The CSMS must use TLS version 1.2 or above</li></ul> <p>At least the following set of cipher suites must be supported:</p> <p>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</p> <p>AND</p> <p>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</p> <p>AND</p> <p>TLS_RSA_WITH_AES_128_GCM_SHA256</p> <p>AND</p> <p>TLS_RSA_WITH_AES_256_GCM_SHA384</p> <p>* Step 8:</p> <p>Message: <b>BootNotificationResponse</b> with <b>status</b> <i>Accepted</i></p>
	<p><b>Post scenario validations:</b> N/a</p>

Table 7. Test Case Id: TC\_A\_08\_CSMS

<b>Test case name</b>	<b>TLS - Client-side certificate - Invalid certificate</b>	
<b>Test case Id</b>	TC_A_08_CSMS	
<b>Use case Id(s)</b>	A00	
<b>Requirement(s)</b>	A00.FR.405,A00.FR.407,A00.FR.409,A00.FR.410	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station uses a client-side certificate to identify itself to the CSMS, when using security profile 3.	
<b>Purpose</b>	To verify whether the CSMS is able to terminate the connection when the received client certificate is invalid.	
<b>Prerequisite(s)</b>	<ul style="list-style-type: none"> <li>- The CSMS supports security profile 3</li> <li>- This testcase can be executed multiple times, using different kinds of invalid certificates: Unknown certificate expired certificate certificate with commonName that does not equal the serial number of the Charging Station.</li> </ul>	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT initiates a TLS handshake and sends a Client Hello to the CSMS.	2. The CSMS responds with a Server Hello With a server certificate
	3. The OCTT performs the following actions: Send <Configured invalid client certificate> Client Key Exchange Certificate verify Change Cipher Spec Finished	4. The CSMS deems the client certificate invalid and terminates the connection.
	5. The OCTT initiates a TLS handshake and sends a Client Hello to the CSMS.	6. The CSMS responds with a Server Hello With a server certificate
	7. The OCTT performs the following actions: Send <Configured client certificate> Client Key Exchange Certificate verify Change Cipher Spec Finished	8. The CSMS performs the following actions: Change Cipher Spec Finished
	9. The OCTT sends a HTTP upgrade request to the CSMS	10. The CSMS upgrades the connection to a (secured) WebSocket connection.
	11. The OCTT sends a <b>BootNotificationRequest</b> with <b>reason</b> <i>PowerUp</i> <b>chargingStation.model</b> <Configured model> <b>chargingStation.vendorName</b> <Configured vendorName>	12. The CSMS responds with a <b>BootNotificationResponse</b>



Test case name	TLS - Client-side certificate - Invalid certificate	
	<p><b>13.</b> The OCTT notifies the CSMS about the current state of all connectors.</p> <p>Message: <b>StatusNotificationRequest</b></p> <ul style="list-style-type: none"> <li>- <b>connectorStatus</b> <i>Available</i></li> </ul> <p>Message: <b>NotifyEventRequest</b></p> <ul style="list-style-type: none"> <li>- <b>trigger</b> <i>Delta</i></li> <li>- <b>actualValue</b> <i>"Available"</i></li> <li>- <b>component.name</b> <i>"Connector"</i></li> <li>- <b>variable.name</b> <i>"AvailabilityState"</i></li> </ul>	<p><b>14.</b> The CSMS responds accordingly.</p>
Tool validations	<p>* Step 12:</p> <p>Message: <b>BootNotificationResponse</b> with <b>status</b> <i>Accepted</i></p>	
	<p><b>Post scenario validations:</b></p> <p>N/a</p>	

Table 8. Test Case Id: TC\_A\_09\_CSMS

<b>Test case name</b>	<b>Update Charging Station Password for HTTP Basic Authentication - Accepted</b>	
<b>Test case Id</b>	TC_A_09_CSMS	
<b>Use case Id(s)</b>	A01	
<b>Requirement(s)</b>	A01.FR.02, A01.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case defines how to use the BasicAuthPassword, the password used to authenticate Charging Stations in security profile 1 (Basic Authentication) and security profile 2 (TLS with Basic Authentication)	
<b>Purpose</b>	To verify if the CSMS is able to successfully set the new BasicAuthPassword and only accepts the new credentials as described at the OCPP specification.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 1 and/or 2	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetVariablesResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetVariablesRequest</b> with: <b>setVariableData[1]</b> : - <b>variable.name</b> = "BasicAuthPassword" - <b>component.name</b> = "SecurityCtrlr" - <b>attributeValue</b> = "<NewPassword>"
	3. The OCTT sends a HTTP upgrade request with an Authorization header, containing a username/password combination (with the new <i>BasicAuthPassword</i> ).  <u>Note(s)</u> : - The Authorization header is formatted as follows: <i>AUTHORIZATION: Basic &lt;Base64 encoded(&lt;Configured ChargingStationId&gt;:&lt;NEW BasicAuthPassword&gt;)&gt;</i>	4. The CSMS validates the username/password combination AND upgrades the connection to a (secured) WebSocket connection.
	5. The OCTT sends a <b>BootNotificationRequest</b>	6. The CSMS responds with a <b>BootNotificationResponse</b>
	7. The OCTT notifies the CSMS about the current state of all connectors.	8. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message: <b>SetVariableRequest</b> - <b>variable.name</b> = "BasicAuthPassword" - <b>component.name</b> = "SecurityCtrlr"	
	* Step 6: Message: <b>BootNotificationResponse</b> - <b>status</b> must be <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 9. Test Case Id: TC\_A\_10\_CSMS

<b>Test case name</b>	<b>Update Charging Station Password for HTTP Basic Authentication - Rejected</b>	
<b>Test case Id</b>	TC_A_10_CSMS	
<b>Use case Id(s)</b>	A01	
<b>Requirement(s)</b>	A01.FR.02, A01.FR.04, A01.FR.05	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case defines how to use the BasicAuthPassword, the password used to authenticate Charging Stations in security profile 1 (Basic Authentication) and security profile 2 (TLS with Basic Authentication)	
<b>Purpose</b>	To verify if the CSMS keeps accepting the old credentials and keeps communication when the new BasicAuthPassword is rejected as described at the OCPP specification.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 1 and/or 2	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetVariablesResponse</b> with <b>status Rejected</b>	1. The CSMS sends a <b>SetVariablesRequest</b> with: <b>setVariableData[1]</b> : - <b>variable.name</b> = "BasicAuthPassword" - <b>component.name</b> = "SecurityCtrlr" - <b>attributeValue</b> = "<NewPassword>"
	3. The OCTT sends a HTTP upgrade request with an Authorization header, containing a username/password combination (with the old <i>BasicAuthPassword</i> ).  <u>Note(s)</u> : - The Authorization header is formatted as follows: <i>AUTHORIZATION: Basic &lt;Base64 encoded(&lt;Configured ChargingStationId&gt;:&lt;OLD Configured BasicAuthPassword&gt;)&gt;</i>	4. The CSMS validates the username/password combination AND upgrades the connection to a (secured) WebSocket connection.
	5. The OCTT sends a <b>BootNotificationRequest</b>	6. The CSMS responds with a <b>BootNotificationResponse</b>
	7. The OCTT notifies the CSMS about the current state of all connectors.	8. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message: <b>SetVariableRequest</b> - <b>variable.name</b> = "BasicAuthPassword" - <b>component.name</b> = "SecurityCtrlr"	
	* Step 6: Message: <b>BootNotificationResponse</b> - <b>status</b> must be <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 10. Test Case Id: TC\_A\_11\_CSMS

<b>Test case name</b>	<b>Update Charging Station Certificate by request of CSMS - Success - Charging Station Certificate</b>	
<b>Test case Id</b>	TC_A_11_CSMS	
<b>Use case Id(s)</b>	A02 & F06	
<b>Requirement(s)</b>	A02.FR.11, A02.FR.14 & F06.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to update its charging station certificate using the TriggerMessageRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to update its Charging Station Certificate.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 3	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <a href="#">RenewChargingStationCertificate</a>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 11. Test Case Id: TC\_A\_14\_CSMS

<b>Test case name</b>	<b>Update Charging Station Certificate by request of CSMS - Invalid certificate</b>	
<b>Test case Id</b>	TC_A_14_CSMS	
<b>Use case Id(s)</b>	A02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to update its charging station certificate using the TriggerMessageRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station rejecting the new Charging Station certificate.	
<b>Prerequisite(s)</b>	The CSMS supports security profile 3	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3 The OCTT sends a <b>SignCertificateRequest</b> With <b>csr</b> <i>&lt;Configured CSR&gt;</i> <b>certificateType</b> <i>ChargingStationCertificate</i>	4. The CSMS responds with a <b>SignCertificateResponse</b>
	6. The OCTT responds with a <b>CertificateSignedResponse</b> With <b>status</b> <i>Rejected</i>	5. The CSMS sends a <b>CertificateSignedRequest</b>
<b>Tool validations</b>	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> <i>SignChargingStationCertificate</i> * Step 4: Message: <b>SignCertificateResponse</b> - <b>status</b> <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 12. Test Case Id: TC\_A\_19\_CSMS

<b>Test case name</b>	<b>Upgrade Charging Station Security Profile - Accepted</b>	
<b>Test case Id</b>	TC_A_19_CSMS	
<b>Use case Id(s)</b>	A05	
<b>Requirement(s)</b>	A05.FR.04, A05.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS updates the connection details on the Charging Station, to increase the security profile level.	
<b>Purpose</b>	To verify if the CSMS is able to set a new network connection profile at one of the by the Charging Station defined configuration slots with a higher security profile than currently configured.	
<b>Prerequisite(s)</b>	- Security profile must be set to 1 or 2. - If Security profile is set to 1, then a trusted certificate must be installed.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> If configured <Security profile> is 2, then <a href="#">RenewChargingStationCertificate</a>	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to set a new NetworkConnectionProfile with a security profile level one higher than currently configured	
	2. The OCTT responds with a <b>SetNetworkProfileResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>SetNetworkProfileRequest</b>
	<u>Manual Action:</u> Request the CSMS to change the NetworkConfigurationPriority to one that contains the configurationSlot of the new NetworkConnectionProfile from step 1	
	4. The OCTT responds with a <b>SetVariablesResponse</b> with <b>status Accepted</b>	3. The CSMS sends a <b>SetVariablesRequest</b>
	<u>Manual Action:</u> Request the CSMS to reboot the Charging Station	
	6. The OCTT responds with a <b>ResetResponse</b> with <b>status Accepted</b>	5. The CSMS sends a <b>ResetRequest</b>
	7. The OCTT reconnects to the CSMS with security profile is <Configured securityProfile + 1>	8. The CSMS accepts the connection attempt.
	9. Execute <b>Reusable State Booted</b>	
	10. The OCTT reconnects to the CSMS with security profile is <Configured securityProfile>	11. The CSMS shall not accept the connection attempt.
<b>Tool validations</b>	* Step 1: Message <b>SetNetworkProfileRequest</b> - <b>connectionData.messageTimeout</b> <Configured messageTimeout> - <b>connectionData.ocppCsmsUrl</b> <Configured ocppCsmsUrl> - <b>connectionData.ocppInterface</b> <Configured ocppInterface> - <b>connectionData.ocppTransport</b> JSON - <b>connectionData.ocppVersion</b> OCPP20 - <b>connectionData.securityProfile</b> <Configured securityProfile + 1> * Step 3: Message <b>SetVariablesRequest</b> <b>setVariableData:</b> - <b>variable.name</b> = "NetworkConfigurationPriority" - <b>component.name</b> = "OCPPCommCtrlr" - <b>attributeValue</b> = <contains configurationSlot provided at step 1>	
	<b>Post scenario validations:</b> - N/a	

### 3. B Provisioning

Table 13. Test Case Id: TC\_B\_01\_CSMS

Test case name	Cold Boot Charge Point - Accepted	
Test case Id	TC_B_01_CSMS	
Use case Id(s)	B01	
Requirement(s)	B01.FR.02	
System under test	CSMS	
Description	The booting mechanism allows a Charging Station to provide some general information about the Charging Station to the CSMS on startup AND it allows the Charging Station to request whether it is allowed to start sending other OCPP messages.	
Purpose	To verify whether the CSMS is able to accept the communications of a registered Charging Station.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>Booted</i>	
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 14. Test Case Id: TC\_B\_06\_CSMS

Test case name	Get Variables – single value	
Test case Id	TC_B_06_CSMS	
Use case Id(s)	B06	
Requirement(s)	B06.FR.01, B06.FR.02, B06.FR.03, B06.FR.04, B06.FR.10, B06.FR.11	
System under test	CSMS	
Description	Get the value of two of the required variables of OCPPCommCtrlr	
Purpose	To test getting single value using GetVariablesRequest for one of the mandatory component/variable combinations that must exist in the DM implementation.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>GetVariablesResponse</b>	1. Manually request CSMS to get data for: - OCPPCommCtrlr.OfflineThreshold
Tool validations	* Step 1: Message: <b>GetVariablesRequest</b> with (in arbitrary order) <b>getVariableData[0]:</b> - <b>attributeType</b> is at least absent or <b>attributeType</b> = <i>Actual</i> , but <i>Target</i> , <i>MinSet</i> , and <i>MaxSet</i> are also allowed - <b>variable.name</b> = "OfflineThreshold" - <b>component.name</b> = "OCPPCommCtrlr"	
	<b>Post scenario validations:</b> Manually validate that CSMS has correctly read the requested variables.	



Table 15. Test Case Id: TC\_B\_07\_CSMS

Test case name	Get Variables – multiple values	
Test case Id	TC_B_07_CSMS	
Use case Id(s)	B06	
Requirement(s)	B06.FR.01, B06.FR.02, B06.FR.03	
System under test	CSMS	
Description	Get the value of two of the required variables of OCPPCommCtrlr	
Purpose	To test getting multiple values using GetVariablesRequest for one of the mandatory component/variable combinations that must exist in the DM implementation.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>GetVariablesResponse</b>	1. Manually request CSMS to get data for: - OCPPCommCtrlr.OfflineThreshold - AuthCtrlr.AuthorizeRemoteStart
Tool validations	* Step 1: Message: <b>GetVariablesRequest</b> with (in arbitrary order) <b>getVariableData[0]:</b> - <b>attributeType</b> is at least absent or <b>attributeType</b> = <i>Actual</i> , but <i>Target</i> , <i>MinSet</i> , and <i>MaxSet</i> are also allowed - <b>variable.name</b> = "OfflineThreshold" - <b>component.name</b> = "OCPPCommCtrlr" <b>getVariableData[1]:</b> - <b>attributeType</b> is at least absent or <b>attributeType</b> = <i>Actual</i> , but <i>Target</i> , <i>MinSet</i> , and <i>MaxSet</i> are also allowed - <b>variable.name</b> = "AuthorizeRemoteStart" - <b>component.name</b> = "AuthCtrlr"	
	<b>Post scenario validations:</b> Manually validate that CSMS has correctly read the requested variables.	

Table 16. Test Case Id: TC\_B\_09\_CSMS

Test case name	Set Variables - single value	
Test case Id	TC_B_09_CSMS	
Use case Id(s)	B05	
Requirement(s)	B05.FR.01, B05.FR.02, B05.FR.03, B05.FR.10, B05.FR.12	
System under test	CSMS	
Description	Set the value of one of the required variables of OCPPCommCtrlr	
Purpose	To test setting a single value using SetVariablesRequest for one of the mandatory component/variable combinations that must exist in the DM implementation.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>SetVariablesResponse</b>	1. <i>Manually request CSMS to set data for:</i> - OCPPCommCtrlr.OfflineThreshold
Tool validations	* Step 1: Message: <b>SetVariablesRequest</b> with (in arbitraty order): <b>setVariableData[1]:</b> - <b>variable.name</b> = "OfflineThreshold" - <b>component.name</b> = "OCPPCommCtrlr" - <b>attributeValue</b> = "123" - <b>attributeType</b> is absent or <b>attributeType</b> = <i>Actual</i>	
	<b>Post scenario validations:</b> Manually validate that CSMS has correctly set the requested variables.	

Table 17. Test Case Id: TC\_B\_10\_CSMS

Test case name	Set Variables - multiple values	
Test case Id	TC_B_10_CSMS	
Use case Id(s)	B05	
Requirement(s)	B05.FR.01, B05.FR.02, B05.FR.03	
System under test	CSMS	
Description	Set the value of two of the required variables of OCPPCommCtrlr	
Purpose	To test setting multiple values using SetVariablesRequest for one of the mandatory component/variable combinations that must exist in the DM implementation.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>SetVariablesResponse</b>	1. Manually request CSMS to set data for: - OCPPCommCtrlr.OfflineThreshold - AuthCtrlr.AuthorizeRemoteStart+
Tool validations	* Step 1: Message: <b>SetVariablesRequest</b> with (in arbitraty order): <b>setVariableData[1]:</b> - <b>variable.name</b> = "OfflineThreshold" - <b>component.name</b> = "OCPPCommCtrlr" - <b>attributeValue</b> = "123" - <b>attributeType</b> is absent or <b>attributeType</b> = Actual <b>setVariableData[2]:</b> - <b>variable.name</b> = "AuthorizeRemoteStart" - <b>component.name</b> = "AuthCtrlr" - <b>attributeValue</b> = "false" - <b>attributeType</b> is absent or <b>attributeType</b> = Actual	
	<b>Post scenario validations:</b> Manually validate that CSMS has correctly set the requested variables.	

Table 18. Test Case Id: TC\_B\_12\_CSMS

<b>Test case name</b>	<b>Get Base Report - ConfigurationInventory</b>	
<b>Test case Id</b>	TC_B_12_CSMS	
<b>Use case Id(s)</b>	B07	
<b>Requirement(s)</b>	B07.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a ConfigurationInventory base report.	
<b>Purpose</b>	To test that CSMS supports the ConfigurationInventory base report.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>GetBaseReportResponse</b>	1. Manually instruct CSMS to retrieve a ConfigurationInventory report.
<b>Tool validations</b>	* Step 1: Message: <b>GetBaseReportRequest</b> with: - <b>requestId</b> has integer value $\geq 0$ - <b>reportBase</b> = ConfigurationInventory	
	<b>Post scenario validations:</b> CSMS receives all <b>NotifyReportRequest</b> message for this <i>requestId</i> and is able to show the result of configuration inventory to an operator.	

Table 19. Test Case Id: TC\_B\_13\_CSMS

<b>Test case name</b>	<b>Get Base Report - FullInventory</b>	
<b>Test case Id</b>	TC_B_13_CSMS	
<b>Use case Id(s)</b>	B07	
<b>Requirement(s)</b>	B07.FR.08	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a FullInventory base report.	
<b>Purpose</b>	To test that CSMS supports the FullInventory base report.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>GetBaseReportResponse</b>	1. Manually instruct CSMS to retrieve a FullInventory report.
<b>Tool validations</b>	* Step 1: <b>GetBaseReportRequest</b> with: - <b>requestId</b> has integer value $\geq 0$ - <b>reportBase</b> = <i>FullInventory</i>	
	<b>Post scenario validations:</b> CSMS receives all <b>NotifyReportRequest</b> message for this <i>requestId</i> and is able to show the result of full inventory to an operator.	

Table 20. Test Case Id: TC\_B\_14\_CSMS

<b>Test case name</b>	<b>Get Base Report - SummaryInventory</b>	
<b>Test case Id</b>	TC_B_14_CSMS	
<b>Use case Id(s)</b>	B07	
<b>Requirement(s)</b>	B07.FR.09	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a SummaryInventory base report.	
<b>Purpose</b>	To test that CSMS supports the SummaryInventory base report.	
<b>Prerequisite(s)</b>	CSMS implementation supports the optional SummaryInventory report	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>GetBaseReportResponse</b>	1. Manually instruct CSMS to retrieve a SummaryInventory report.
<b>Tool validations</b>	* Step 1: <b>GetBaseReportRequest</b> with: - <b>requestId</b> has integer value $\geq 0$ - <b>reportBase</b> = <i>SummaryInventory</i>	
	<b>Post scenario validations:</b> CSMS receives all <b>NotifyReportRequest</b> message for this <i>requestId</i> and is able to show the result of summary inventory to an operator.	

Table 21. Test Case Id: TC\_B\_18\_CSMS

<b>Test case name</b>	<b>Get Custom Report - with componentCriteria and component/variables</b>	
<b>Test case Id</b>	TC_B_18_CSMS	
<b>Use case Id(s)</b>	B08	
<b>Requirement(s)</b>	B08.FR.01, B08.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a report of components that match both the component criteria and the given list of components and variables.	
<b>Purpose</b>	To test that CSMS supports requesting a report for both the component criteria and a given list of components and optionally with variables and that it handles an empty result set.	
<b>Prerequisite(s)</b>		
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>GetReportResponse</b> with <b>status</b> <i>EmptyResultSet</i>	1. <i>Manually instruct CSMS to get the value of:</i> - EVSE #1::AvailabilityState - from all <i>Problem</i> components
	4. OCTT responds with: <b>GetReportResponse</b> with <b>status</b> <i>Accepted</i>	3. <i>Manually instruct CSMS to get the value of:</i> - EVSE #1::AvailabilityState - from all <i>Available</i> components
	5. OCTT responds with: <b>NotifyReportRequest</b>	6. CSMS sends <b>NotifyReportResponse</b>
<b>Tool validations</b>	* Step 1: Message: <b>GetReportRequest</b> - <b>componentCriteria</b> = <i>Problem</i> - <b>componentVariable[0].component.name</b> = "EVSE" - <b>componentVariable[0].component.evse.id</b> = 1 - <b>componentVariable[0].variable.name</b> = "AvailabilityState"	
	* Step 3: Message: <b>GetReportRequest</b> - <b>componentCriteria</b> is <i>Available</i> - <b>componentVariable[0].component.name</b> = "EVSE" - <b>componentVariable[0].component.evse.id</b> = 1 - <b>componentVariable[0].variable.name</b> = "AvailabilityState"	
	<b>Post scenario validations:</b> N/A	

Table 22. Test Case Id: TC\_B\_20\_CSMS

<b>Test case name</b>	<b>Reset Charging Station - Without ongoing transaction</b>	
<b>Test case Id</b>	TC_B_20_CSMS	
<b>Use case Id(s)</b>	B11	
<b>Requirement(s)</b>	B11.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the CSMS can request the Charging Station to reset itself by sending a ResetRequest without any ongoing transaction. This could for example be necessary if the Charging Station is not functioning correctly.	
<b>Purpose</b>	To verify if the CSMS is able to perform the reset mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to reboot the Charging Station with <b>type _OnIdle</b>	
	<b>2. The OCTT responds with a <b>ResetResponse</b> with <b>status Accepted</b></b>	<b>1. The CSMS sends a <b>ResetRequest</b></b>
	<b>3. The OCTT sends a <b>BootNotificationRequest</b></b>	<b>4. The CSMS responds with a <b>BootNotificationResponse</b></b>
	<b>5. The OCTT notifies the CSMS about the current state of all connectors. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus Available</b> Message: <b>NotifyEventRequest</b> - <b>trigger Delta</b> - <b>actualValue "Available"</b> - <b>component.name "Connector"</b> - <b>variable.name "AvailabilityState"</b></b>	<b>6. The CSMS responds accordingly.</b>
<b>Tool validations</b>	* Step 4: Message <b>BootNotificationResponse</b> - <b>status Accepted</b>	
	<b>Post scenario validations:</b> - N/a	



Table 23. Test Case Id: TC\_B\_21\_CSMS

<b>Test case name</b>	<b>Reset Charging Station - With Ongoing Transaction - OnIdle</b>	
<b>Test case Id</b>	TC_B_21_CSMS	
<b>Use case Id(s)</b>	B12	
<b>Requirement(s)</b>	B12.FR.01, B12.FR.03, E07.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	<p>This test case covers how the CSMS can remotely request the Charging Station to reset itself by sending a ResetRequest during a transaction. When ResetRequest "OnIdle" is send the charging stations schedules a reboot after all transactions are stopped.</p> <p>This could for example be necessary if the Charging Station is not functioning correctly.</p>	
<b>Purpose</b>	To verify if the CSMS is able to perform the reset mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to reboot the Charging Station with status OnIdle	
	2. The OCTT responds with a <b>ResetResponse</b> with <b>status Scheduled</b>	1. The CSMS sends a <b>ResetRequest</b> with <b>status OnIdle</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> . - <b>eventType</b> <i>Updated</i> - <b>triggerReason</b> <i>StopAuthorized</i> - <b>transactionInfo.chargingState</b> <i>EVConnected</i> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i>	4. The CSMS responds with a <b>TransactionEventResponse</b> .
	5. The OCTT sends a <b>TransactionEventRequest</b> . - <b>eventType</b> <i>Ended</i> - <b>triggerReason</b> <i>EVCommunicationLost</i> - <b>transactionInfo.chargingState</b> <i>Idle</i> - <b>transactionInfo.stoppedReason</b> <i>EVDisconnected</i>	6. The CSMS responds with a <b>TransactionEventResponse</b> .
	7. The OCTT sends a <b>BootNotificationRequest</b> with <b>reason ScheduledReset</b>	8. The CSMS responds with a <b>BootNotificationResponse</b>
	9. The OCTT notifies the CSMS about the current state of all connectors. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	10. The CSMS responds accordingly.

Test case name	Reset Charging Station - With Ongoing Transaction - OnIdle
Tool validations	* Step 1: Message <b>ResetRequest</b> - <b>type</b> <i>OnIdle</i> * Step 8: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>
	<b>Post scenario validations:</b> - N/a

Table 24. Test Case Id: TC\_B\_22\_CSMS

<b>Test case name</b>	<b>Reset Charging Station - With Ongoing Transaction - Immediate</b>	
<b>Test case Id</b>	TC_B_22_CSMS	
<b>Use case Id(s)</b>	B12	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	<p>This test case covers how the CSMS can remotely request the Charging Station to reset itself by sending a ResetRequest during a transaction. When ResetRequest "Immediate" is send the charging stations will try to stop all transactions before rebooting.</p> <p>This could for example be necessary if the Charging Station is not functioning correctly.</p>	
<b>Purpose</b>	To verify if the CSMS is able to perform the reset mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to reboot the Charging Station with status Immediate	
	2. The OCTT responds with a <b>ResetResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ResetRequest</b> with <b>status Immediate</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> . - <b>eventType</b> <i>Ended</i> - <b>triggerReason</b> <i>ResetCommand</i> - <b>transactionInfo.chargingState</b> <i>EVConnected</i> - <b>transactionInfo.stoppedReason</b> <i>ImmediateReset</i> - <b>idToken</b> is omitted	4. The CSMS responds with a <b>TransactionEventResponse</b> .
	5. The OCTT sends a <b>BootNotificationRequest</b> with <b>reason RemoteReset</b>	6. The CSMS responds with a <b>BootNotificationResponse</b>
	7. The OCTT notifies the CSMS about the current state of all connectors. For <Configured connectorId>: Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Occupied</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Occupied"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i> For <Other connector(s)>: Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	8. The CSMS responds accordingly.

Test case name	Reset Charging Station - With Ongoing Transaction - Immediate
Tool validations	<p>* Step 1: Message <b>ResetRequest</b> - <b>type</b> <i>Immediate</i></p> <p>* Step 6: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i></p> <p><b>Post scenario validations:</b> - N/a</p>

Table 25. Test Case Id: TC\_B\_25\_CSMS

<b>Test case name</b>	<b>Reset EVSE - Without ongoing transaction</b>	
<b>Test case Id</b>	TC_B_25_CSMS	
<b>Use case Id(s)</b>	B11	
<b>Requirement(s)</b>	B11.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the CSMS can request the Charging Station to reset an EVSE by sending a ResetRequest without any ongoing transaction. This could for example be necessary if the Charging Station is not functioning correctly.	
<b>Purpose</b>	To verify if the CSMS is able to perform the reset mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to reboot an EVSE with status OnIdle	
	2. The OCTT responds with a <b>ResetResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ResetRequest</b> with <b>status OnIdle</b> and <b>evseID &lt;Configured evseID&gt;</b>
<b>Tool validations</b>	* Step 1: Message <b>ResetRequest</b> - <b>type OnIdle</b> - <b>evseID &lt;Configured evseID&gt;</b>	
	<b>Post scenario validations:</b> - N/a	

Table 26. Test Case Id: TC\_B\_26\_CSMS

<b>Test case name</b>	<b>Reset EVSE - With Ongoing Transaction - OnIdle</b>	
<b>Test case Id</b>	TC_B_26_CSMS	
<b>Use case Id(s)</b>	B12	
<b>Requirement(s)</b>	B12.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	<p>This test case covers how the CSMS can remotely request the Charging Station to reset an EVSE by sending a ResetRequest during a transaction. When ResetRequest "OnIdle" is send the charging stations schedules a reboot after all transactions are stopped.</p> <p>This could for example be necessary if the Charging Station is not functioning correctly.</p>	
<b>Purpose</b>	To verify if the CSMS is able to perform the reset mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to reboot the charging EVSE with status OnIdle	
	2. The OCTT responds with a <b>ResetResponse</b> with <b>status Scheduled</b>	1. The CSMS sends a <b>ResetRequest</b> with <b>status OnIdle</b> and <b>evseld</b> <Configured evseld>
	3. The OCTT sends a <b>TransactionEventRequest</b> . - <b>eventType</b> <i>Updated</i> - <b>triggerReason</b> <i>StopAuthorized</i> - <b>transactionInfo.chargingState</b> <i>EVConnected</i> - <b>idToken.idToken</b> <Configured <i>valid_idtoken_idtoken</i> > - <b>idToken.type</b> <Configured <i>valid_idtoken_type</i> >	4. The CSMS responds with a <b>TransactionEventResponse</b> .
	5. The OCTT sends a <b>TransactionEventRequest</b> . - <b>eventType</b> <i>Ended</i> - <b>triggerReason</b> <i>EVCommunicationLost</i> - <b>transactionInfo.chargingState</b> <i>Idle</i> - <b>transactionInfo.stoppedReason</b> <i>EVDisconnected</i>	6. The CSMS responds with a <b>TransactionEventResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>ResetRequest</b> - <b>type</b> <i>OnIdle</i> - <b>evseld</b> <Configured evseld>	
	<b>Post scenario validations:</b> - N/a	

Table 27. Test Case Id: TC\_B\_27\_CSMS

<b>Test case name</b>	<b>Reset EVSE - With Ongoing Transaction - Immediate</b>	
<b>Test case Id</b>	TC_B_27_CSMS	
<b>Use case Id(s)</b>	B12	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	<p>This test case covers how the CSMS can remotely request the Charging Station to reset an EVSE by sending a ResetRequest during a transaction. When ResetRequest "Immediate" is send the charging stations will try to stop all transactions before rebooting.</p> <p>This could for example be necessary if the Charging Station is not functioning correctly.</p>	
<b>Purpose</b>	To verify if the CSMS is able to perform the reset mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to reboot the charging EVSE with status Immediate	
	<b>2. The OCTT responds with a ResetResponse</b> with <b>status</b> Accepted	<b>1. The CSMS sends a ResetRequest</b> with <b>status</b> Immediate and <b>evseld</b> <Configured evseld>
	<b>3. The OCTT sends a TransactionEventRequest.</b> - <b>eventType</b> Ended - <b>triggerReason</b> ResetCommand - <b>transactionInfo.chargingState</b> EVConnected - <b>transactionInfo.stoppedReason</b> ImmediateReset	<b>4. The CSMS responds with a TransactionEventResponse.</b>
<b>Tool validations</b>	* Step 1: Message <b>ResetRequest</b> - <b>type</b> Immediate - <b>evseld</b> <Configured evseld>	
	<b>Post scenario validations:</b> N/a	

Table 28. Test Case Id: TC\_B\_42\_CSMS

<b>Test case name</b>	<b>Set new NetworkConnectionProfile - Accepted</b>	
<b>Test case Id</b>	TC_B_42_CSMS	
<b>Use case Id(s)</b>	B09	
<b>Requirement(s)</b>	B09.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS updates the connection details on the Charging Station. For instance in preparation of a migration to a new CSMS.	
<b>Purpose</b>	To verify if the CSMS is able to set a new network connection profile at one of the by the Charging Station defined configuration slots.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetNetworkProfileResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>SetNetworkProfileRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetNetworkProfileRequest</b> - <b>configurationSlot</b> is <Configured configurationSlot> - <b>connectionData.messageTimeout</b> <Configured messageTimeout> - <b>connectionData.ocppCsmsUrl</b> <Configured ocppCsmsUrl> - <b>connectionData.ocppInterface</b> <Configured ocppInterface> - <b>connectionData.ocppTransport</b> JSON - <b>connectionData.ocppVersion</b> OCPP20 - <b>connectionData.securityProfile</b> <Configured securityProfile>	
	<b>Post scenario validations:</b> - N/a	



Table 29. Test Case Id: TC\_B\_44\_CSMS

<b>Test case name</b>	<b>Set new NetworkConnectionProfile - Failed</b>	
<b>Test case Id</b>	TC_B_44_CSMS	
<b>Use case Id(s)</b>	B09	
<b>Requirement(s)</b>	B09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS updates the connection details on the Charging Station. For instance in preparation of a migration to a new CSMS.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station responding with status Failed, when setting a new network connection profile at one of the by the Charging Station defined configuration slots.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetNetworkProfileResponse</b> With <b>status Failed</b>	1. The CSMS sends a <b>SetNetworkProfileRequest</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

## 4. C Authorization

Table 30. Test Case Id: TC\_C\_02\_CSMS

Test case name	Local start transaction - Authorization Invalid/Unknown	
Test case Id	TC_C_02_CSMS	
Use case Id(s)	C01, C04, C06	
Requirement(s)	C01.FR.07 OR C04.FR.01 OR C06.FR.04	
System under test	CSMS	
Description	When a Charging Station needs to charge an EV, it needs to authorize the EV Driver first at the CSMS before the charging can be started or stopped.	
Purpose	To verify whether the CSMS is able to report that an idToken is NOT valid.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured <i>invalid_idtoken_idtoken</i> > <b>idToken.type</b> <Configured <i>invalid_idtoken_type</i> >	2. The CSMS responds with an <b>AuthorizeResponse</b>
Tool validations	* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> <i>Invalid</i> or <i>Unknown</i>	
	<b>Post scenario validations:</b> - N/a	

Table 31. Test Case Id: TC\_C\_06\_CSMS

<b>Test case name</b>	<b>Local start transaction - Authorization Blocked</b>	
<b>Test case Id</b>	TC_C_06_CSMS	
<b>Use case Id(s)</b>	C01	
<b>Requirement(s)</b>	C01.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	When a Charging Station needs to charge an EV, it needs to authorize the EV Driver first at the CSMS before the charging can be started or stopped.	
<b>Purpose</b>	To verify whether the CSMS is able to report that an idToken is Blocked.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> The IdToken configured as Blocked at the OCTT, must be set as Blocked at the CSMS.	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured blocked_idtoken_idtoken> <b>idToken.type</b> <Configured blocked_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
<b>Tool validations</b>	* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> Blocked or Invalid	
	<b>Post scenario validations:</b>	

Table 32. Test Case Id: TC\_C\_07\_CSMS

<b>Test case name</b>	<b>Local start transaction - Authorization Expired</b>	
<b>Test case Id</b>	TC_C_07_CSMS	
<b>Use case Id(s)</b>	C01	
<b>Requirement(s)</b>	C01.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	When a Charging Station needs to charge an EV, it needs to authorize the EV Driver first at the CSMS before the charging can be started or stopped.	
<b>Purpose</b>	To verify whether the CSMS is able to report that an idToken is Expired.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> The IdToken configured as Expired at the OCTT, must be set as Expired at the CSMS.	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured expired_idtoken_idtoken> <b>idToken.type</b> <Configured expired_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
<b>Tool validations</b>	* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> Expired or Invalid	
	<b>Post scenario validations:</b>	

Table 33. Test Case Id: TC\_C\_08\_CSMS

<b>Test case name</b>	<b>Authorization through authorization cache - Accepted</b>	
<b>Test case Id</b>	TC_C_08_CSMS	
<b>Use case Id(s)</b>	C12	
<b>Requirement(s)</b>	C12_FR_03	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the EV Driver is authorized to start a transaction while the Charging Station uses Cached IdToken. This enables the EV Driver to Online start a transaction by using the Authorization Cache in which the Charging Station can respond faster, as no AuthorizeRequest is being sent.	
<b>Purpose</b>	To verify if the CSMS is able to respond correctly when an idToken which has status "Accepted" in the charging stations cache is presented according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> State is <i>EVConnectedPreSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1. The OCTT sends a <i>TransactionEventRequest</i> with</b> - <b>triggerReason</b> <i>Authorized</i> - <b>idToken</b> <i>&lt;Valid id token configured in Authorization Cache&gt;</i> - <b>eventType</b> <i>Updated</i>  <u>Note(s):</u> - <b>TxStartPoint</b> <i>contains ParkingBayOccupancy</i>	<b>2. The CSMS responds with a <i>TransactionEventResponse</i></b>
<b>Tool validations</b>	* Step 2: Message <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i>	
	<b>Post scenario validations:</b> - N/a	

Table 34. Test Case Id: TC\_C\_20\_CSMS

<b>Test case name</b>	<b>Authorization through authorization cache - Invalid</b>	
<b>Test case Id</b>	TC_C_20_CSMS	
<b>Use case Id(s)</b>	C12	
<b>Requirement(s)</b>	C12_FR_03	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the EV Driver is authorized to start a transaction while the Charging Station uses Cached IdToken. This enables the EV Driver to Online start a transaction by using the Authorization Cache in which the Charging Station can respond faster, as no AuthorizeRequest is being sent.	
<b>Purpose</b>	To verify if the CSMS is able to respond correctly when an idToken, which has status "Invalid" in the charging stations cache but not in the CSMS, is presented according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> State is <i>EVConnectedPreSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1. The OCTT sends a <b>TransactionEventRequest</b> with</b> - <b>triggerReason</b> <i>Authorized</i> - <b>idToken.idToken</b> <i>&lt;Configured invalid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured invalid_idtoken_type&gt;</i> - <b>eventType</b> <i>Updated</i>  <u>Note(s):</u> - <b>TxStartPoint</b> <i>contains ParkingBayOccupancy</i>	<b>2. The CSMS responds with a <b>TransactionEventResponse</b></b>
<b>Tool validations</b>	* Step 2: Message <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> <i>Invalid or Unknown</i>	
	<b>Post scenario validations:</b> - N/a	

Table 35. Test Case Id: TC\_C\_37\_CSMS

<b>Test case name</b>	<b>Clear Authorization Data in Authorization Cache - Accepted</b>	
<b>Test case Id</b>	TC_C_37_CSMS	
<b>Use case Id(s)</b>	C11	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the Charging Station autonomously stores a record of previously presented identifiers that have been successfully authorized by the CSMS in the Authorization Cache. (Successfully meaning: a response received on a message containing an IdToken)	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to clear all identifiers from the Authorization Cache according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>ClearCacheResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ClearCacheRequest</b>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	

Table 36. Test Case Id: TC\_C\_38\_CSMS

<b>Test case name</b>	<b>Clear Authorization Data in Authorization Cache - Rejected</b>	
<b>Test case Id</b>	TC_C_38_CSMS	
<b>Use case Id(s)</b>	C11	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the Charging Station autonomously stores a record of previously presented identifiers that have been successfully authorized by the CSMS in the Authorization Cache. (Successfully meaning: a response received on a message containing an IdToken)	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to clear all identifiers from the Authorization Cache according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>ClearCacheResponse</b> with <b>status Rejected</b>	1. The CSMS sends a <b>ClearCacheRequest</b>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	



Table 37. Test Case Id: TC\_C\_39\_CSMS

<b>Test case name</b>	<b>Authorization by GroupId - Success</b>	
<b>Test case Id</b>	TC_C_39_CSMS	
<b>Use case Id(s)</b>	C09	
<b>Requirement(s)</b>	C09_FR_02, C09_FR_03	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how a Charging Station can authorize an action for an EV Driver based on GroupId information. This could for example be used if 2 people regularly use the same EV: they can use their own IdToken (e.g. RFID card), and can deauthorize transactions that were started with the other idToken (with the same GroupId).	
<b>Purpose</b>	To verify if the CSMS is able to correctly handle the Authorization of idTokens with the same GroupId according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> Two valid idTokens with the same GroupId are configured	
	<b>Reusable State(s):</b> state is <i>EVConnectedPreSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured <i>valid_idtoken2_idtoken</i> > <b>idToken.type</b> <Configured <i>valid_idtoken2_type</i> >	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>Authorized</i> - <b>idToken.idToken</b> <Configured <i>valid_idtoken_idtoken</i> > - <b>idToken.type</b> <Configured <i>valid_idtoken_type</i> > if transaction was already started - <b>eventType</b> <i>Updated</i> else - <b>eventType</b> <i>Started</i>	4. The CSMS responds with a <b>TransactionEventResponse</b>
	5. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
	6. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured <i>valid_idtoken2_idtoken</i> > <b>idToken.type</b> <Configured <i>valid_idtoken2_type</i> >	7. The CSMS responds with an <b>AuthorizeResponse</b>
	8. The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>StopAuthorized</i> - <b>idToken.idToken</b> <Configured <i>valid_idtoken2_idtoken</i> > - <b>idToken.type</b> <Configured <i>valid_idtoken2_type</i> > - <b>eventType</b> <i>Updated</i>	9. The CSMS responds with a <b>TransactionEventResponse</b>
	10. Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	11. Execute <b>Reusable State</b> <i>EVDisconnected</i>	

Test case name	Authorization by GroupId - Success
Tool validations	<p>* Step 2:  Message <b>AuthorizeResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p> <p>* Step 4:  Message <b>TransactionEventResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p> <p>* Step 7:  Message <b>AuthorizeResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p> <p>* Step 9:  Message <b>TransactionEventResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p>
	<p><b>Post scenario validations:</b>  - N/a</p>

Table 38. Test Case Id: TC\_C\_40\_CSMS

<b>Test case name</b>	<b>Authorization by GroupId - Success with Local Authorization List</b>	
<b>Test case Id</b>	TC_C_40_CSMS	
<b>Use case Id(s)</b>	C09	
<b>Requirement(s)</b>	C09_FR_02, C09_FR_03	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how a Charging Station can authorize an action for an EV Driver based on GroupId information. This could for example be used if 2 people regularly use the same EV: they can use their own IdToken (e.g. RFID card), and can deauthorize transactions that were started with the other idToken (with the same GroupId).	
<b>Purpose</b>	To verify if the CSMS is able to correctly handle the Authorization of idTokens with the same GroupId which are located in the Local Authorization List according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> Two valid idTokens with same GroupId are configured	
	<b>Reusable State(s):</b> state is <i>EVConnectedPreSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>Authorized</i> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> (with a configured GroupId) which is configured in the local Authorization List - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i> (with a configured GroupId) which is configured in the local Authorization List If transaction was already started - <b>eventType</b> <i>Updated</i> else - <b>eventType</b> <i>Started</i>	<b>2.</b> The CSMS responds with a <b>TransactionEventResponse</b>
	<b>3.</b> Execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
	<b>5.</b> The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>StopAuthorized</i> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken2_idtoken&gt;</i> (with same configured GroupId) which is configured in the local Authorization List - <b>idToken.type</b> <i>&lt;Configured valid_idtoken2_type&gt;</i> - <b>eventType</b> <i>Updated</i>	<b>6.</b> The CSMS responds with a <b>TransactionEventResponse</b>
	<b>7.</b> Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	<b>8.</b> Execute <b>Reusable State</b> <i>EVDisconnected</i>	
<b>Tool validations</b>	* Step 2: Message <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i> - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i>	
	* Step 6: Message <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i> - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i>	
	<b>Post scenario validations:</b> - N/a	

Table 39. Test Case Id: TC\_C\_43\_CSMS

<b>Test case name</b>	<b>Authorization by GroupId - Invalid status with Local Authorization List</b>	
<b>Test case Id</b>	TC_C_43_CSMS	
<b>Use case Id(s)</b>	C09	
<b>Requirement(s)</b>	C09_FR_02, C09_FR_03	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how a Charging Station can authorize an action for an EV Driver based on GroupId information. This could for example be used if 2 people regularly use the same EV: they can use their own IdToken (e.g. RFID card), and can deauthorize transactions that were started with the other idToken (with the same GroupId).	
<b>Purpose</b>	To verify if the CSMS is able to correctly handle the Authorization of idTokens with the same GroupId which are located in the Local Authorization List according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> Two known valid idTokens with same GroupId are configured.	
	<b>Reusable State(s):</b> state is <i>EVConnectedPreSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>Authorized</i> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i> if transaction was already started - <b>eventType</b> <i>Updated</i> else - <b>eventType</b> <i>Started</i>	<b>2.</b> The CSMS responds with a <b>TransactionEventResponse</b>
	<b>3.</b> Execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
	<b>4.</b> The OCTT sends an <b>AuthorizeRequest</b> with - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken2_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken2_type&gt;</i>	<b>5.</b> The CSMS responds with an <b>AuthorizeResponse</b>
	<b>6.</b> The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>StopAuthorized</i> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken2_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken2_type&gt;</i> - <b>eventType</b> <i>Updated</i>	<b>7.</b> The CSMS responds with a <b>TransactionEventResponse</b>
	<b>8.</b> Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	<b>9.</b> Execute <b>Reusable State</b> <i>EVDisconnected</i>	

Test case name	Authorization by GroupId - Invalid status with Local Authorization List
Tool validations	<p>* Step 1:  Message <b>TransactionEventResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p> <p>* Step 5:  Message <b>AuthorizeResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p> <p>* Step 7:  Message <b>TransactionEventResponse</b>  - <b>idTokenInfo.status</b> <i>Accepted</i>  - <b>idTokenInfo.groupIdToken.idToken</b> <i>&lt;Configured groupIdToken&gt;</i></p>
	<p><b>Post scenario validations:</b>  - N/a</p>

Table 40. Test Case Id: TC\_C\_47\_CSMS

<b>Test case name</b>	<b>Stop Transaction with a Master Pass - With UI - All transactions</b>	
<b>Test case Id</b>	TC_C_47_CSMS	
<b>Use case Id(s)</b>	C16	
<b>Requirement(s)</b>	C16_FR_01	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how somebody with a Master Pass (User) can stop (selected) ongoing transactions, so the cable becomes unlocked. This Master Pass can be configured in: MasterPassGroupId. This could for example be usefull for Law Enforcement officials.	
<b>Purpose</b>	To verify if the CSMS is able to correctly respond on a request to stop all transactions when an idToken which has the MasterPass as GroupId is used and the user has selected to stop all transactions in the User Interface according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> An idToken with the MastersPass as GroupId is configured	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i> for EVSE 1 with idToken valid idToken State is <i>EnergyTransferStarted</i> for EVSE 2 with idToken valid idToken2	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured <i>masterpass_idtoken_idtoken</i> > <b>idToken.type</b> <Configured <i>masterpass_idtoken_type</i> >	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with - <b>transactionInfo.stoppedReason</b> <i>MasterPass</i> - <b>idToken.idToken</b> <Configured <i>valid_idtoken_idtoken</i> > - <b>idToken.type</b> <Configured <i>valid_idtoken_type</i> > - <b>eventType</b> <i>Ended</i> for both EVSE	4. The CSMS responds with a <b>TransactionEventResponse</b> for both EVSE

Table 41. Test Case Id: TC\_C\_48\_CSMS

<b>Test case name</b>	<b>Stop Transaction with a Master Pass - With UI - With UI - Specific transactions</b>	
<b>Test case Id</b>	TC_C_48_CSMS	
<b>Use case Id(s)</b>	C16	
<b>Requirement(s)</b>	C16_FR_01	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how somebody with a Master Pass (User) can stop (selected) ongoing transactions, so the cable becomes unlocked. This Master Pass can be configured in: MasterPassGroupId. This could for example be usefull for Law Enforcement officials.	
<b>Purpose</b>	To verify if the CSMS is able to correctly respond on a request to stop a transaction when an idToken which has the MasterPass as GroupId is used and the user has selected to stop one transaction in the User Interface according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> An idToken with the MastersPass as GroupId is configured	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i> for all EVSE	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with - <b>transactionInfo.stoppedReason</b> <i>MasterPass</i> - <b>idToken.idToken</b> <Configured <i>masterpass_idtoken_idtoken</i> > - <b>idToken.type</b> <Configured <i>masterpass_idtoken_type</i> > - <b>eventType</b> <i>Ended</i>	4. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 2: Message <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i> - <b>idTokenInfo.groupIdToken.idToken</b> <Configured <i>masterPassGroupId</i> >	
	* Step 4: Message <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i> - <b>idTokenInfo.groupIdToken.idToken</b> <Configured <i>masterPassGroupId</i> >	
	<b>Post scenario validations:</b> - N/a	

Table 42. Test Case Id: TC\_C\_49\_CSMS

<b>Test case name</b>	<b>Stop Transaction with a Master Pass - Without UI</b>	
<b>Test case Id</b>	TC_C_49_CSMS	
<b>Use case Id(s)</b>	C16	
<b>Requirement(s)</b>	C16_FR_02	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how somebody with a Master Pass (User) can stop (selected) ongoing transactions, so the cable becomes unlocked. This Master Pass can be configured in: MasterPassGroupId. This could for example be usefull for Law Enforcement officials.	
<b>Purpose</b>	To verify if the CSMS is able to correctly respond on a request to stop all transactions when an idToken which has the MasterPass as GroupId is used and the Charging Station does not have a User Interface according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> An idToken with the MastersPass as GroupId is configured	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i> for EVSE 1 with idToken valid idToken State is <i>EnergyTransferStarted</i> for EVSE 2 with idToken valid idToken2	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured <i>masterpass_idtoken_idtoken</i> > <b>idToken.type</b> <Configured <i>masterpass_idtoken_type</i> >	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with - <b>transactionInfo.stoppedReason</b> <i>MasterPass</i> - <b>idToken.idToken</b> <Configured <i>valid_idtoken_idtoken</i> > - <b>idToken.type</b> <Configured <i>valid_idtoken_type</i> > - <b>eventType</b> <i>Ended</i> for both EVSE	4. The CSMS responds with a <b>TransactionEventResponse</b> for both EVSE



## 5. D Local Authorization List Management

Table 43. Test Case Id: TC\_D\_02\_CSMS

Test case name	Send Local Authorization List - Differential Update	
Test case Id	TC_D_02_CSMS	
Use case Id(s)	D01	
Requirement(s)	D01_FR_01, D01_FR_06, D01_FR_18	
System under test	CSMS	
Description	The CSMS sends a Local Authorization List which a Charging Station can use for the authorization of idTokens. The list MAY be either a full list to replace the current list in the Charging Station or it MAY be a differential list with updates to be applied to the current list in the Charging Station.	
Purpose	To verify if the CSMS is able to send a Differential Local Authorization List according to the mechanism as described in the OCPP specification.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a Local Authorization list to the Charging Station with type Differential and some idTokens in the message	
	2 The OCTT responds with a <b>SendLocalListResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SendLocalListRequest</b>
	<u>Note(s):</u> If the Local Authorization List is too big for one message, step 1 and 2 will be repeated	
Tool validations	* Step 1: Message <b>SendLocalListRequest</b> - <b>updateType</b> Differential - <b>versionNumber</b> <Bigger than currently configured in OCTT> - <b>localAuthorizationList</b> <Not empty>	
	<b>Post scenario validations:</b> - N/a	

Table 44. Test Case Id: TC\_D\_04\_CSMS

<b>Test case name</b>	<b>Send Local Authorization List - Full with empty list</b>	
<b>Test case Id</b>	TC_D_04_CSMS	
<b>Use case Id(s)</b>	D01	
<b>Requirement(s)</b>	D01_FR_01, D01_FR_06, D01_FR_18	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sends a Local Authorization List which a Charging Station can use for the authorization of idTokens. The list MAY be either a full list to replace the current list in the Charging Station or it MAY be a differential list with updates to be applied to the current list in the Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to send a Full Local Authorization List without data according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a Local Authorization list to the Charging Station with type full and without AuthorizationData elements in the message	
	<b>2</b> The OCTT responds with a <b>SendLocalListResponse</b> with <b>status Accepted</b>	<b>1.</b> The CSMS sends a <b>SendLocalListRequest</b>
	<u>Note(s):</u> If the Local Authorization List is too big for one message, step 1 and 2 will be repeated	
<b>Tool validations</b>	* Step 1: Message <b>SendLocalListRequest</b> - <b>updateType Full</b> - <b>localAuthorizationList &lt;Empty&gt;</b>	
	<b>Post scenario validations:</b> - N/a	

Table 45. Test Case Id: TC\_D\_08\_CSMS

<b>Test case name</b>	<b>Get Local List Version - Success</b>	
<b>Test case Id</b>	TC_D_08_CSMS	
<b>Use case Id(s)</b>	D02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS can request a Charging Station for the version number of the Local Authorization List by sending a GetLocalListVersionRequest.	
<b>Purpose</b>	To verify if the CSMS is able to request the Local Authorization List version according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to get a Local Authorization list version	
	2 The OCTT responds with a <b>GetLocalListVersionResponse</b> with <b>versionNumber</b> <Configured versionNumber>	1. The CSMS sends a <b>GetLocalListVersionRequest</b>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	

Table 46. Test Case Id: TC\_D\_09\_CSMS

<b>Test case name</b>	<b>Get Local List Version - No list available</b>	
<b>Test case Id</b>	TC_D_09_CSMS	
<b>Use case Id(s)</b>	D02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS can request a Charging Station for the version number of the Local Authorization List by sending a GetLocalListVersionRequest.	
<b>Purpose</b>	To verify if the CSMS is able to request the Local Authorization List version according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to get a Local Authorization list version	
	2 The OCTT responds with a <b>GetLocalListVersionResponse</b> with <b>versionNumber 0</b>	1. The CSMS sends a <b>GetLocalListVersionRequest</b>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	

## 6. E Transactions

Table 47. Test Case Id: TC\_E\_01\_CSMS

Test case name	Start transaction options - PowerPathClosed	
Test case Id	TC_E_01_CSMS	
Use case Id(s)	E01(S5)	
Requirement(s)	E01.FR.05	
System under test	CSMS	
Description	Ocpp 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
Purpose	To verify if the CSMS is able to handle a Charging Station that starts a transaction when the power path has been closed.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> With <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT notifies the CSMS about the status change of the connector.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> is <i>Occupied</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> is <i>Delta</i> - <b>actualValue</b> is <i>Occupied</i> - <b>component.name</b> is <i>Connector</i> - <b>variable.name</b> is <i>AvailabilityState</i>	4. The CSMS responds accordingly.
	5. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Started</i> <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type> <b>evse.id</b> is <Configured evseld> <b>evse.connectorId</b> is <Configured connectorId> <b>transactionInfo.chargingState</b> is <i>SuspendedEVSE</i>	6. The CSMS responds with a <b>TransactionEventResponse</b>
	7. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Updated</i> <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>Charging</i>	8. The CSMS responds with a <b>TransactionEventResponse</b>

Test case name	Start transaction options - PowerPathClosed
Tool validations	<p>* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i></p> <p>* Step 6: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i></p>
	<p><b>Post scenario validations:</b> N/a</p>

DRAFT

Table 48. Test Case Id: TC\_E\_02\_CSMS

<b>Test case name</b>	<b>Start transaction options - EnergyTransfer</b>	
<b>Test case Id</b>	TC_E_02_CSMS	
<b>Use case Id(s)</b>	E01(S6)	
<b>Requirement(s)</b>	E01.FR.06	
<b>System under test</b>	CSMS	
<b>Description</b>	Ocpp 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that starts a transaction when the energy transfer starts.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> With <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT notifies the CSMS about the status change of the connector.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> is <i>Occupied</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> is <i>Delta</i> - <b>actualValue</b> is <i>Occupied</i> - <b>component.name</b> is <i>Connector</i> - <b>variable.name</b> is <i>AvailabilityState</i>	4. The CSMS responds accordingly.
	5. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Started</i> <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type> <b>evse.id</b> is <Configured evseld> <b>evse.connectorId</b> is <Configured connectorId> <b>transactionInfo.chargingState</b> is <i>Charging</i>	6. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i> * Step 6: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 49. Test Case Id: TC\_E\_03\_CSMS

<b>Test case name</b>	<b>Local start transaction - Cable plugin first</b>	
<b>Test case Id</b>	TC_E_03_CSMS	
<b>Use case Id(s)</b>	E02	
<b>Requirement(s)</b>	E02.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first connect the EV and EVSE OR present a form of identification. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that is able to start a charging session when the EV driver first connects the EV and EVSE, before authorization.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EVConnectedPreSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>Authorized</i>	
	2. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	



Table 50. Test Case Id: TC\_E\_04\_CSMS

<b>Test case name</b>	<b>Local start transaction - Authorization first - Success</b>	
<b>Test case Id</b>	TC_E_04_CSMS	
<b>Use case Id(s)</b>	E03	
<b>Requirement(s)</b>	E03.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first connect the EV and EVSE OR present a form of identification. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that is able to start a charging session when the EV driver first presents a form of identification, before connecting the EV and EVSE.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>Authorized</i>	
	2. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 51. Test Case Id: TC\_E\_39\_CSMS

<b>Test case name</b>	<b>Stop transaction options - Deauthorized - timeout</b>	
<b>Test case Id</b>	TC_E_39_CSMS	
<b>Use case Id(s)</b>	E03, E06	
<b>Requirement(s)</b>	E03.FR.04, E03.FR.05, E06.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first connect the EV and EVSE OR present a form of identification. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that deauthorizes the transaction after the <b>EVConnectionTimeout</b> has expired.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>Authorized</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1. The OCTT sends a <b>TransactionEventRequest</b></b> With <b>triggerReason</b> is <i>EVConnectTimeout</i> <b>transactionInfo.stoppedReason</b> is <i>Timeout</i> <b>eventType</b> is <i>Ended</i>  <u>Note(s):</u> - This step will be executed after the _<Configured EV connection timeout> expires._	<b>2. The CSMS responds with a <b>TransactionEventResponse</b></b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 52. Test Case Id: TC\_E\_14\_CSMS

<b>Test case name</b>	<b>Stop transaction options - EVDisconnected - Charging Station side</b>	
<b>Test case Id</b>	TC_E_14_CSMS	
<b>Use case Id(s)</b>	E06(S2)	
<b>Requirement(s)</b>	E06.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the EV and EVSE are disconnected at the Charging Station side.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EVConnectedPostSession</i>	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>EVDisconnected</i>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 53. Test Case Id: TC\_E\_20\_CSMS

<b>Test case name</b>	<b>Stop transaction options - EVDisconnected - EV side</b>	
<b>Test case Id</b>	TC_E_20_CSMS	
<b>Use case Id(s)</b>	E06(S2), E10	
<b>Requirement(s)</b>	E06.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the EV and EVSE are disconnected at the EV side.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferSuspended</i>	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>EVDisconnected</i>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 54. Test Case Id: TC\_E\_15\_CSMS

<b>Test case name</b>	<b>Stop transaction options - StopAuthorized - Local</b>	
<b>Test case Id</b>	TC_E_15_CSMS	
<b>Use case Id(s)</b>	E06(S3)	
<b>Requirement(s)</b>	E06.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the EV driver locally stops the transaction.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>StopAuthorized</i> <b>transactionInfo.stoppedReason</b> is <i>Local</i> <b>eventType</b> is <i>Ended</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 55. Test Case Id: TC\_E\_21\_CSMS

<b>Test case name</b>	<b>Stop transaction options - StopAuthorized - Remote</b>	
<b>Test case Id</b>	TC_E_21_CSMS	
<b>Use case Id(s)</b>	E06(S3) AND F03	
<b>Requirement(s)</b>	E06.FR.03,F03.FR.01,F03.FR.09, F03.FR.10	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when it receives a RequestStopTransactionRequest.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to request the Charging Station to stop the ongoing transaction.	
	2. The OCTT responds with a <b>RequestStopTransactionResponse</b> with <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>RequestStopTransactionRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> . with <b>triggerReason</b> is <i>RemoteStop</i> <b>transactionInfo.stoppedReason</b> is <i>Remote</i> <b>eventType</b> is <i>Ended</i>	4. The CSMS responds with a <b>TransactionEventResponse</b> .
<b>Tool validations</b>	* Step 1: Message: <b>RequestStopTransactionRequest</b> - <b>transactionId</b> must equal <i>&lt;transactionId provided by the OCTT in before state.&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 56. Test Case Id: TC\_E\_09\_CSMS

Test case name	Start transaction options - EVConnected	
Test case Id	TC_E_09_CSMS	
Use case Id(s)	E01(S2)	
Requirement(s)	E01.FR.02	
System under test	CSMS	
Description	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
Purpose	To verify if the CSMS is able to handle a Charging Station that starts a transaction when the EV and EVSE are connected.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. The OCTT notifies the CSMS about the status change of the connector.</p> <p>Message: <b>StatusNotificationRequest</b></p> <ul style="list-style-type: none"> <li>- <b>connectorStatus</b> is <i>Occupied</i></li> </ul> <p>Message: <b>NotifyEventRequest</b></p> <ul style="list-style-type: none"> <li>- <b>trigger</b> is <i>Delta</i></li> <li>- <b>actualValue</b> is <i>Occupied</i></li> <li>- <b>component.name</b> is <i>Connector</i></li> <li>- <b>variable.name</b> is <i>AvailabilityState</i></li> </ul>	<p>2. The CSMS responds accordingly.</p>
	<p>3. The OCTT sends a <b>TransactionEventRequest</b></p> <p>With <b>eventType</b> is <i>Started</i></p> <p><b>triggerReason</b> is <i>CablePluggedIn</i></p> <p><b>evse.id</b> is <i>&lt;Configured evseld&gt;</i></p> <p><b>evse.connectorId</b> is <i>&lt;Configured connectorId&gt;</i></p> <p><b>transactionInfo.chargingState</b> is <i>EVConnected</i></p>	<p>4. The CSMS responds with a <b>TransactionEventResponse</b></p>
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 57. Test Case Id: TC\_E\_10\_CSMS

Test case name	Start transaction options - Authorized - Local	
Test case Id	TC_E_10_CSMS	
Use case Id(s)	E01(S3)	
Requirement(s)	E01.FR.03	
System under test	CSMS	
Description	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
Purpose	To verify if the CSMS is able to handle a Charging Station that starts a transaction when the EV and EVSE are connected.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> With <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is Started <b>triggerReason</b> is Authorized <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	4. The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> must be Accepted * Step 4: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be Accepted	
	<b>Post scenario validations:</b> N/a	



Table 58. Test Case Id: TC\_E\_11\_CSMS

Test case name	Start transaction options - DataSigned	
Test case Id	TC_E_11_CSMS	
Use case Id(s)	E01(S4)	
Requirement(s)	E01.FR.04	
System under test	CSMS	
Description	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
Purpose	To verify if the CSMS is able to handle a Charging Station that starts a transaction when the signed meter values are received.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> With <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT notifies the CSMS about the status change of the connector.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> is <i>Occupied</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> is <i>Delta</i> - <b>actualValue</b> is <i>Occupied</i> - <b>component.name</b> is <i>Connector</i> - <b>variable.name</b> is <i>AvailabilityState</i>	4. The CSMS responds accordingly.
	5. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Started</i> <b>triggerReason</b> is <i>SignedDataReceived</i> <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type> <b>evse.id</b> is <Configured evseld> <b>evse.connectorId</b> is <Configured connectorId> <b>meterValue</b> is provided with the following values: <b>sampledValue.value</b> is <i>0.0</i> <b>sampledValue.context</b> is <i>Transaction.Begin</i> <b>sampledValue.signedMeterValue</b> is <Generated SignedMeterValueType>	6. The CSMS responds with a <b>TransactionEventResponse</b>
	7. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Updated</i> <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>Charging</i>	8. The CSMS responds with a <b>TransactionEventResponse</b>

Test case name	Start transaction options - DataSigned
Tool validations	<p>* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i></p> <p>* Step 6: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i></p> <p><b>Post scenario validations:</b> N/a</p>

Table 59. Test Case Id: TC\_E\_12\_CSMS

<b>Test case name</b>	<b>Start transaction options - ParkingBayOccupied</b>	
<b>Test case Id</b>	TC_E_12_CSMS	
<b>Use case Id(s)</b>	E01(S1)	
<b>Requirement(s)</b>	E01.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that starts a transaction when the EV and EVSE are connected.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Started</i> <b>triggerReason</b> is <i>EVDetected</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 60. Test Case Id: TC\_E\_38\_CSMS

<b>Test case name</b>	<b>Start Transaction - EV not ready</b>	
<b>Test case Id</b>	TC_E_38_CSMS	
<b>Use case Id(s)</b>	E03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first connect the EV and EVSE OR present a form of identification. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that reports an EV is not ready to start the energy transfer (yet).	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>Authorized</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>EVConnectedPreSession</i>	
	2. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>SuspendedEV</i> <b>eventType</b> is <i>Updated</i>	3. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 61. Test Case Id: TC\_E\_07\_CSMS

<b>Test case name</b>	<b>Stop transaction options - PowerPathClosed - Local stop</b>	
<b>Test case Id</b>	TC_E_07_CSMS	
<b>Use case Id(s)</b>	E06(S5)	
<b>Requirement(s)</b>	E06.FR.06	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when it is locally stopped by an EV driver.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>StopAuthorized</i> <b>transactionInfo.stoppedReason</b> is <i>Local</i> <b>eventType</b> is <i>Ended</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 62. Test Case Id: TC\_E\_08\_CSMS

<b>Test case name</b>	<b>Stop transaction options - EnergyTransfer stopped - StopAuthorized</b>	
<b>Test case Id</b>	TC_E_08_CSMS	
<b>Use case Id(s)</b>	E06(S6)	
<b>Requirement(s)</b>	E06.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the energy transfer stopped normally.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>StopAuthorized</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>EVConnected</i> <b>transactionInfo.stoppedReason</b> is <i>Local</i> <b>eventType</b> is <i>Ended</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 63. Test Case Id: TC\_E\_16\_CSMS

<b>Test case name</b>	<b>Stop transaction options - Deauthorized - Invalid idToken</b>	
<b>Test case Id</b>	TC_E_16_CSMS	
<b>Use case Id(s)</b>	E06(S3)	
<b>Requirement(s)</b>	E06.FR.04,E01.FR.11,E01.FR.12	
<b>System under test</b>	CSMS	
<b>Description</b>	Ocpp 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the transaction gets deauthorized by the status from the idTokenInfo at a TransactionEventResponse message and it has been configured to do so.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>Authorized</i> <b>idToken.idToken</b> <Configured <i>invalid_idtoken_idtoken</i> > <b>idToken.type</b> <Configured <i>invalid_idtoken_type</i> > <b>eventType</b> is <i>Started</i>	<b>2.</b> The CSMS responds with a <b>TransactionEventResponse</b>
	<b>3.</b> The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> <i>Ended</i> <b>triggerReason</b> <i>Deauthorized</i> <b>transactionInfo.stoppedReason</b> <i>DeAuthorized</i>	<b>4.</b> The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 2: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be <i>Invalid</i> or <i>Unknown</i> +	
	<b>Post scenario validations:</b> N/a	

Table 64. Test Case Id: TC\_E\_17\_CSMS

<b>Test case name</b>	<b>Stop transaction options - Deauthorized - EV side disconnect</b>	
<b>Test case Id</b>	TC_E_17_CSMS	
<b>Use case Id(s)</b>	E06(S3)	
<b>Requirement(s)</b>	E06.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the start options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the transaction gets deauthorized by a connection loss from the EV side and it has been configured to do so.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferSuspended</i>	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> <b>triggerReason</b> must be <i>EVCommunicationLost</i> <b>transactionInfo.chargingState</b> must be <i>Idle</i> <b>transactionInfo.stoppedReason</b> must be <i>EVDisconnected</i> <b>eventType</b> must be <i>Ended</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	



Table 65. Test Case Id: TC\_E\_22\_CSMS

<b>Test case name</b>	<b>Stop transaction options - EnergyTransfer stopped - SuspendedEV</b>	
<b>Test case Id</b>	TC_E_22_CSMS	
<b>Use case Id(s)</b>	E06(S6)	
<b>Requirement(s)</b>	E06.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the energy transfer stopped by the EV.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>SuspendedEV</i> <b>transactionInfo.stoppedReason</b> is <i>StoppedByEV</i> <b>eventType</b> is <i>Ended</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 66. Test Case Id: TC\_E\_19\_CSMS

<b>Test case name</b>	<b>Stop transaction options - ParkingBayUnoccupied</b>	
<b>Test case Id</b>	TC_E_19_CSMS	
<b>Use case Id(s)</b>	E06(S1)	
<b>Requirement(s)</b>	E06.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x has a flexible transaction mechanism that allows the transaction start and stop points to be configured differently. This test covers one of the stop options.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that stops a transaction when the EV left the parking bay.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EVDisconnected</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>EVDeparted</i> <b>transactionInfo.stoppedReason</b> is <i>Local</i> <b>eventType</b> is <i>Ended</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 67. Test Case Id: TC\_E\_26\_CSMS

<b>Test case name</b>	<b>Disconnect cable on EV-side - Suspend transaction</b>	
<b>Test case Id</b>	TC_E_26_CSMS	
<b>Use case Id(s)</b>	E10	
<b>Requirement(s)</b>	E10.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station can behave in several different ways when the cable is disconnected at the EV side, based on its configuration. This test case tests one of the possible configuration settings.	
<b>Purpose</b>	To verify if the CSMS can handle a Charging Station that suspends the transaction when the EV and EVSE are disconnected at the EV side AND is able restart the energy transfer after reconnecting the EV and EVSE.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferSuspended</i>	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>EVCommunicationLost</i> <b>transactionInfo.chargingState</b> is <i>Idle</i> <b>eventType</b> is <i>Updated</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
	3. The OCTT notifies the CSMS about the current state of the connector. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> - <b>evseld</b> <i>&lt;Configured evseld&gt;</i> - <b>connectorId</b> <i>&lt;Configured connectorId&gt;</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>component.evse.id</b> <i>&lt;Configured evseld&gt;</i> - <b>component.evse.connectorId</b> <i>&lt;Configured connectorId&gt;</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	4. The CSMS responds accordingly.
	5. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>CablePluggedIn</i> <b>transactionInfo.chargingState</b> is <i>EVConnected</i> <b>eventType</b> is <i>Updated</i>	6. The CSMS responds with a <b>TransactionEventResponse</b>
	7. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>Charging</i> <b>eventType</b> is <i>Updated</i>	8. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 68. Test Case Id: TC\_E\_29\_CSMS

<b>Test case name</b>	<b>Check Transaction status - Transaction with id ongoing - with message in queue</b>	
<b>Test case Id</b>	TC_E_29_CSMS	
<b>Use case Id(s)</b>	E14	
<b>Requirement(s)</b>	E14.FR.02,E14.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the status of a transaction and to find out whether there are queued transaction-related messages, using the <b>GetTransactionStatusRequest</b> message.	
<b>Purpose</b>	To verify if the CSMS is able to request the status of queued TransactionEventRequest messages from a specific transaction by sending a <b>GetTransactionStatusRequest</b> with a transactionId. The OCTT will respond that there are message(s) queued belonging to the ongoing transaction with the requested id.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT closes the WebSocket connection.	
	2. The OCTT waits a number of seconds equal to <i>&lt;Configured Transaction Duration&gt;</i> , then it will reconnect to the CSMS._	
	4. The OCTT responds with a <b>GetTransactionStatusResponse</b> With <b>ongoingIndicator</b> is <i>true</i> <b>messagesInQueue</b> is <i>true</i>	3. The CSMS sends a <b>GetTransactionStatusRequest</b>
	5. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Updated</i> <b>meterValues</b> is present. <b>offline</b> is <i>true</i>	6. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 3: Message: <b>GetTransactionStatusRequest</b> - <b>transactionId</b> <i>&lt;Generated transactionId from Before&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 69. Test Case Id: TC\_E\_30\_CSMS

<b>Test case name</b>	<b>Check Transaction status - Transaction with id ongoing - without message in queue</b>	
<b>Test case Id</b>	TC_E_30_CSMS	
<b>Use case Id(s)</b>	E14	
<b>Requirement(s)</b>	E14.FR.02,E14.FR.05	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the status of a transaction and to find out whether there are queued transaction-related messages, using the <b>GetTransactionStatusRequest</b> message.	
<b>Purpose</b>	To verify if the CSMS is able to request the status of queued TransactionEventRequest messages from a specific transaction by sending a <b>GetTransactionStatusRequest</b> with a transactionId. The OCTT will respond that there is NO message queued belonging to the ongoing transaction with the requested id.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetTransactionStatusResponse</b> With <b>ongoingIndicator</b> is <i>true</i> <b>messagesInQueue</b> is <i>false</i>	1. The CSMS sends a <b>GetTransactionStatusRequest</b>
<b>Tool validations</b>	* Step 1: Message: <b>GetTransactionStatusRequest</b> - <b>transactionId</b> must be <Generated transactionId from Before>	
	<b>Post scenario validations:</b> N/a	

Table 70. Test Case Id: TC\_E\_31\_CSMS

<b>Test case name</b>	<b>Check Transaction status - Transaction with id ended - with message in queue</b>	
<b>Test case Id</b>	TC_E_31_CSMS	
<b>Use case Id(s)</b>	E14	
<b>Requirement(s)</b>	E14.FR.03,E14.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the status of a transaction and to find out whether there are queued transaction-related messages, using the <b>GetTransactionStatusRequest</b> message.	
<b>Purpose</b>	To verify if the CSMS is able to request the status of queued TransactionEventRequest messages from a specific transaction by sending a <b>GetTransactionStatusRequest</b> with a transactionId. The OCTT will respond that there are message(s) queued belonging to an ended transaction with the requested id.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT closes the WebSocket connection.	
	2. The OCTT waits a number of seconds equal to <i>&lt;Configured Transaction duration&gt;</i> , then it will reconnect to the CSMS._	
	3. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Ended</i> <b>offline</b> is <i>true</i> <b>triggerReason</b> is <i>EVCommunicationLost</i> <b>transactionInfo.chargingState</b> is <i>Idle</i> <b>seqNo</b> <i>&lt;Skips two sequence number values&gt;</i>	4. The CSMS responds with a <b>TransactionEventResponse</b>
	6. The OCTT responds with a <b>GetTransactionStatusResponse</b> With <b>ongoingIndicator</b> is <i>false</i> <b>messagesInQueue</b> is <i>true</i>	5. The CSMS sends a <b>GetTransactionStatusRequest</b>
	7. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>StopAuthorized</i> <b>eventType</b> is <i>Updated</i> <b>offline</b> is <i>true</i> <b>seqNo</b> <i>&lt;This is the first of the two skipped values&gt;</i>	8. The CSMS responds with a <b>TransactionEventResponse</b>
	9. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>EVConnected</i> <b>eventType</b> is <i>Updated</i> <b>offline</b> is <i>true</i> <b>seqNo</b> <i>&lt;This is the second of the two skipped values&gt;</i>	10. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 5: Message: <b>GetTransactionStatusRequest</b> - <b>transactionId</b> <i>&lt;Generated transactionId from Before&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 71. Test Case Id: TC\_E\_33\_CSMS

<b>Test case name</b>	<b>Check Transaction status - Without transactionId - with message in queue</b>	
<b>Test case Id</b>	TC_E_33_CSMS	
<b>Use case Id(s)</b>	E14	
<b>Requirement(s)</b>	E14.FR.06,E14.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the status of a transaction and to find out whether there are queued transaction-related messages, using the <b>GetTransactionStatusRequest</b> message.	
<b>Purpose</b>	To verify if the CSMS is able to request the status of queued TransactionEventRequest messages by sending a <b>GetTransactionStatusRequest</b> without a transactionId. The OCTT will respond that there are message(s) queued.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT closes the WebSocket connection.	
	2. The OCTT waits a number of seconds equal to <i>Configured Transaction Duration</i> , then it will reconnect to the CSMS.	
	4. The OCTT responds with a <b>GetTransactionStatusResponse</b> With <b>ongoingIndicator</b> is omitted. <b>messagesInQueue</b> is <i>true</i>	3. The CSMS sends a <b>GetTransactionStatusRequest</b>
	5. The OCTT sends a <b>TransactionEventRequest</b> With <b>eventType</b> is <i>Updated</i> <b>meterValues</b> is present. <b>offline</b> is <i>true</i>	6. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 3: Message: <b>GetTransactionStatusRequest</b> - <b>transactionId</b> must be omitted.	
	<b>Post scenario validations:</b> N/a	

Table 72. Test Case Id: TC\_E\_34\_CSMS

<b>Test case name</b>	<b>Check Transaction status - Without transactionId - without message in queue</b>	
<b>Test case Id</b>	TC_E_34_CSMS	
<b>Use case Id(s)</b>	E14	
<b>Requirement(s)</b>	E14.FR.06,E14.FR.08	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the status of a transaction and to find out whether there are queued transaction-related messages, using the <b>GetTransactionStatusRequest</b> message.	
<b>Purpose</b>	To verify if the CSMS is able to request the status of queued TransactionEventRequest messages by sending a <b>GetTransactionStatusRequest</b> without a transactionId. The OCTT will respond that there are NO message(s) queued.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetTransactionStatusResponse</b> With <b>ongoingIndicator</b> is omitted. <b>messagesInQueue</b> is <i>false</i>	1. The CSMS sends a <b>GetTransactionStatusRequest</b>
<b>Tool validations</b>	* Step 1: Message: <b>GetTransactionStatusRequest</b> - <b>transactionId</b> must be omitted.	
	<b>Post scenario validations:</b> N/a	



Table 73. Test Case Id: TC\_E\_53\_CSMS

<b>Test case name</b>	<b>CSMS accepting seqNo = 0 at start of transaction</b>	
<b>Test case Id</b>	TC_E_53_CSMS	
<b>Use case Id(s)</b>	E01	
<b>Requirement(s)</b>	E01.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	Ocpp 2.0.1 Edition 2 recommends that seqNo starts at 0 for every transaction. CSMS must therefore be robust to a seqNo that is not continuously increasing, but that restarts for new transactions. Since a TransactionEventRequest cannot be rejected, this can only be detected by either the complete absence of a TransactionEventResponse from CSMS or an otherwise misbehaving CSMS.	
<b>Purpose</b>	To verify if the CSMS accepts that a new transactions starts with a seqNo = 0.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i> <u>Note(s)</u> : New transaction will use seqNo 0 for the first TransactionEventRequest.	
	2. Execute <b>Reusable State</b> <i>EVDIsconnected</i>	
	3. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i> <u>Note(s)</u> : New transaction will use seqNo 0 for the first TransactionEventRequest.	
	4. Execute <b>Reusable State</b> <i>EVDIsconnected</i>	
<b>Tool validations</b>	* Step 1: CSMS accepts the message <b>TransactionEventRequest</b> with <i>eventType</i> = Started and <i>seqNo</i> = 0 and answers with a <b>TransactionEventResponse</b> message.	
	* Step 3: CSMS accepts the message <b>TransactionEventRequest</b> with <i>eventType</i> = Started and <i>seqNo</i> = 0 and answers with a <b>TransactionEventResponse</b> message.	

## 7. F Remote Control

Table 74. Test Case Id: TC\_F\_01\_CSMS

Test case name	Remote start transaction - Cable plugin first	
Test case Id	TC_F_01_CSMS	
Use case Id(s)	F01	
Requirement(s)	N/a	
System under test	CSMS	
Description	OCPP 2.x.x allows an EV driver to either first connect the EV and EVSE OR wait for/trigger a RequestStartTransactionRequest. Both sequences will result in being able to charge.	
Purpose	To verify if the CSMS is able to handle a Charging Station that starts a charging session when the EV driver first connects the EV and EVSE, before receiving a RequestStartTransactionRequest message.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EVConnectedPreSession</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to request the Charging Station to start a transaction.	
	2. The OCTT responds with a <b>RequestStartTransactionResponse</b> with <b>status</b> <i>Accepted</i> and <b>transactionId</b> is <i>&lt;Generated transactionId&gt;</i>	1. The CSMS sends a <b>RequestStartTransactionRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with <b>triggerReason</b> is <i>RemoteStart</i> and <b>transactionInfo.remoteStartId</b> is <i>&lt;By CSMS provided remoteStartId&gt;</i> and <b>eventType</b> is <i>Updated</i>	4. The CSMS responds with a <b>TransactionEventResponse</b> .
	5. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i> (State is <i>Authorized</i> and <i>_EVConnected</i> = true)	
Tool validations	* Step 1: Message: <b>RequestStartTransactionRequest</b> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 75. Test Case Id: TC\_F\_02\_CSMS

<b>Test case name</b>	<b>Remote start transaction - Remote start first - AuthorizeRemoteStart is true</b>	
<b>Test case Id</b>	TC_F_02_CSMS	
<b>Use case Id(s)</b>	F02	
<b>Requirement(s)</b>	F02.FR.01, F01.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first wait for/trigger a RequestStartTransactionRequest OR connect the EV and EVSE. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that starts a charging session when the Charging Stations receives a RequestStartTransactionRequest message (while AuthorizeRemoteStart is true), before the EV driver connects the EV and EVSE (within the connectionTimeout). The Charging Station has to authorize beforehand like a local action to start a transaction.	
<b>Prerequisite(s)</b>	<b>AuthEnabled</b> is NOT implemented with mutability ReadOnly and the value set to false	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to request the Charging Station to start a transaction.	
	2. The OCTT responds with a <b>RequestStartTransactionResponse</b> with <b>status</b> <i>Accepted</i> and <b>transactionId</b> is omitted.	1. The CSMS sends a <b>RequestStartTransactionRequest</b>
	3. The OCTT sends a <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured <i>valid_idtoken_idtoken</i> > and <b>idToken.type</b> <Configured <i>valid_idtoken_type</i> >	4. The CSMS responds with a <b>AuthorizeResponse</b> .
	5. The OCTT sends a <b>TransactionEventRequest</b> with <b>triggerReason</b> is <i>RemoteStart</i> and <b>transactionInfo.remoteStartId</b> is <By OCTT generated <i>remoteStartId</i> > and <b>eventType</b> is <i>Started</i>	6. The CSMS responds with a <b>TransactionEventResponse</b> .
	7. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i> ( <b>State</b> is <i>Authorized</i> and <i>_EVConnected</i> = false)	
<b>Tool validations</b>	* Step 1: Message: <b>RequestStartTransactionRequest</b> - <b>idToken.idToken</b> <Configured <i>valid_idtoken_idtoken</i> > - <b>idToken.type</b> <Configured <i>valid_idtoken_type</i> > * Step 4: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 76. Test Case Id: TC\_F\_03\_CSMS

<b>Test case name</b>	<b>Remote start transaction - Remote start first - AuthorizeRemoteStart is false</b>	
<b>Test case Id</b>	TC_F_03_CSMS	
<b>Use case Id(s)</b>	F02	
<b>Requirement(s)</b>	F02.FR.01, F01.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first wait for/trigger a RequestStartTransactionRequest OR connect the EV and EVSE. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that starts a charging session when the Charging Stations receives a RequestStartTransactionRequest message (while AuthorizeRemoteStart is false), before the EV driver connects the EV and EVSE (within the connectionTimeout). The Charging station does NOT have to authorize beforehand like a local action to start a transaction.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to request the Charging Station to start a transaction.	
	2. The OCTT responds with a <b>RequestStartTransactionResponse</b> with <b>status</b> <i>Accepted</i> and <b>transactionId</b> is omitted.	1. The CSMS sends a <b>RequestStartTransactionRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with <b>triggerReason</b> is <i>RemoteStart</i> and <b>transactionInfo.remoteStartId</b> is <i>&lt;By OCTT generated remoteStartId&gt;</i> and <b>eventType</b> is <i>Started</i>	4. The CSMS responds with a <b>TransactionEventResponse</b> .
	5. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i> ( <b>State</b> is <i>Authorized</i> and <b>_EVConnected</b> = false)	
<b>Tool validations</b>	* Step 1: Message: <b>RequestStartTransactionRequest</b> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 77. Test Case Id: TC\_F\_04\_CSMS

<b>Test case name</b>	<b>Remote start transaction - Remote start first - Cable plugin timeout</b>	
<b>Test case Id</b>	TC_F_04_CSMS	
<b>Use case Id(s)</b>	F02, E03	
<b>Requirement(s)</b>	E03.FR.04, E03.FR.05	
<b>System under test</b>	CSMS	
<b>Description</b>	OCPP 2.x.x allows an EV driver to either first wait for/trigger a RequestStartTransactionRequest OR connect the EV and EVSE. Both sequences will result in being able to charge.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station that deauthorizes the transaction after the <b>EVConnectTimeout</b> has been reached.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to request the Charging Station to start a transaction.	
	2. The OCTT responds with a <b>RequestStartTransactionResponse</b> with <b>status</b> <i>Accepted</i> <b>transactionId</b> is omitted.	1. The CSMS sends a <b>RequestStartTransactionRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> . with <b>triggerReason</b> is <i>RemoteStart</i> <b>transactionInfo.remoteStartId</b> is <i>&lt;By OCTT generated remoteStartId&gt;</i> <b>eventType</b> is <i>Started</i>	4. The CSMS responds with a <b>TransactionEventResponse</b> .
	5. The OCTT sends a <b>TransactionEventRequest</b> . with <b>triggerReason</b> is <i>EVConnectTimeout</i> <b>eventType</b> is <i>Updated</i>	6. The CSMS responds with a <b>TransactionEventResponse</b> .
	<u>Note(s):</u> - This step will be executed after the <i>&lt;Configured Transaction Duration&gt;</i> has been reached._	
<b>Tool validations</b>	* Step 1: Message: <b>RequestStartTransactionRequest</b> - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 78. Test Case Id: TC\_F\_06\_CSMS

<b>Test case name</b>	<b>Remote unlock Connector - Without ongoing transaction - Accepted</b>	
<b>Test case Id</b>	TC_F_06_CSMS	
<b>Use case Id(s)</b>	F05	
<b>Requirement(s)</b>	n/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to sent an UnlockConnectorRequest to the charging station. It sometimes happens that a connector of a Charging Station socket does not unlock correctly. This happens most of the time when there is tension on the charging cable. This means the driver cannot unplug his charging cable from the Charging Station. To help a driver, the CSO can send a UnlockConnectorRequest to the Charging Station. The Charging Station will then try to unlock the connector again.	
<b>Purpose</b>	To verify if the CSMS is able to perform the remote unlock connector mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>		
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UnlockConnectorResponse</b> with <b>status</b> <i>Unlocked</i>	1. The CSMS sends a <b>UnlockConnectorRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>UnlockConnectorRequest</b> - <b>evseld</b> <Configured evseld> - <b>connectorId</b> <Configured connectorId>	
	<b>Post scenario validations:</b> - N/a	

Table 79. Test Case Id: TC\_F\_11\_CSMS

Test case name	Trigger message - MeterValues - Specific EVSE	
Test case Id	TC_F_11_CSMS	
Use case Id(s)	F06	
Requirement(s)	F06.FR.01,F06.FR.02	
System under test	CSMS	
Description	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
Purpose	To verify if the CSMS is able to trigger the Charging Station to send a MeterValuesRequest for a specific EVSE, using a TriggerMessageRequest.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>MeterValuesRequest</b> With <b>evseld</b> <Configured evseld> <b>meterValue[0].sampledValue.context Trigger</b>	4. The CSMS responds with a <b>MeterValuesResponse</b>
Tool validations	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>MeterValues</i> - <b>evse.id</b> must be <Configured evseld>	
	<b>Post scenario validations:</b> N/a	

Table 80. Test Case Id: TC\_F\_12\_CSMS

<b>Test case name</b>	<b>Trigger message - MeterValues - All EVSE</b>	
<b>Test case Id</b>	TC_F_12_CSMS	
<b>Use case Id(s)</b>	F06	
<b>Requirement(s)</b>	F06.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
<b>Purpose</b>	To verify if the CSMS is able to trigger the Charging Station to send a MeterValuesRequest for all EVSE, using a TriggerMessageRequest.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>MeterValuesRequest</b> With <b>evseld</b> omitted <b>meterValue[0].sampledValue.context</b> <i>Trigger</i>  <u>Note(s):</u> - <i>This step will be executed for every EVSE.</i>	4. The CSMS responds with a <b>MeterValuesResponse</b>
<b>Tool validations</b>	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>MeterValues</i>	
	<b>Post scenario validations:</b> N/a	



Table 81. Test Case Id: TC\_F\_13\_CSMS

Test case name	Trigger message - TransactionEvent - Specific EVSE	
Test case Id	TC_F_13_CSMS	
Use case Id(s)	F06	
Requirement(s)	F06.FR.01,F06.FR.02	
System under test	CSMS	
Description	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
Purpose	To verify if the CSMS is able to trigger the Charging Station to send a TransactionEventRequest for a specific EVSE, using a TriggerMessageRequest.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> With <b>evse.id</b> <Configured evseld> <b>triggerReason</b> <i>Trigger</i> <b>transactionInfo.chargingState</b> <i>Charging</i> <b>meterValue</b> is present <b>meterValue[0].sampledValue.context</b> <i>Trigger</i>	4. The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>TransactionEvent</i> - <b>evse.id</b> must be <Configured evseld>	
	<b>Post scenario validations:</b> N/a	

Table 82. Test Case Id: TC\_F\_14\_CSMS

<b>Test case name</b>	<b>Trigger message - TransactionEvent - All EVSE</b>	
<b>Test case Id</b>	TC_F_14_CSMS	
<b>Use case Id(s)</b>	F06	
<b>Requirement(s)</b>	F06.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
<b>Purpose</b>	To verify if the CSMS is able to trigger the Charging Station to send a TransactionEventRequest for all EVSE, using a TriggerMessageRequest.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> With <b>evse.id</b> omitted <b>triggerReason</b> <i>Trigger</i> <b>transactionInfo.chargingState</b> <i>Charging</i> <b>meterValue</b> is present <b>meterValue[0].sampledValue.context</b> <i>Trigger</i>  <u>Note(s):</u> - This step will be executed for every EVSE.	4. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>TransactionEvent</i>	
	<b>Post scenario validations:</b> N/a	

Table 83. Test Case Id: TC\_F\_15\_CSMS

Test case name	Trigger message - LogStatusNotification - Idle	
Test case Id	TC_F_15_CSMS	
Use case Id(s)	F06	
Requirement(s)	F06.FR.01	
System under test	CSMS	
Description	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
Purpose	To verify if the CSMS is able to trigger the Charging Station to send a LogStatusNotificationRequest, using a TriggerMessageRequest.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>LogStatusNotificationRequest</b> with status <i>Idle</i>	4. The CSMS responds with a <b>LogStatusNotificationResponse</b>
Tool validations	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>LogStatusNotification</i>	
	<b>Post scenario validations:</b> N/a	

Table 84. Test Case Id: TC\_F\_18\_CSMS

<b>Test case name</b>	<b>Trigger message - FirmwareNotification - Idle</b>	
<b>Test case Id</b>	TC_F_18_CSMS	
<b>Use case Id(s)</b>	F06	
<b>Requirement(s)</b>	F06.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
<b>Purpose</b>	To verify if the CSMS is able to trigger the Charging Station to send a FirmwareStatusNotificationRequest, using a TriggerMessageRequest.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> with status <i>Idle</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b>
<b>Tool validations</b>	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>FirmwareStatusNotification</i>	
	<b>Post scenario validations:</b> N/a	

Table 85. Test Case Id: TC\_F\_20\_CSMS

<b>Test case name</b>	<b>Trigger message - Heartbeat</b>	
<b>Test case Id</b>	TC_F_20_CSMS	
<b>Use case Id(s)</b>	F06	
<b>Requirement(s)</b>	F06.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
<b>Purpose</b>	To verify if the CSMS is able to trigger the Charging Station to send a HeartbeatRequest, using a TriggerMessageRequest.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT sends a <b>HeartbeatRequest</b>	4. The CSMS responds with a <b>HeartbeatResponse</b>
<b>Tool validations</b>	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>Heartbeat</i>	
	<b>Post scenario validations:</b> N/a	

Table 86. Test Case Id: TC\_F\_23\_CSMS

Test case name	Trigger message - StatusNotification - Specific EVSE - Available	
Test case Id	TC_F_23_CSMS	
Use case Id(s)	F06	
Requirement(s)	F06.FR.01,F06.FR.02,F06.FR.13	
System under test	CSMS	
Description	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
Purpose	To verify if the CSMS is able to trigger the Charging Station to send a StatusNotificationRequest for a specific available EVSE, using a TriggerMessageRequest.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
	3. The OCTT notifies the CSMS about the current state of the connector. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> - <b>evseld</b> <i>&lt;Configured evseld&gt;</i> - <b>connectorId</b> <i>&lt;Configured connectorId&gt;</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>component.evse.id</b> <i>&lt;Configured evseld&gt;</i> - <b>component.evse.connectorId</b> <i>&lt;Configured connectorId&gt;</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	4. The CSMS responds accordingly.
Tool validations	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>StatusNotification</i> - <b>evse.id</b> must be <i>&lt;Configured evseld&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 87. Test Case Id: TC\_F\_24\_CSMS

Test case name	Trigger message - StatusNotification - Specific EVSE - Occupied	
Test case Id	TC_F_24_CSMS	
Use case Id(s)	F06	
Requirement(s)	F06.FR.01,F06.FR.02,F06.FR.13	
System under test	CSMS	
Description	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
Purpose	To verify if the CSMS is able to trigger the Charging Station to send a StatusNotificationRequest for a specific occupied EVSE, using a TriggerMessageRequest.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT notifies the CSMS about the current state of the connector. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Occupied</i> - <b>evseld</b> <i>&lt;Configured evseld&gt;</i> - <b>connectorId</b> <i>&lt;Configured connectorId&gt;</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Occupied"</i> - <b>component.name</b> <i>"Connector"</i> - <b>component.evse.id</b> <i>&lt;Configured evseld&gt;</i> - <b>component.evse.connectorid</b> <i>&lt;Configured connectorId&gt;</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	<b>2.</b> The CSMS responds accordingly.
	<b>4.</b> The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>Accepted</i>	<b>3.</b> The CSMS sends a <b>TriggerMessageRequest</b>
	<b>5.</b> The OCTT notifies the CSMS about the current state of the connector. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Occupied</i> - <b>evseld</b> <i>&lt;Configured evseld&gt;</i> - <b>connectorId</b> <i>&lt;Configured connectorId&gt;</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Occupied"</i> - <b>component.name</b> <i>"Connector"</i> - <b>component.evse.id</b> <i>&lt;Configured evseld&gt;</i> - <b>component.evse.connectorid</b> <i>&lt;Configured connectorId&gt;</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	<b>6.</b> The CSMS responds accordingly.

Test case name	Trigger message - StatusNotification - Specific EVSE - Occupied
Tool validations	* Step 1: Message: <b>TriggerMessageRequest</b> - <b>requestedMessage</b> must be <i>StatusNotification</i> - <b>evse.id</b> must be <i>&lt;Configured evseld&gt;</i>
	<b>Post scenario validations:</b> N/a

DRAFT



Table 88. Test Case Id: TC\_F\_27\_CSMS

Test case name	Trigger message - NotImplemented	
Test case Id	TC_F_27_CSMS	
Use case Id(s)	F06	
Requirement(s)	F06.FR.08	
System under test	CSMS	
Description	The CSMS can request a Charging Station to send Charging Station-initiated messages. In the request the CSMS indicates which message it wishes to receive.	
Purpose	To verify if the CSMS is able to handle a Charging Station that does not support the requested message value from a TriggerMessageRequest.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>TriggerMessageResponse</b> with status <i>NotImplemented</i>	1. The CSMS sends a <b>TriggerMessageRequest</b>
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

## 8. G Availability

Table 89. Test Case Id: TC\_G\_03\_CSMS

Test case name	Change Availability EVSE - Operative to inoperative	
Test case Id	TC_G_03_CSMS	
Use case Id(s)	G03	
Requirement(s)	N/a	
System under test	CSMS	
Description	This test case covers how the CSMS requests the Charging Station to change the availability of one of the EVSEs from Operative to Inoperative. An EVSE is considered Operative in any status other than Faulted and Unavailable.	
Purpose	To verify if the CSMS is able to perform the change availability mechanism as described at the OCPP specification.	
Prerequisite(s)	n/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>Unavailable</i> for <Configured evseld>	
Tool validations	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 90. Test Case Id: TC\_G\_04\_CSMS

<b>Test case name</b>	<b>Change Availability EVSE - Inoperative to operative</b>	
<b>Test case Id</b>	TC_G_04_CSMS	
<b>Use case Id(s)</b>	G03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the CSMS requests the Charging Station to change the availability of one of the EVSEs from Inoperative to Operative. An EVSE is considered Operative in any status other than Faulted and Unavailable.	
<b>Purpose</b>	To verify if the CSMS is able to perform the change availability mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> <i>Unavailable</i> for <Configured evseld>	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to change the availability of an EVSE to Operative.	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	3. The OCTT notifies the CSMS about the current state of all connectors belonging to the specified EVSE (and optionally also from the EVSE itself). Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> - <b>evseld</b> <Configured evseld> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"EVSE" / Connector</i> - <b>component.evse.id</b> <Configured evseld> - <b>variable.name</b> <i>"AvailabilityState"</i>	4. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Operative</i> - <b>evse.id</b> <Configured evseld> - <b>connectorId</b> <i>omit</i>	
	<b>Post scenario validations:</b> - N/a	

Table 91. Test Case Id: TC\_G\_05\_CSMS

<b>Test case name</b>	<b>Change Availability Charging Station - Operative to inoperative</b>	
<b>Test case Id</b>	TC_G_05_CSMS	
<b>Use case Id(s)</b>	G04	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	<p>This test case describes how the CSMS requests the Charging Station to change the availability from operative to inoperative.</p> <p>A Charging Station is considered Operative when it is charging or ready for charging.</p> <p>A Charging Station is considered Inoperative when it does not allow any charging.</p>	
<b>Purpose</b>	To verify if the CSMS is able to perform the change availability mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to change the availability of the Charging Station to Inoperative.	
	<p>2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status Accepted</b></p>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	<p>3. The OCTT notifies the CSMS about the current state of all connectors</p> <p>Message: <b>StatusNotificationRequest</b></p> <ul style="list-style-type: none"> <li>- <b>connectorStatus</b> <i>Unavailable</i></li> <li>Message: <b>NotifyEventRequest</b></li> <li>- <b>trigger</b> <i>Delta</i></li> <li>- <b>actualValue</b> <i>"Unavailable"</i></li> <li>- <b>component.name</b> <i>"Connector"</i></li> <li>- <b>variable.name</b> <i>"AvailabilityState"</i></li> </ul>	4. The CSMS responds accordingly.
<b>Tool validations</b>	<p>* Step 1:</p> <p>Message <b>ChangeAvailabilityRequest</b></p> <ul style="list-style-type: none"> <li>- <b>operationalStatus</b> <i>Inoperative</i></li> <li>- <b>evseld</b> <i>omit</i></li> <li>- <b>connectorId</b> <i>omit</i></li> </ul>	
	<p><b>Post scenario validations:</b></p> <ul style="list-style-type: none"> <li>- N/a</li> </ul>	

Table 92. Test Case Id: TC\_G\_06\_CSMS

<b>Test case name</b>	<b>Change Availability Charging station - Inoperative to operative</b>	
<b>Test case Id</b>	TC_G_06_CSMS	
<b>Use case Id(s)</b>	G04	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	<p>This test case describes how the CSMS requests the Charging Station to change the availability from inoperative to operative.</p> <p>A Charging Station is considered Operative when it is charging or ready for charging.</p> <p>A Charging Station is considered Inoperative when it does not allow any charging.</p>	
<b>Purpose</b>	To verify if the CSMS is able to perform the change availability mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> Charging Station set to <i>Unavailable</i> (Original status was Available)	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to change the availability of the Charging Station to Inoperative.	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	3. The OCTT notifies the CSMS about the current state of all connectors. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	4. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Operative</i> - <b>evseld</b> <i>omit</i> - <b>connectorId</b> <i>omit</i>	
	<b>Post scenario validations:</b> - N/a	

Table 93. Test Case Id: TC\_G\_07\_CSMS

<b>Test case name</b>	<b>Change Availability Connector - Operative to inoperative</b>	
<b>Test case Id</b>	TC_G_07_CSMS	
<b>Use case Id(s)</b>	G03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the CSMS requests the Charging Station to change the availability of one of the Connectors of one EVSE from Operative to Inoperative. A Connector is considered Operative in any status other than Faulted and Unavailable.	
<b>Purpose</b>	To verify if the CSMS is able to perform the change availability mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to change the availability of a Connector to Inoperative.	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	3. The OCTT notifies the CSMS about the current state of the connector. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Unavailable</i> - <b>evseId</b> <i>&lt;Configured evseId&gt;</i> - <b>connectorId</b> <i>&lt;Configured connectorId&gt;</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Unavailable"</i> - <b>component.name</b> <i>"Connector"</i> - <b>component.evse.id</b> <i>&lt;Configured evseId&gt;</i> - <b>component.evse.connectorId</b> <i>&lt;Configured connectorId&gt;</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	4. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Inoperative</i> - <b>evse.id</b> <i>&lt;Configured evseId&gt;</i> - <b>evse.connectorId</b> <i>&lt;Configured connectorId&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 94. Test Case Id: TC\_G\_08\_CSMS

<b>Test case name</b>	<b>Change Availability Connector - Inoperative to operative</b>	
<b>Test case Id</b>	TC_G_08_CSMS	
<b>Use case Id(s)</b>	G03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers how the CSMS requests the Charging Station to change the availability of one of the Connectors from one EVSE from Inoperative to Operative. A Connector is considered Operative in any status other than Faulted and Unavailable.	
<b>Purpose</b>	To verify if the CSMS is able to perform the change availability mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> <i>Unavailable</i> for <Configured connectorId>	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to change the availability of a Connector to Operative.	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	3. The OCTT notifies the CSMS about the current state of the connector. Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> - <b>evseId</b> <Configured evseId> - <b>connectorId</b> <Configured connectorId> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> "Available" - <b>component.name</b> "Connector" - <b>component.evse.id</b> <Configured evseId> - <b>component.evse.connectorId</b> <Configured connectorId> - <b>variable.name</b> "AvailabilityState"	4. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Operative</i> - <b>evse.id</b> <Configured evseId> - <b>evse.connectorId</b> <Configured connectorId>	
	<b>Post scenario validations:</b> N/a	

Table 95. Test Case Id: TC\_G\_11\_CSMS

<b>Test case name</b>	<b>Change Availability EVSE - With ongoing transaction</b>	
<b>Test case Id</b>	TC_G_11_CSMS	
<b>Use case Id(s)</b>	G03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	Charging Station	
<b>Description</b>	This test case covers how the CSMS requests the Charging Station to change the availability of one of the EVSEs from Operative to Inoperative. An EVSE is considered Operative in any status other than Faulted and Unavailable.	
<b>Purpose</b>	To verify if the CSMS is able to send a change availability request during a transaction according to the mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State:</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Note(s):</u> Request the CSMS to change the availability to inoperative	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status</b> <i>Scheduled</i>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	<u>Note(s):</u> Wait for <Configured Transaction Duration>	
	3. Execute <b>Reusable State</b> <i>StopAuthorized</i>	
	4. Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	5. Execute <b>Reusable State</b> <i>EVDisconnected</i>	
	6. The OCTT notifies the CSMS about the current state of all connectors with Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Unavailable</i> - <b>evseId</b> <Configured evseId> OR Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> "Unavailable" - <b>component.name</b> "Connector" - <b>component.evse.id</b> <Configured evseId> - <b>variable.name</b> "AvailabilityState"	7. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Inoperative</i> - <b>evse.id</b> <Configured evseId> - <b>connectorId</b> omit	
	<b>Post scenario validations:</b> - A respond to report the state of a connector has been received for all connectors.	



Table 96. Test Case Id: TC\_G\_14\_CSMS

Test case name	Change Availability - Charging Station - With ongoing transaction	
Test case Id	TC_G_14_CSMS	
Use case Id(s)	G04	
Requirement(s)	N/a	
System under test	Charging Station	
Description	This test case covers how the CSMS requests the Charging Station to change the availability of one of the EVSEs from Operative to Inoperative. An EVSE is considered Operative in any status other than Faulted and Unavailable.	
Purpose	To verify if the CSMS is able to send a change availability request during a transaction according to the mechanism as described at the OCPP specification.	
Prerequisite(s)	n/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State:</b> State is <i>EnergyTransferStarted</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Note(s):</u> Request the CSMS to change the availability of the station to inoperative	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status</b> <i>Scheduled</i>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	3. The OCTT notifies the CSMS about the current state of all unoccupied connectors with Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Unavailable</i>	4. The CSMS responds accordingly.
	<u>Note(s):</u> Wait for <Configured Transaction Duration>	
	5. Execute <b>Reusable State</b> <i>StopAuthorized</i>	
	6. Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	7. Execute <b>Reusable State</b> <i>EVDisconnected</i>	
	8. The OCTT notifies the CSMS about the current state of the configured connector with Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Unavailable</i>	9. The CSMS responds accordingly.
Tool validations	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Inoperative</i> - <b>evseld</b> <i>omit</i> - <b>connectorId</b> <i>omit</i>	
	<b>Post scenario validations:</b> - A respond to report the state of a connector has been received for all connectors.	

Table 97. Test Case Id: TC\_G\_17\_CSMS

<b>Test case name</b>	<b>Change Availability - Connector - With ongoing transaction</b>	
<b>Test case Id</b>	TC_G_17_CSMS	
<b>Use case Id(s)</b>	G03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	Charging Station	
<b>Description</b>	This test case covers how the CSMS requests the Charging Station to change the availability of one of the EVSEs from Operative to Inoperative. An EVSE is considered Operative in any status other than Faulted and Unavailable.	
<b>Purpose</b>	To verify if the CSMS is able to send a change availability request during a transaction according to the mechanism as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State:</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Note(s):</u> Request the CSMS to change the availability of one connector to inoperative	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status</b> <i>Scheduled</i>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	<u>Note(s):</u> Wait for <Configured Transaction Duration>	
	3. Execute <b>Reusable State</b> <i>StopAuthorized</i>	
	4. Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	5. Execute <b>Reusable State</b> <i>EVDisconnected</i>	
	6. The OCTT notifies the CSMS about the current state of all connectors with Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Unavailable</i> - <b>evseId</b> <Configured evseId> - <b>connectorId</b> <Configured connectorId>	7. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Inoperative</i> - <b>evse.id</b> <Configured evseId> - <b>evse.connectorId</b> <Configured connectorId>	
	<b>Post scenario validations:</b> - A respond to report the state of a connector has been received for all connectors.	

Table 98. Test Case Id: TC\_G\_20\_CSMS

<b>Test case name</b>	<b>Lock Failure</b>	
<b>Test case Id</b>	TC_G_20_CSMS	
<b>Use case Id(s)</b>	G05	
<b>Requirement(s)</b>	G05.FR.03	
<b>System under test</b>	Charging Station	
<b>Description</b>	This test case describes how the EV Driver is prevented from starting a charge session at the Charging Station while the Connector is not locked properly.	
<b>Purpose</b>	To verify if the CSMS responds on a notifyeventrequest as described at the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1. The OCTT sends a <code>NotifyEventRequest</code> with</b> - <code>eventData.trigger</code> <i>Delta</i> - <code>eventData.component.name</code> <i>"ConnectorPlugRetentionLock"</i> - <code>eventData.variable.name</code> <i>"Problem"</i> - <code>eventData.actualValue</code> <i>"true"</i>	<b>2. The CSMS responds with a <code>NotifyEventResponse</code></b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

## 9. H Reservation

## 10. I Tariff and Cost

Table 99. Test Case Id: TC\_I\_01\_CSMS

Test case name	Show EV Driver running total cost during charging	
Test case Id	TC_I_01_CSMS	
Use case Id(s)	I02	
Requirement(s)	I02.FR.01	
System under test	CSMS	
Description	While a transaction is ongoing, the driver wants to know how much the running total cost is, updated at a relevant interval.	
Purpose	To verify if the CSMS is able to correctly send the running total cost as described in the OCPP specification.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> with <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> Authorized - <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> - <b>idToken.type</b> <Configured valid_idtoken_type> - <b>eventType</b> Updated	4. The CSMS responds with a <b>TransactionEventResponse</b>
	5. Execute Reusable State <b>EVConnectedPreSession</b>	
	6. Execute Reusable State <b>EnergyTransferStarted</b>	
	7. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <b>MeterValuePeriodic</b> <b>eventType</b> is <b>Updated</b> <b>timestamp</b> <The intervals between the timestamps of the received Meter Value messages equals the configured sampled Meter Values interval>. <b>sampledValue.context</b> is <b>Sample.Periodic</b>  <u>Note(s):</u> - This step will be executed every <Configured sampled Meter Values interval> - The OCTT will end the testcase after two MeterValues.	8. The OCTT responds with a <b>TransactionEventResponse</b>
	10. The OCTT responds with a <b>CostUpdatedResponse</b>	9. The CSMS sends a <b>CostUpdatedRequest</b>  <u>Note(s):</u> - This step will be executed after every <b>TransactionEventResponse</b> , if the message did not contain a <b>totalCost</b> .

Test case name	Show EV Driver running total cost during charging
Tool validations	<p>* Step 2: Message <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i></p> <p>* Step 4: Message <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> <i>Accepted</i> - <b>totalCost</b> <i>&lt;Optional&gt;</i></p> <p>* Step 7: Message (Optional) <b>CostUpdatedRequest</b> - <b>transactionId</b> <i>&lt;Generated TransactionId&gt;</i></p> <p><b>Post scenario validations:</b> - N/a</p>

Table 100. Test Case Id: TC\_L02\_CSMS

<b>Test case name</b>	<b>Show EV Driver Final Total Cost After Charging</b>	
<b>Test case Id</b>	TC_L02_CSMS	
<b>Use case Id(s)</b>	I03	
<b>Requirement(s)</b>	I03.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	While a transaction is ongoing, the driver wants to know how much the running total cost is, updated at a relevant interval.	
<b>Purpose</b>	To verify if the CSMS is able to correctly send the total cost as described in the OCPP specification.	
<b>Prerequisite(s)</b>	- N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> state is <i>EVConnectedPostSession</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT notifies the CSMS about the current state of the configured connector.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Available"</i> - <b>component.name</b> <i>"Connector"</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	<b>2.</b> The CSMS responds accordingly.
	<b>3.</b> The OCTT sends a <b>TransactionEventRequest</b> with - <b>triggerReason</b> <i>EVCommunicationLost</i> - <b>eventType</b> <i>Ended</i> - <b>transactionInfo.chargingState</b> <i>Idle</i> - <b>transactionInfo.stoppedReason</b> <i>EVDisconnected</i>	<b>4.</b> The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 4: Message <b>TransactionEventResponse</b> - <b>totalCost</b> <i>&lt;Not omitted&gt;</i>	
	<b>Post scenario validations:</b> - N/a	

# 11. J MeterValues

Table 101. Test Case Id: TC\_J\_01\_CSMS

Test case name	Clock-aligned Meter Values - No transaction ongoing	
Test case Id	TC_J_01_CSMS	
Use case Id(s)	J01	
Requirement(s)	J01.FR.18	
System under test	CSMS	
Description	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
Purpose	To verify if the CSMS is able to handle a Charging Station sending clock-aligned Meter Values, when there is no ongoing transaction.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. The OCTT notifies the CSMS about its measured Meter Values.</p> <p>Message: <b>MeterValuesRequest</b></p> <ul style="list-style-type: none"> <li>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the _&lt;Configured clock-aligned Meter Values interval&gt;.</li> <li>- <b>sampledValue.context</b> is <i>Sample.Clock</i></li> </ul> <p>Message: <b>NotifyEventRequest</b></p> <ul style="list-style-type: none"> <li>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the _&lt;Configured clock-aligned Meter Values interval&gt;.</li> <li>- <b>trigger</b> is <i>Periodic</i></li> <li>- <b>component.name</b> is <i>FiscalMetering</i></li> </ul> <p><u>Note(s):</u></p> <ul style="list-style-type: none"> <li>- This step will be executed every _&lt;Configured clock-aligned Meter Values interval&gt;</li> <li>- This step will be executed for evseld=0 and all configured EVSE.</li> <li>- The OCTT will end the testcase after it has send three Meter Value messages.</li> </ul>	<p>2. The CSMS responds accordingly.</p>
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 102. Test Case Id: TC\_J\_02\_CSMS

<b>Test case name</b>	<b>Clock-aligned Meter Values - Transaction ongoing</b>	
<b>Test case Id</b>	TC_J_02_CSMS	
<b>Use case Id(s)</b>	J01	
<b>Requirement(s)</b>	J01.FR.18	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station sending clock-aligned Meter Values, when there is an ongoing transaction.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i> for <Configured evseld>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. The OCTT notifies the CSMS about its measured Meter Values.</p> <p>Message: <b>MeterValuesRequest</b></p> <ul style="list-style-type: none"> <li>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured clock-aligned Meter Values interval&gt;.</li> <li>- <b>sampledValue.context</b> is <i>Sample.Clock</i></li> </ul> <p>Message: <b>NotifyEventRequest</b></p> <ul style="list-style-type: none"> <li>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured clock-aligned Meter Values interval&gt;.</li> <li>- <b>trigger</b> is <i>Periodic</i></li> <li>- <b>component.name</b> is <i>FiscalMetering</i></li> </ul> <p><u>Note(s):</u></p> <ul style="list-style-type: none"> <li>- This step will be executed every &lt;Configured clock-aligned Meter Values interval&gt;</li> <li>- This step will be executed for evseld=0 and all configured idle EVSE.</li> </ul>	<p>2. The CSMS responds accordingly.</p>
	<p>3. The OCTT sends a <b>TransactionEventRequest</b></p> <p>With <b>triggerReason</b> is <i>MeterValueClock</i></p> <p><b>eventType</b> is <i>Updated</i></p> <p><b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured clock-aligned Meter Values interval&gt;.</p> <p><b>sampledValue.context</b> is <i>Sample.Clock</i></p> <p><u>Note(s):</u></p> <ul style="list-style-type: none"> <li>- This step will be executed every &lt;Configured clock-aligned Meter Values interval&gt;</li> <li>- The OCTT will end the testcase after the &lt;Configured transaction duration&gt; is reached.</li> </ul>	<p>4. The CSMS responds with a <b>TransactionEventResponse</b></p>



Test case name	Clock-align Meter Values - Transaction ongoing
Tool validations	N/a
	Post scenario validations: N/a

DRAFT

Table 103. Test Case Id: TC\_J\_03\_CSMS

<b>Test case name</b>	<b>Clock-aligned Meter Values - EventType Ended</b>	
<b>Test case Id</b>	TC_J_03_CSMS	
<b>Use case Id(s)</b>	J01	
<b>Requirement(s)</b>	J01.FR.18	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station sending clock-aligned Meter Values, when a transaction ends.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. Execute <b>Reusable State</b> <i>EVDisconnected</i></p> <p>- The <b>TransactionEventRequest</b> containing eventType <i>Ended</i> contains the MeterValue field.</p> <p>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured value at configured clock-aligned Tx ended Meter Values interval&gt;.</p> <p>- <b>sampledValue.context</b> is <i>Sample.Clock</i> AND the last one has <i>Transaction.End</i></p> <p><u>Note(s):</u></p> <p>- This step will be executed after the _&lt;Configured transaction duration&gt; is reached._</p> <p>- This causes the transaction to stop.</p>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 104. Test Case Id: TC\_J\_04\_CSMS

<b>Test case name</b>	<b>Clock-aligned Meter Values - Signed</b>	
<b>Test case Id</b>	TC_J_04_CSMS	
<b>Use case Id(s)</b>	J01	
<b>Requirement(s)</b>	J01.FR.21	
<b>System under test</b>	CSMS	
<b>Description</b>	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station sending clock-aligned Meter Values, when a transaction ends.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. Execute <b>Reusable State</b> <i>EVDisconnected</i></p> <p>- The <b>TransactionEventRequest</b> containing eventType <i>Ended</i> contains the MeterValue field.</p> <p>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured value at configured clock-aligned Tx ended Meter Values interval&gt;.</p> <p>- <b>sampledValue.context</b> is <i>Sample.Clock</i> AND the last one has <i>Transaction.End</i></p> <p>- <b>sampledValue.signedMeterValue</b> is &lt;Generated <i>SignedMeterValueType</i>&gt;</p> <p><u>Note(s):</u></p> <p>- This step will be executed after the &lt;Configured transaction duration&gt; is reached.</p> <p>- This causes the transaction to stop.</p>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 105. Test Case Id: TC\_J\_07\_CSMS

Test case name	Sampled Meter Values - EventType Started - EVSE known	
Test case Id	TC_J_07_CSMS	
Use case Id(s)	J02	
Requirement(s)	J02.FR.19	
System under test	CSMS	
Description	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
Purpose	To verify if the CSMS is able to handle a Charging Station sending start sampled Meter Values, when a transaction starts.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. Execute <b>Reusable State</b> <i>EVConnectedPreSession</i></p> <p>- The <b>TransactionEventRequest</b> contains the MeterValue field.</p> <p>- <b>sampledValue.context</b> is <i>Transaction.Begin</i></p>	
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 106. Test Case Id: TC\_J\_08\_CSMS

Test case name	Sampled Meter Values - EventType Started - EVSE not known	
Test case Id	TC_J_08_CSMS	
Use case Id(s)	J02	
Requirement(s)	J02.FR.19	
System under test	CSMS	
Description	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
Purpose	To verify if the CSMS is able to handle a Charging Station sending start sampled Meter Values, when a transaction starts.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>Authorized</i>	
	2. Execute <b>Reusable State</b> <i>EVConnectedPreSession</i>	
	<ul style="list-style-type: none"> <li>- The <b>TransactionEventRequest</b> contains the MeterValue field.</li> <li>- <b>sampledValue.context</b> is <i>Transaction.Begin</i></li> </ul>	
	3. Execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 107. Test Case Id: TC\_J\_09\_CSMS

Test case name	Sampled Meter Values - EventType Updated	
Test case Id	TC_J_09_CSMS	
Use case Id(s)	J02	
Requirement(s)	J02.FR.19	
System under test	CSMS	
Description	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
Purpose	To verify if the CSMS is able to handle a Charging Station sending sampled Meter Values, when there is an ongoing transaction.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p><b>1. The OCTT sends a <b>TransactionEventRequest</b></b> With <b>triggerReason</b> is <i>MeterValuePeriodic</i> <b>eventType</b> is <i>Updated</i> <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured sampled Meter Values interval&gt;. <b>sampledValue.context</b> is <i>Sample.Periodic</i></p> <p><u>Note(s):</u>  - This step will be executed every _&lt;Configured sampled Meter Values interval&gt;  - The OCTT will end the testcase after three MeterValues.</p>	<p><b>2. The CSMS responds with a <b>TransactionEventResponse</b></b></p>
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 108. Test Case Id: TC\_J\_10\_CSMS

Test case name	Sampled Meter Values - EventType Ended	
Test case Id	TC_J_10_CSMS	
Use case Id(s)	J02	
Requirement(s)	J02.FR.19	
System under test	CSMS	
Description	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
Purpose	To verify if the CSMS is able to handle a Charging Station sending sampled Meter Values, when a transaction ends.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. Execute <b>Reusable State</b> <i>EVDIsconnected</i></p> <p>- The <b>TransactionEventRequest</b> containing eventType <i>Ended</i> contains the MeterValue field.</p> <p>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured value at configured clock-aligned Tx ended Meter Values interval&gt;.</p> <p>- <b>sampledValue.context</b> is <i>Sample.Periodic</i> AND the last one has <i>Transaction.End</i></p> <p><u>Note(s):</u></p> <p>- This step will be executed after the _&lt;Configured transaction duration&gt; is reached._</p> <p>- This causes the transaction to stop.</p>	
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	

Table 109. Test Case Id: TC\_J\_11\_CSMS

Test case name	Sampled Meter Values - Signed	
Test case Id	TC_J_11_CSMS	
Use case Id(s)	J02	
Requirement(s)	J02.FR.21	
System under test	CSMS	
Description	The Charging Station samples the electrical meter or other sensor/transducer hardware to provide information about its Meter Values. Depending on configuration settings, the Charging Station will send Meter Values.	
Purpose	To verify if the CSMS is able to handle a Charging Station sending sampled Meter Values, when a transaction ends.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. Execute <b>Reusable State</b> <i>EVDIsconnected</i></p> <p>- The <b>TransactionEventRequest</b> containing eventType <i>Ended</i> contains the MeterValue field.</p> <p>- <b>timestamp</b> &lt;The intervals between the timestamps of the received Meter Value messages equals the configured value at configured clock-aligned Tx ended Meter Values interval&gt;.</p> <p>- <b>sampledValue.context</b> is <i>Sample.Periodic</i> AND the last one has <i>Transaction.End</i></p> <p>- <b>sampledValue.signedMeterValue</b> is &lt;Generated SignedMeterValueType&gt;</p> <p><u>Note(s):</u></p> <p>- This step will be executed after the &lt;Configured transaction duration&gt; is reached.</p> <p>- This causes the transaction to stop.</p>	
Tool validations	N/a	
	<b>Post scenario validations:</b> N/a	



## 12. K SmartCharging

Table 110. Test Case Id: TC\_K\_01\_CSMS

Test case name	Set charging profile - TxDefaultProfile - Specific EVSE	
Test case Id	TC_K_01_CSMS	
Use case Id(s)	K01	
Requirement(s)	K01.FR.31	
System under test	CSMS	
Description	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
Purpose	To verify if the CSMS is able to send a TxDefaultProfile charging profile for a specific EVSE as described at the OCPP specification.	
Prerequisite(s)	n/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetChargingProfileRequest</b> with - <b>chargingProfile.id</b> <Configured chargingProfileId>
Tool validations	* Step 1: Message <b>SetChargingProfileRequest</b> <b>evseId</b> <Configured evseId> AND <b>chargingProfile.stackLevel</b> <Configured stackLevel> AND <b>chargingProfile.chargingProfilePurpose</b> TxDefaultProfile AND <b>chargingProfile.chargingProfileKind</b> Absolute AND <b>chargingProfile.validFrom</b> now AND <b>chargingProfile.validTo</b> now + <Configured Charging Schedule Duration> AND <b>chargingProfile.chargingSchedule.startSchedule</b> now AND <b>chargingProfile.chargingSchedule.chargingRateUnit</b> <Configured chargingRateUnit> AND <b>chargingProfile.chargingSchedule.duration</b> <Configured duration> AND <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod</b> <Configured startPeriod> AND <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit</b> 6.0 or 6000.0 AND <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> where <Configured numberPhases> not 3 OR <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> or <omit> where <Configured numberPhases> 3	
	<b>Post scenario validations:</b> - N/a	

Table 111. Test Case Id: TC\_K\_02\_CSMS

<b>Test case name</b>	<b>Set charging profile - TxProfile without ongoing transaction on the specified EVSE</b>	
<b>Test case Id</b>	TC_K_02_CSMS	
<b>Use case Id(s)</b>	K01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to send a TxProfile and read the charger's feedback while no transaction is ongoing for a specific EVSE as described at the OCPP specification.	
<b>Prerequisite(s)</b>	If the CSMS supports sending a TxProfile while there is no transaction ongoing.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with status <i>Rejected</i>	1. The CSMS sends a <b>SetChargingProfileRequest</b> - <b>chargingProfile.id</b> <Configured chargingProfileId>
<b>Tool validations</b>	* Step 1: Message <b>SetChargingProfileRequest</b> - <b>evsId</b> <Configured evsId> AND - <b>chargingProfile.chargingProfilePurpose</b> <i>TxProfile</i> AND - <b>chargingProfile.stackLevel</b> <Configured stackLevel> AND - <b>chargingProfile.chargingProfileKind</b> <i>Relative</i> AND - <b>chargingProfile.chargingSchedule.chargingRateUnit</b> <Configured chargingRateUnit> AND - <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod</b> 0 AND - <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit</b> 7.0 or 7000.0 AND - <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> where <Configured numberPhases> not 3 OR - <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> or <omit> where <Configured numberPhases> 3	
	<b>Post scenario validations:</b> - N/a	

Table 112. Test Case Id: TC\_K\_03\_CSMS

<b>Test case name</b>	<b>Set charging profile - ChargingStationMaxProfile</b>	
<b>Test case Id</b>	TC_K_03_CSMS	
<b>Use case Id(s)</b>	K01	
<b>Requirement(s)</b>	K01.FR.31, K01.FR.38	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to send a ChargingStationMaxProfile charging profile as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetChargingProfileRequest - chargingProfile.id</b> <Configured chargingProfileId>
<b>Tool validations</b>	* Step 1: Message <b>SetChargingProfileRequest</b> <b>evseld 0</b> AND <b>chargingProfile.stackLevel</b> <Configured stackLevel> AND <b>chargingProfile.chargingProfilePurpose</b> ChargingStationMaxProfile_ AND <b>chargingProfile.chargingProfileKind</b> Absolute OR Relative <b>chargingProfile.chargingSchedule.chargingRateUnit</b> <Configured ChargingRateUnit> <b>chargingProfile.chargingSchedule.duration</b> <Configured duration> <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0</b> <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 8.0 or 8000.0</b> <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> where <Configured numberPhases> not 3 OR <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> or <omit> where <Configured numberPhases> 3 <b>chargingProfile.validFrom</b> <Not omitted> <b>chargingProfile.validTo</b> <Not omitted> <b>chargingProfile.chargingSchedule.startSchedule</b> <Not omitted>	
	<b>Post scenario validations:</b> - N/a	

Table 113. Test Case Id: TC\_K\_04\_CSMS

<b>Test case name</b>	<b>Replace charging profile - With chargingProfileId</b>	
<b>Test case Id</b>	TC_K_04_CSMS	
<b>Use case Id(s)</b>	n/a	
<b>Requirement(s)</b>	n/a	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to replace a charging profile with the same ProfileKind, Purpose, and stackLevel, but a different limit.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetChargingProfileRequest</b> with <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 8.0 or 8000.0</b>
	4. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	3. The CSMS sends a <b>SetChargingProfileRequest</b> with <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6.0 or 6000.0</b>
<b>Tool validations</b>	* Step 3: Message <b>SetChargingProfileRequest</b> <b>chargingProfile.id</b> <same id for both chargingProfiles>	
	<b>Post scenario validations:</b> - N/a	

Table 114. Test Case Id: TC\_K\_05\_CSMS

<b>Test case name</b>	<b>Clear Charging Profile - With chargingProfileId</b>	
<b>Test case Id</b>	TC_K_05_CSMS	
<b>Use case Id(s)</b>	K10	
<b>Requirement(s)</b>	K10.FR.02	
<b>System under test</b>	Charging Station	
<b>Description</b>	If the CSMS wishes to clear some or all of the charging profiles that were previously sent to the Charging Station, then the CSMS sends a ClearChargingProfileRequest to the Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to request the charging station to clear a specific charging profile (not TxDefault) with only a chargingProfileId as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> CSMS sends a <b>GetChargingProfilesRequest</b> OCTT responds with a <b>GetChargingProfilesResponse</b> with status <i>Accepted</i> OCTT sends a <b>ReportChargingProfilesRequest</b> CSMS responds with a <b>ReportChargingProfilesResponse</b>	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>ClearChargingProfileResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>ClearChargingProfileRequest</b> with <b>chargingProfileId</b> <Generated chargingProfileId> AND <b>chargingProfileCriteria</b> omit
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 115. Test Case Id: TC\_K\_06\_CSMS

<b>Test case name</b>	<b>Clear Charging Profile - With stackLevel/purpose combination for one profile</b>	
<b>Test case Id</b>	TC_K_06_CSMS	
<b>Use case Id(s)</b>	K10	
<b>Requirement(s)</b>	K10.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	If the CSMS wishes to clear some or all of the charging profiles that were previously sent to the Charging Station, then the CSMS sends a ClearChargingProfileRequest to the Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to request the charging station to clear a specific charging profile with a stackLevel/purpose combination for a chargingProfileId as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>ClearChargingProfileResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ClearChargingProfileRequest</b> with <b>chargingProfilePurpose TxDefaultProfile</b> AND <b>evseld &lt;Configured evseld&gt;</b> AND <b>stackLevel &lt;Configured stackLevel&gt;</b>
<b>Tool validations</b>	* Step 1: Message <b>ClearChargingProfileRequest</b> <b>chargingProfileCriteria.chargingProfilePurpose TxDefaultProfile</b> AND <b>chargingProfileCriteria.stackLevel &lt;Configured stackLevel&gt;</b> AND <b>chargingProfileCriteria.evseld &lt;Configured evseld&gt;</b>	
	<b>Post scenario validations:</b> - N/a	

Table 116. Test Case Id: TC\_K\_08\_CSMS

<b>Test case name</b>	<b>Clear Charging Profile - Without previous charging profile</b>	
<b>Test case Id</b>	TC_K_08_CSMS	
<b>Use case Id(s)</b>	K10	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	If the CSMS wishes to clear some or all of the charging profiles that were previously sent to the Charging Station, then the CSMS sends a ClearChargingProfileRequest to the Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to request the charging station to clear a specific charging profile with a chargingProfileId and stackLevel/purpose combination while the Charging stations does not accept as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>ClearChargingProfileResponse</b> with <b>status Unknown</b>	1. The CSMS sends a <b>ClearChargingProfileRequest</b> with <b>chargingProfilePurpose TxDefaultProfile</b> AND <b>evseld &lt;Configured evseld&gt;</b> AND <b>stackLevel &lt;Configured stackLevel&gt;</b>
<b>Tool validations</b>	* Step 1: Message <b>ClearChargingProfileRequest</b> <b>chargingProfilePurpose TxDefaultProfile</b> AND <b>evseld &lt;Configured evseld&gt;</b> AND <b>stackLevel &lt;Configured stackLevel&gt;</b>	
	<b>Post scenario validations:</b> - N/a	

Table 117. Test Case Id: TC\_K\_10\_CSMS

<b>Test case name</b>	<b>Set charging profile - TxDefaultProfile - All EVSE</b>	
<b>Test case Id</b>	TC_K_10_CSMS	
<b>Use case Id(s)</b>	K01	
<b>Requirement(s)</b>	K01.FR.31	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to send a TxDefaultProfile charging profile for all EVSE as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetChargingProfileRequest</b> with - <b>chargingProfile.id</b> <Configured chargingProfileId>
<b>Tool validations</b>	* Step 1: Message <b>SetChargingProfileRequest</b> <b>evseld 0</b> AND <b>chargingProfile.stackLevel</b> <Configured stackLevel> AND <b>chargingProfile.chargingProfilePurpose</b> TxDefaultProfile AND <b>chargingProfile.chargingProfileKind</b> Absolute AND <b>chargingProfile.validFrom</b> <Not omitted> AND <b>chargingProfile.validTo</b> <Not omitted> AND <b>chargingProfile.chargingSchedule.startSchedule</b> <Not omitted> AND <b>chargingProfile.chargingSchedule.chargingRateUnit</b> <Configured ChargingRateUnit> AND <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0</b> AND <b>chargingProfile.chargingSchedule.duration</b> <Configured duration> <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6.0 or 6000.0</b> AND <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> where <Configured numberPhases> not 3 OR <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> <Configured numberPhases> or <omit> where <Configured numberPhases> 3	
	<b>Post scenario validations:</b> - N/a	



Table 118. Test Case Id: TC\_K\_15\_CSMS

<b>Test case name</b>	<b>Set charging profile - Not Supported</b>	
<b>Test case Id</b>	TC_K_15_CSMS	
<b>Use case Id(s)</b>	K01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to send a Profile, while the charging station does not support chargingprofiles, and read the response as described at the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>2. The OCTT responds with RPC Framework: CALLERROR: NotSupported.</p>	<p>1. The CSMS sends a <b>SetChargingProfileRequest</b> with:</p> <p><b>evseld</b> &lt;Configured evseld&gt; AND</p> <p><b>chargingProfile.stackLevel</b> &lt;Configured stackLevel&gt; AND</p> <p><b>chargingProfile.chargingProfilePurpose</b> TxDefaultProfile AND</p> <p><b>chargingProfile.chargingProfileKind</b> Absolute AND</p> <p><b>chargingProfile.validFrom</b> &lt;Not omitted&gt; AND</p> <p><b>chargingProfile.validTo</b> &lt;Not omitted&gt; AND</p> <p><b>chargingProfile.chargingSchedule.startSchedule</b> &lt;Not omitted&gt; AND</p> <p><b>chargingProfile.chargingSchedule.chargingRateUnit</b> &lt;Configured ChargingRateUnit&gt; AND</p> <p><b>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod</b> 0 AND</p> <p><b>chargingProfile.chargingSchedule.duration</b> &lt;Configured duration&gt;</p> <p><b>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit</b> 6.0 or 6000.0 AND</p> <p><b>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases</b> &lt;Configured numberPhases&gt;</p>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	

Table 119. Test Case Id: TC\_K\_19\_CSMS

<b>Test case name</b>	<b>Set charging profile - Recurrency</b>	
<b>Test case Id</b>	TC_K_19_CSMS	
<b>Use case Id(s)</b>	K01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to send a Profile with a recurrencyKind specified as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>SetChargingProfileRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetChargingProfileRequest</b> - <b>evseld</b> <Configured evseld> AND - <b>chargingProfile.stackLevel</b> <Configured stackLevel> AND - <b>chargingProfile.chargingProfilePurpose</b> <i>TxDefaultProfile</i> AND - <b>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod</b> 0 AND - <b>chargingProfile.chargingProfileKind</b> <i>Recurring</i> AND - <b>chargingProfile.recurrencyKind</b> <Configured recurrencyKind>	
	<b>Post scenario validations:</b> - N/a	

Table 120. Test Case Id: TC\_K\_29\_CSMS

<b>Test case name</b>	<b>Get charging profile - Evseld 0</b>	
<b>Test case Id</b>	TC_K_29_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request charging profiles installed on the charging station itself and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b> with - <b>evseld 0</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>&lt;Received requestId&gt;</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld 0</b> AND - <b>chargingProfile.chargingProfilePurpose</b> <i>&lt;Configured chargingProfilePurpose&gt;</i>	
	<b>Post scenario validations:</b> - N/a	

Table 121. Test Case Id: TC\_K\_30\_CSMS

<b>Test case name</b>	<b>Get charging profile - Evseld &gt; 0</b>	
<b>Test case Id</b>	TC_K_30_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request charging profiles installed on a specific EVSE and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>&lt;Received requestId&gt;</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld</b> <i>&lt;Configured evseld&gt;</i>	
	<b>Post scenario validations:</b> - N/a	

Table 122. Test Case Id: TC\_K\_31\_CSMS

<b>Test case name</b>	<b>Get charging profile - No Evseld</b>	
<b>Test case Id</b>	TC_K_31_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request all charging profiles installed on a charger and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b> with - <b>requestId</b> <i>&lt;Received requestId&gt;</i>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>&lt;Received requestId&gt;</i> AND - <b>tbc</b> <i>true</i> AND - <b>evseld</b> <i>i</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
	<u>Note(s):</u> - Step 3 and 4 are repeated for every evse	
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld</b> <i>omit</i>	
	<b>Post scenario validations:</b> - N/a	

Table 123. Test Case Id: TC\_K\_32\_CSMS

<b>Test case name</b>	<b>Get charging profile - chargingProfileId</b>	
<b>Test case Id</b>	TC_K_32_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request a specific charging profile and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest - chargingProfileId</b> <Received chargingProfileId>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>Generated Id</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>chargingProfileId</b> <received chargingProfileId> AND - <b>requestId</b> <Generated Id>	
	<b>Post scenario validations:</b> - N/a	

Table 124. Test Case Id: TC\_K\_33\_CSMS

<b>Test case name</b>	<b>Get charging profile - Evseld &gt; 0 + stackLevel</b>	
<b>Test case Id</b>	TC_K_33_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request charging profiles with a specific stackLevel installed on a specific EVSE and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>Generated Id</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld</b> <Configured evseld> AND - <b>chargingProfile.stackLevel</b> <Configured stackLevel>	
	<b>Post scenario validations:</b> - N/a	

Table 125. Test Case Id: TC\_K\_34\_CSMS

<b>Test case name</b>	<b>Get charging profile - Evseld &gt; 0 + chargingLimitSource</b>	
<b>Test case Id</b>	TC_K_34_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request charging profiles with a specific chargingLimitSource installed on a specific EVSE and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>Generated Id</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld</b> <Configured evseld> AND - <b>chargingProfile.chargingLimitSource</b> <Configured chargingLimitSource>	
	<b>Post scenario validations:</b> - N/a	



Table 126. Test Case Id: TC\_K\_35\_CSMS

<b>Test case name</b>	<b>Get charging profile - Evseld &gt; 0 + chargingProfilePurpose</b>	
<b>Test case Id</b>	TC_K_35_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request charging profiles with a specific chargingProfilePurpose installed on a specific EVSE and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>Generated Id</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld</b> <Configured evseld> AND - <b>chargingProfile.chargingProfilePurpose</b> <Configured chargingProfilePurpose>	
	<b>Post scenario validations:</b> - N/a	

Table 127. Test Case Id: TC\_K\_36\_CSMS

<b>Test case name</b>	<b>Get charging profile - Evseld &gt; 0 + chargingProfilePurpose + stackLevel</b>	
<b>Test case Id</b>	TC_K_36_CSMS	
<b>Use case Id(s)</b>	K09	
<b>Requirement(s)</b>	K09.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	With the GetChargingProfilesRequest message the CSMS can ask a Charging Station to report all, or a subset of all the install Charging Profiles from the different possible sources. This can be used for some automatic smart charging control system, or for debug purposes by a CSO.	
<b>Purpose</b>	To verify if the CSMS is able to request charging profiles with a specific chargingProfilePurpose AND stackLevel installed on a specific EVSE and read in the reports as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>Generated Id</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>GetChargingProfilesRequest</b> - <b>evseld</b> <Configured evseld> AND - <b>chargingProfile.chargingProfilePurpose</b> <Configured chargingProfilePurpose> - <b>chargingProfile.stackLevel</b> <Configured stackLevel>	
	<b>Post scenario validations:</b> - N/a	

Table 128. Test Case Id: TC\_K\_60\_CSMS

<b>Test case name</b>	<b>Set charging profile - TxProfile with ongoing transaction on the specified EVSE</b>	
<b>Test case Id</b>	TC_K_60_CSMS	
<b>Use case Id(s)</b>	K01	
<b>Requirement(s)</b>	K01.FR.03, K01.FR.31	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sets a TxProfile on a specific EVSE for a currently ongoing transaction.	
<b>Purpose</b>	To verify if the CSMS is able to exchange messages to set a TxProfile on a specific EVSE for a currently ongoing transaction.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> With <b>status</b> is <i>Accepted</i>	1. The CSMS sends a <b>SetChargingProfileRequest</b>
<b>Tool validations</b>	* Step 1: (Message: <b>SetChargingProfileRequest</b> ) <b>ChargingProfilePurpose</b> is <i>TxProfile</i> AND <b>evseld</b> is <i>&lt;Configured evseld&gt;</i> AND <b>transactionId</b> <i>&lt;Generated transactionId&gt;</i>	
	<b>Post scenario validations:</b> N/a	

Table 129. Test Case Id: TC\_K\_37\_CSMS

<b>Test case name</b>	<b>Remote start transaction with charging profile - Success</b>	
<b>Test case Id</b>	TC_K_37_CSMS	
<b>Use case Id(s)</b>	K05,F01	
<b>Requirement(s)</b>	K05.FR.02,F01.FR.08,F01.FR.09,F01.FR.11	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sets a TxProfile on a specific EVSE inside a RequestStartTransactionRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to set a TxProfile on a specific EVSE in a RequestStartTransactionRequest message.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>RequestStartTransactionResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>RequestStartTransactionRequest</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason RemoteStart</b> <b>transactionInfo.remoteStartId</b> is present.	4. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	* Step 1: Message: <b>RequestStartTransactionRequest</b> with <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type> <b>idToken.idToken</b> <Configured valid idToken> <b>idToken.type</b> <Configured valid idToken type> <b>evseld</b> <Configured evseld> <b>chargingProfile</b> contains: <b>chargingProfile.chargingProfilePurpose</b> is TxProfile <b>chargingProfile.transactionId</b> is omitted <b>chargingProfile.chargingProfileKind</b> is Relative	
	<b>Post scenario validations:</b> N/a	

Table 130. Test Case Id: TC\_K\_43\_CSMS

<b>Test case name</b>	<b>Get Composite Schedule - Specific EVSE</b>	
<b>Test case Id</b>	TC_K_43_CSMS	
<b>Use case Id(s)</b>	K08	
<b>Requirement(s)</b>	K08.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS requests a composite schedule which is a combination of local limits and the prevailing Charging Profiles of the different chargingProfilePurposes and stack levels.	
<b>Purpose</b>	To verify if the CSMS is able to calculate request a composite schedule from the Charging Station for a specific EVSE.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>2. The OCTT responds with a <b>GetCompositeScheduleResponse</b> With <b>status Accepted</b>  <b>schedule.evseId 1</b>  <b>schedule.duration</b> is 300  <b>schedule.chargingRateUnit</b> &lt;Specified chargingRateUnit from step 1&gt;  <b>schedule.chargingSchedulePeriod[0].startPeriod 0</b>  <i>Note: Multiply limit by 1000 if chargingRateUnit is W</i>  <b>schedule.chargingSchedulePeriod[0].limit 10</b></p>	<p>1. The CSMS sends a <b>GetCompositeScheduleRequest</b></p>
<b>Tool validations</b>	<p>* Step 1:  (Message: <b>GetCompositeScheduleRequest</b>)  <b>evseId 1</b>  <b>duration</b> is &lt;Configured duration&gt;  <b>chargingRateUnit</b> &lt;Configured chargingRateUnit&gt;</p>	
	<p><b>Post scenario validations:</b>  N/a</p>	

Table 131. Test Case Id: TC\_K\_44\_CSMS

<b>Test case name</b>	<b>Get Composite Schedule - Charging Station</b>	
<b>Test case Id</b>	TC_K_44_CSMS	
<b>Use case Id(s)</b>	K08	
<b>Requirement(s)</b>	K08.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS requests a composite schedule which is a combination of local limits and the prevailing Charging Profiles of the different chargingProfilePurposes and stack levels.	
<b>Purpose</b>	To verify if the CSMS is able to calculate request a composite schedule from the Charging Station.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>2. The OCTT responds with a <b>GetCompositeScheduleResponse</b> With <b>status Accepted</b> <b>schedule.evseId 0</b> <b>schedule.duration</b> is 300 <b>schedule.chargingRateUnit</b> &lt;Specified chargingRateUnit from step 1&gt; <b>schedule.chargingSchedulePeriod[0].startPeriod 0</b> Note: Multiply limit by 1000 if chargingRateUnit is W <b>schedule.chargingSchedulePeriod[0].limit 10</b></p>	<p>1. The CSMS sends a <b>GetCompositeScheduleRequest</b></p>
<b>Tool validations</b>	<p>* Step 1: (Message: <b>GetCompositeScheduleRequest</b>) <b>evseId 0</b> <b>duration</b> is &lt;Configured duration&gt; <b>chargingRateUnit</b> &lt;Configured chargingRateUnit&gt;</p>	
	<p><b>Post scenario validations:</b> N/a</p>	

Table 132. Test Case Id: TC\_K\_48\_CSMS

Test case name	Set / Update External Charging Limit (not on a transaction)	
Test case Id	TC_K_48_CSMS	
Use case Id(s)	K12	
Requirement(s)	N/a	
System under test	CSMS	
Description	A charging schedule or charging limit can be imposed by an external system on the Charging Station for new transactions or on the grid connection. An External Control System sends a charging limit to a Charging Station. This limit is then sent to the CSMS.	
Purpose	To verify if the CSMS is able to receive the request from a charging station and respond correctly as described at the OCPP specification.	
Prerequisite(s)	n/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>NotifyChargingLimitRequest</b> with - <b>chargingLimit.chargingLimitSource</b> <i>EMS</i>	2. The CSMS responds with a <b>NotifyChargingLimitResponse</b>
Tool validations	- N/a	
	<b>Post scenario validations:</b> - N/a	

Table 133. Test Case Id: TC\_K\_50\_CSMS

<b>Test case name</b>	<b>Reset / release external charging limit - Without ongoing transaction</b>	
<b>Test case Id</b>	TC_K_50_CSMS	
<b>Use case Id(s)</b>	K13	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	A charging schedule or charging limit can be removed by an external system on the Charging Station. An external control system sends a signal to release a previously imposed charging limit to a Charging Station. The Charging Station notifies the CSMS about this.	
<b>Purpose</b>	To verify if the CSMS is able to receive the notify from a charging station and respond correctly as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>ClearedChargingLimitRequest</b> with - <b>chargingLimitSource</b> <i>EMS</i>	2. The CSMS responds with a <b>ClearedChargingLimitResponse</b>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	



Table 134. Test Case Id: TC\_K\_51\_CSMS

<b>Test case name</b>	<b>Reset / release external charging limit - With ongoing transaction</b>	
<b>Test case Id</b>	TC_K_51_CSMS	
<b>Use case Id(s)</b>	K13	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	A charging schedule or charging limit can be removed by an external system on the Charging Station. An external control system sends a signal to release a previously imposed charging limit to a Charging Station. The Charging Station notifies the CSMS about this.	
<b>Purpose</b>	To verify if the CSMS is able to receive the notify from a charging station and respond correctly as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>ClearedChargingLimitRequest</b> with - <b>chargingLimitSource</b> <i>EMS</i>	2. The CSMS responds with a <b>ClearedChargingLimitResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with - <b>eventType</b> <i>Updated</i> - <b>triggerReason</b> <i>ChargingRateChanged</i>	4. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	- N/a	
	<b>Post scenario validations:</b> - N/a	

Table 135. Test Case Id: TC\_K\_52\_CSMS

Test case name	Set / Update External Charging Limit (not on a transaction) - ChargingStationExternalConstraints in report	
Test case Id	TC_K_52_CSMS	
Use case Id(s)	K12	
Requirement(s)	N/a	
System under test	CSMS	
Description	A charging schedule or charging limit can be removed by an external system on the Charging Station. An external control system sends a signal to release a previously imposed charging limit to a Charging Station. The Charging Station notifies the CSMS about this.	
Purpose	To verify if the CSMS is able to correctly receive the report when a charging limit has been externally changed in a charging station as described at the OCPP specification.	
Prerequisite(s)	n/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetChargingProfilesResponse</b> with - <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>GetChargingProfilesRequest</b>
	3. The OCTT sends a <b>ReportChargingProfilesRequest</b> with - <b>requestId</b> <i>Generated Id</i> - <b>chargingProfile.chargingProfilePurpose</b> <i>ChargingStationExternalConstraints</i>	4. The CSMS responds with a <b>ReportChargingProfilesResponse</b>
Tool validations	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 136. Test Case Id: TC\_K\_70\_CSMS

<b>Test case name</b>	<b>Set charging profile - 2 Profiles</b>	
<b>Test case Id</b>	TC_K_70_CSMS	
<b>Use case Id(s)</b>	n/a	
<b>Requirement(s)</b>	n/a	
<b>System under test</b>	CSMS	
<b>Description</b>	To enable the CSMS to influence the charging power or current drawn from a specific EVSE or the entire Charging Station over a period of time. The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by EVs. The CSMS calculates a ChargingSchedule to stay within certain limits, which MAY be imposed by any external system.	
<b>Purpose</b>	To verify if the CSMS is able to set a charging profile with the same ProfileKind, Purpose, and limit, but with a different stackLevel.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetChargingProfileRequest</b> with <b>stackLevel</b> <Configured stackLevel1>
	4. The OCTT responds with a <b>SetChargingProfileResponse</b> with <b>status Accepted</b>	3. The CSMS sends a <b>SetChargingProfileRequest</b> with <b>stackLevel</b> <Configured stackLevel2>
<b>Tool validations</b>	* Step 3: Message <b>SetChargingProfileRequest</b> <b>chargingProfile.id</b> <different id for both chargingProfiles> <b>chargingProfile.stackLevel</b> <different stackLevel for both chargingProfiles>	
	<b>Post scenario validations:</b> - N/a	

## 13. L Firmware Management

Table 137. Test Case Id: TC\_L\_01\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - Installation successful</b>
<b>Test case Id</b>	TC_L_01_CSMS
<b>Use case Id(s)</b>	L01
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11,L01.FR.15
<b>System under test</b>	CSMS
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an UpdateFirmwareRequest with a signingCertificate.
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to securely download and install a new firmware.
<b>Prerequisite(s)</b>	N/a
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a
	<b>Memory State:</b> N/a
	<b>Reusable State(s):</b> N/a

Test case name	Secure Firmware Update - Installation successful	
Main (Test scenario)	Charging Station	CSMS
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloading</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloaded</i>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>SignatureVerified</i>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installing</i>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallRebooting</i>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	13. The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	14. The CSMS responds with a <b>BootNotificationResponse</b>
	15. The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Available"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	16. The CSMS responds accordingly.
	17. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installed</i>	18. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
Tool validations	* Step 1: Message <b>UpdateFirmwareRequest</b> - <b>firmware.signingCertificate</b> <i>&lt;Configured signingCertificate&gt;</i> - <b>firmware.signature</b> <i>&lt;Configured signature&gt;</i> * Step 14: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 138. Test Case Id: TC\_L\_02\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - InstallScheduled</b>	
<b>Test case Id</b>	TC_L_02_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11,L01.FR.15	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <code>UpdateFirmwareRequest</code> with a <code>signingCertificate</code> .	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to securely download a new firmware and install it	
<b>Prerequisite(s)</b>	The CSMS configuration firmware <code>installDateTime</code> needs to be set to a future <code>dateTime</code> .	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloading</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloaded</i>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>SignatureVerified</i>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallScheduled</i>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installing</i>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	<u>Note(s):</u> - This step will be executed after the given <code>installDateTime</code> from step 1 has been reached.	
	13. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallRebooting</i>	14. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .

Test case name	Secure Firmware Update - InstallScheduled	
	<b>15.</b> The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	<b>16.</b> The CSMS responds with a <b>BootNotificationResponse</b>
	<b>17.</b> The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Available"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	<b>18.</b> The CSMS responds accordingly.
	<b>19.</b> The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installed</i>	<b>20.</b> The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
Tool validations	* Step 1: Message <b>UpdateFirmwareRequest</b> - <b>firmware.installDateTime</b> <i>&lt;A dateTime in the future&gt;</i> * Step 16: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 139. Test Case Id: TC\_L\_03\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - DownloadScheduled</b>	
<b>Test case Id</b>	TC_L_03_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11,L01.FR.15	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <b>UpdateFirmwareRequest</b> with a <b>signingCertificate</b> .	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to schedule securely downloading a new firmware.	
<b>Prerequisite(s)</b>	The CSMS configuration <b>firmware retrieveDateTime</b> needs to be set to a future <b>dateTime</b> .	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status DownloadScheduled</b>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloading</b>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	<u>Note(s):</u> - This step will be executed after the given <b>retrieveDateTime</b> from step 1 has been reached.	
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloaded</b>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status SignatureVerified</b>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Installing</b>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	13. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status InstallRebooting</b>	14. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .



Test case name	Secure Firmware Update - DownloadScheduled	
	<b>15.</b> The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	<b>16.</b> The CSMS responds with a <b>BootNotificationResponse</b>
	<b>17.</b> The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Available"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	<b>18.</b> The CSMS responds accordingly.
	<b>19.</b> The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installed</i>	<b>20.</b> The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
Tool validations	* Step 1: Message <b>UpdateFirmwareRequest</b> - <b>firmware.retrieveDateTime</b> <i>&lt;A dateTime in the future&gt;</i> * Step 16: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

Table 140. Test Case Id: TC\_L\_04\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - RevokedCertificate</b>	
<b>Test case Id</b>	TC_L_04_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <b>UpdateFirmwareRequest</b> with a <b>signingCertificate</b> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting the firmware signing certificate is revoked.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>RevokedCertificate</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 141. Test Case Id: TC\_L\_05\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - InvalidCertificate</b>	
<b>Test case Id</b>	TC_L_05_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <code>UpdateFirmwareRequest</code> with a <code>signingCertificate</code> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting the firmware signing certificate is invalid.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <code>UpdateFirmwareResponse</code> With <code>status InvalidCertificate</code>	1. The CSMS sends a <code>UpdateFirmwareRequest</code>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 142. Test Case Id: TC\_L\_06\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - InvalidSignature</b>	
<b>Test case Id</b>	TC_L_06_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <code>UpdateFirmwareRequest</code> with a <code>signingCertificate</code> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting the signature is invalid.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloading</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloaded</i>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InvalidSignature</i>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 143. Test Case Id: TC\_L\_07\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - DownloadFailed</b>	
<b>Test case Id</b>	TC_L_07_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <b>UpdateFirmwareRequest</b> with a <b>signingCertificate</b> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting it failed to download the firmware.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloading</b>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status DownloadFailed</b>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 144. Test Case Id: TC\_L\_08\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - InstallVerificationFailed</b>	
<b>Test case Id</b>	TC_L_08_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <b>UpdateFirmwareRequest</b> with a <b>signingCertificate</b> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting the verification of the firmware failed during installation.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloading</b>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloaded</b>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status SignatureVerified</b>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Installing</b>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status InstallVerificationFailed</b>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 145. Test Case Id: TC\_L\_09\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - InstallationFailed</b>	
<b>Test case Id</b>	TC_L_09_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <b>UpdateFirmwareRequest</b> with a <b>signingCertificate</b> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting the installation of the firmware failed.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloading</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloaded</i>	6. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>SignatureVerified</i>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installing</i>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallRebooting</i>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	13. The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	14. The CSMS responds with a <b>BootNotificationResponse</b>
	15. The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Available"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	16. The CSMS responds accordingly.
	17. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallationFailed</i>	18. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .

Test case name	Secure Firmware Update - InstallationFailed
Tool validations	* Step 14: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>
	<b>Post scenario validations:</b> N/a

DRAFT



Table 146. Test Case Id: TC\_L\_10\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - AcceptedCanceled</b>	
<b>Test case Id</b>	TC_L_10_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11,L01.FR.24	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <b>UpdateFirmwareRequest</b> with a <b>signingCertificate</b> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting an ongoing installation of a firmware was canceled and it is now starting the new firmware update.	
<b>Prerequisite(s)</b>	The CSMS is able to request a new firmware update, while there is already one ongoing on the Charging Station.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status Accepted</b>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloading</b>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	6. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status AcceptedCanceled</b>	5. The CSMS sends a <b>UpdateFirmwareRequest</b>
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloading</b>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Downloaded</b>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status SignatureVerified</b>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	13. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status Installing</b>	14. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	15. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status InstallRebooting</b>	16. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .

Test case name	Secure Firmware Update - AcceptedCanceled	
	17. The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	18. The CSMS responds with a <b>BootNotificationResponse</b>
	19. The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Available"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	20. The CSMS responds accordingly.
	21. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installed</i>	22. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
Tool validations	* Step 18: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>	
	Post scenario validations: N/a	

Table 147. Test Case Id: TC\_L\_11\_CSMS

<b>Test case name</b>	<b>Secure Firmware Update - Unable to cancel</b>	
<b>Test case Id</b>	TC_L_11_CSMS	
<b>Use case Id(s)</b>	L01	
<b>Requirement(s)</b>	L01.FR.01,L01.FR.11,L01.FR.27	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an <code>UpdateFirmwareRequest</code> with a <code>signingCertificate</code> .	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting the ongoing installation of a firmware cannot be canceled.	
<b>Prerequisite(s)</b>	The CSMS is able to request a new firmware update, while there is already one ongoing on the Charging Station.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloading</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	6. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Rejected</i>	5. The CSMS sends a <b>UpdateFirmwareRequest</b>
	7. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloaded</i>	8. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	9. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>SignatureVerified</i>	10. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	11. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installing</i>	12. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	13. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallRebooting</i>	14. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	15. The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	16. The CSMS responds with a <b>BootNotificationResponse</b>

Test case name	Secure Firmware Update - Unable to cancel	
	<p>17. The OCTT notifies the CSMS about the current state of all connectors.</p> <p>Message: <b>StatusNotificationRequest</b>  <b>connectorStatus</b> <i>Available</i>  Message: <b>NotifyEventRequest</b>  <b>trigger</b> <i>Delta</i>  <b>actualValue</b> <i>"Available"</i>  <b>component.name</b> <i>"Connector"</i>  <b>variable.name</b> <i>"AvailabilityState"</i></p>	<p>18. The CSMS responds accordingly.</p>
	<p>19. The OCTT sends a <b>FirmwareStatusNotificationRequest</b>.  With <b>status</b> <i>Installed</i></p>	<p>20. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b>.</p>
Tool validations	<p>* Step 16:  Message <b>BootNotificationResponse</b>  - <b>status</b> <i>Accepted</i></p>	
	<p><b>Post scenario validations:</b>  N/a</p>	

Table 148. Test Case Id: TC\_L\_13\_CSMS

Test case name	Secure Firmware Update - Unable to download/install firmware with ongoing transaction - AllowNewSessionsPendingFirmwareUpdate is false	
Test case Id	TC_L_13_CSMS	
Use case Id(s)	L01	
Requirement(s)	L01.FR.01,L01.FR.11	
System under test	CSMS	
Description	The CSMS is able to request the Charging Station to securely download and install a new firmware by sending an UpdateFirmwareRequest with a signingCertificate.	
Purpose	To verify if the CSMS is able to handle a Charging Station setting connectors to Unavailable while preparing a firmware update when there is a transaction ongoing.	
Prerequisite(s)	The CSMS is able to request a new firmware update when there is a transaction ongoing on the Charging Station.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> State is <i>EnergyTransferStarted</i>	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>UpdateFirmwareResponse</b> With <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>UpdateFirmwareRequest</b>
	3. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>DownloadScheduled</i>	4. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	5. The OCTT notifies the CSMS about the state change of all connectors that don't have a running transaction.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Unavailable</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Unavailable"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	6. The CSMS responds accordingly.
	7. Execute <b>Reusable State</b> <i>StopAuthorized</i> <u>Note(s)</u> Wait <configured transaction duration> before executing this step	
	8. Execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
	9. Execute <b>Reusable State</b> <i>EVDisconnected</i>	
	10. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloading</i>  <u>Note(s)</u> : - This step will be executed after the given retrieveDateTime from step 1 has been reached.	11. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .

Test case name	Secure Firmware Update - Unable to download/install firmware with ongoing transaction - AllowNewSessionsPendingFirmwareUpdate is false	
	12. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Downloaded</i>	13. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	14. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>SignatureVerified</i>	15. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	16. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installing</i>	17. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	18. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>InstallRebooting</i>	19. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
	20. The OCTT sends a <b>BootNotificationRequest</b> With <b>reason</b> <i>FirmwareUpdate</i>	21. The CSMS responds with a <b>BootNotificationResponse</b>
	22. The OCTT notifies the CSMS about the current state of all connectors.  Message: <b>StatusNotificationRequest</b> <b>connectorStatus</b> <i>Available</i> Message: <b>NotifyEventRequest</b> <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Available"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	23. The CSMS responds accordingly.
	24. The OCTT sends a <b>FirmwareStatusNotificationRequest</b> . With <b>status</b> <i>Installed</i>	25. The CSMS responds with a <b>FirmwareStatusNotificationResponse</b> .
Tool validations	* Step 1: Message <b>UpdateFirmwareRequest</b> - <b>firmware.signingCertificate</b> <i>&lt;configured signingCertificate&gt;</i> * Step 19: Message <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>	
	<b>Post scenario validations:</b> N/a	

## 14. M ISO IEC 15118 CertificateManagement

Table 149. Test Case Id: TC\_M\_01\_CSMS

Test case name	Install CA certificate - CSMSRootCertificate	
Test case Id	TC_M_01_CSMS	
Use case Id(s)	M05	
Requirement(s)	M05.FR.01	
System under test	CSMS	
Description	The CSMS is able to request the Charging Station to install new Root CA certificates using the InstallCertificateRequest message.	
Purpose	To verify if the CSMS is able to request a Charging Station to install a new CSMSRootCertificate.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>CertificateInstalled</i> for certificateType CSMSRootCertificate	
Tool validations	N.a	
	<b>Post scenario validations:</b> N/a	

Table 150. Test Case Id: TC\_M\_02\_CSMS

<b>Test case name</b>	<b>Install CA certificate - ManufacturerRootCertificate</b>	
<b>Test case Id</b>	TC_M_02_CSMS	
<b>Use case Id(s)</b>	M05	
<b>Requirement(s)</b>	M05.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to install new Root CA certificates using the InstallCertificateRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to request a Charging Station to install a new ManufacturerRootCertificate.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>CertificateInstalled</i> for certificateType <i>ManufacturerRootCertificate</i>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	



Table 151. Test Case Id: TC\_M\_05\_CSMS

<b>Test case name</b>	<b>Install CA certificate - Failed</b>	
<b>Test case Id</b>	TC_M_05_CSMS	
<b>Use case Id(s)</b>	M05	
<b>Requirement(s)</b>	M05.FR.01,M05.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to install new Root CA certificates using the InstallCertificateRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to handle a Charging Station reporting it failed to install the requested certificate.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>Manual Action:</b> Trigger the CSMS to send an InstallCertificateRequest with certificateType CSMSRootCertificate.	
	2. The OCTT responds with a <b>InstallCertificateResponse</b> With <b>status</b> is <i>Failed</i>	1. The CSMS sends a <b>InstallCertificateRequest</b>

Table 152. Test Case Id: TC\_M\_12\_CSMS

<b>Test case name</b>	<b>Retrieve certificates from Charging Station - CSMSRootCertificate</b>	
<b>Test case Id</b>	TC_M_12_CSMS	
<b>Use case Id(s)</b>	M03	
<b>Requirement(s)</b>	M03.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to retrieve the certificates installed at the Charging Station using the GetInstalledCertificateIdsRequest message. It supports all available hash algorithms, including SHA256, SHA384, and SHA512.	
<b>Purpose</b>	To verify if the CSMS is able to retrieve the hashData from all CSMSRootCertificates stored at the Charging Station, using all available hash algorithms, including SHA256, SHA384, and SHA512.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <a href="#">GetInstalledCertificates</a> for certificateType <i>CSMSRootCertificate</i> . The OCTT responds with data hashed with SHA256.	
	2. Execute <b>Reusable State</b> <a href="#">GetInstalledCertificates</a> for certificateType <i>CSMSRootCertificate</i> . The OCTT responds with data hashed with SHA384.	
	3. Execute <b>Reusable State</b> <a href="#">GetInstalledCertificates</a> for certificateType <i>CSMSRootCertificate</i> . The OCTT responds with data hashed with SHA512.	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 153. Test Case Id: TC\_M\_13\_CSMS

<b>Test case name</b>	<b>Retrieve certificates from Charging Station - ManufacturerRootCertificate</b>	
<b>Test case Id</b>	TC_M_13_CSMS	
<b>Use case Id(s)</b>	M03	
<b>Requirement(s)</b>	M03.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to retrieve the certificates installed at the Charging Station using the GetInstalledCertificateIdsRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to retrieve the hashData from all ManufacturerRootCertificate stored at the Charging Station.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. Execute <b>Reusable State</b> <i>GetInstalledCertificates</i> for certificateType <i>ManufacturerRootCertificate</i>	
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

Table 154. Test Case Id: TC\_M\_18\_CSMS

<b>Test case name</b>	<b>Retrieve certificates from Charging Station - All certificateTypes</b>	
<b>Test case Id</b>	TC_M_18_CSMS	
<b>Use case Id(s)</b>	M03	
<b>Requirement(s)</b>	M03.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to retrieve the certificates installed at the Charging Station using the GetInstalledCertificateIdsRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to retrieve the hashData from all Root CA and V2GCertificateChain certificates stored at the Charging Station.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to send a GetInstalledCertificateIdsRequest without certificateType.	
	2. The OCTT responds with a <b>GetInstalledCertificateIdsResponse</b> With <b>status</b> is <i>Accepted</i> <b>certificateHashDataChain</b> contains <The hashData of all certificates stored at the OCTT truststore>	1. The CSMS sends a <b>GetInstalledCertificateIdsRequest</b>
<b>Tool validations</b>	* Step 1: Message: <b>GetInstalledCertificateIdsRequest</b> - <b>certificateType</b> is omitted	
	<b>Post scenario validations:</b> N/a	

Table 155. Test Case Id: TC\_M\_19\_CSMS

<b>Test case name</b>	<b>Retrieve certificates from Charging Station - No matching certificate found</b>	
<b>Test case Id</b>	TC_M_19_CSMS	
<b>Use case Id(s)</b>	M03	
<b>Requirement(s)</b>	M03.FR.01,M03.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to retrieve the certificates installed at the Charging Station using the GetInstalledCertificateIdsRequest message.	
<b>Purpose</b>	To verify if the CSMS is able to handle a response from the Charging Station indicating it was not able to find a certificate for the requested criteria.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to send a GetInstalledCertificateIdsRequest with certificateType ManufacturerRootCertificate.	
	2. The OCTT responds with a <b>GetInstalledCertificateIdsResponse</b> With <b>status</b> is <i>NotFound</i> <b>certificateHashDataChain</b> is omitted.	1. The CSMS sends a <b>GetInstalledCertificateIdsRequest</b>
<b>Tool validations</b>	* Step 1: Message: <b>GetInstalledCertificateIdsRequest</b> - <b>certificateType</b> is <i>ManufacturerRootCertificate</i>	
	<b>Post scenario validations:</b> N/a	

Table 156. Test Case Id: TC\_M\_20\_CSMS

Test case name	Delete a certificate from a Charging Station - Success	
Test case Id	TC_M_20_CSMS	
Use case Id(s)	M04	
Requirement(s)	M04.FR.01,M04.FR.07	
System under test	CSMS	
Description	The CSMS is able to request the Charging Station to delete an installed certificate using the DeleteCertificateRequest message, using all available hash algorithms, including SHA256, SHA384, and SHA512.	
Purpose	To verify if CSMS is able to request a Charging Station to delete an installed certificate, using all available hash algorithms, including SHA256, SHA384, and SHA512.	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. <i>CertificateInstalled</i> with certificateType <i>CSMSRootCertificate</i> .	
	<u>Manual Action:</u> Request the CSMS to send a DeleteCertificateRequest.	
	3. The OCTT responds with a <b>GetInstalledCertificateIdsResponse</b> With <b>status</b> is <i>Accepted</i> <b>certificateHashDataChain</b> contains an entry with following values: <b>certificateHashDataChain[0].certificateType</b> is <i>CSMSRootCertificate</i> <b>certificateHashDataChain[0].certificateHashData.hashAlgorithm</b> is <i>SHA256</i>	2. The CSMS sends a <b>GetInstalledCertificateIdsRequest</b>
	5. The OCTT responds with a <b>DeleteCertificateResponse</b> With <b>status</b> is <i>Accepted</i>	4. The CSMS sends a <b>DeleteCertificateRequest</b>
	<u>Note(s):</u> - Steps 1 - 5 will be repeated for each hash algorithm (SHA256, SHA384, SHA512).	
Tool validations	* Step 2: Message: <b>GetInstalledCertificateIdsRequest</b> - <b>certificateType</b> contains <i>CSMSRootCertificate</i> OR is omitted.	
	* Step 4: Message: <b>DeleteCertificateRequest</b> - <b>certificateHashData</b> is <Returned certificateHashData at Step 3>.	
	<b>Post scenario validations:</b> N/a	

Table 157. Test Case Id: TC\_M\_21\_CSMS

<b>Test case name</b>	<b>Delete a certificate from a Charging Station - Failed</b>	
<b>Test case Id</b>	TC_M_21_CSMS	
<b>Use case Id(s)</b>	M04	
<b>Requirement(s)</b>	M04.FR.01,M04.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS is able to request the Charging Station to delete an installed certificate using the DeleteCertificateRequest message.	
<b>Purpose</b>	To verify if CSMS is able to handle a Charging Station that fails to delete an installed certificate.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> <i>CertificateInstalled</i> with certificateType CSMSRootCertificate.	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a DeleteCertificateRequest.	
	<p>2. The OCTT responds with a <b>GetInstalledCertificateIdsResponse</b> With <b>status</b> is <i>Accepted</i> <b>certificateHashDataChain</b> contains an entry with following values: <b>certificateHashDataChain[0].certificateType</b> is <i>CSMSRootCertificate</i> <b>certificateHashDataChain[0].certificateHashData.hashAlgorithm</b> is <i>SHA256</i></p>	<p>1. The CSMS sends a <b>GetInstalledCertificateIdsRequest</b></p>
	<p>4. The OCTT responds with a <b>DeleteCertificateResponse</b> With <b>status</b> is <i>Failed</i></p>	<p>3. The CSMS sends a <b>DeleteCertificateRequest</b></p>
<b>Tool validations</b>	<p>* Step 1: Message: <b>GetInstalledCertificateIdsRequest</b> - <b>certificateType</b> contains <i>CSMSRootCertificate</i> OR is <i>omitted</i>.</p> <p>* Step 3: Message: <b>DeleteCertificateRequest</b> - <b>certificateHashData</b> contains <i>&lt;Returned certificateHashData at Step 2&gt;</i>.</p> <p><b>Post scenario validations:</b> N/a</p>	

## 15. N Diagnostics

Table 158. Test Case Id: TC\_N\_01\_CSMS

<b>Test case name</b>	<b>Get Monitoring Report - with component criteria</b>	
<b>Test case Id</b>	TC_N_01_CSMS	
<b>Use case Id(s)</b>	N02	
<b>Requirement(s)</b>	N02.FR.05, N02.FR.10	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a report of monitors that match the component criteria.	
<b>Purpose</b>	To test that CSMS supports requesting a monitoring report for the component criteria and that it handles an empty result set.	
<b>Prerequisite(s)</b>	CS has implemented device model monitoring and MonitoringCtrlr.Enabled = true.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> CSMS requests ClearVariableMonitoring ItemsPerMessage from CS.	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	Manually instruct CSMS to get a report of monitors for: - all <i>DeltaMonitoring</i>	
	2. OCTT responds with: <b>GetMonitoringReportResponse</b> with: <b>Status</b> <i>EmptyResultSet</i>	1. CSMS sends <b>GetMonitoringReportRequest</b>
	Manually instruct CSMS to get a report of monitors for: - all <i>ThresholdMonitoring</i>	
	4. OCTT responds with: <b>GetMonitoringReportResponse</b> with: <b>Status</b> <i>Accepted</i>	3. CSMS sends <b>GetMonitoringReportRequest</b>
	5. OCTT responds with: <b>NotifyMonitoringReportRequest</b>	6. CSMS sends <b>NotifyMonitoringReportResponse</b>
	Step 5 and 6 are repeated as often as needed to report all configuration variables.	
<b>Tool validations</b>	* Step 1: Message: <b>GetMonitoringReportRequest</b> - <b>monitoringCriteria</b> = <i>DeltaMonitoring</i>	
	* Step 3: Message: <b>GetMonitoringReportRequest</b> - <b>monitoringCriteria</b> = <i>ThresholdMonitoring</i>	
	<b>Post scenario validations:</b> Check that CSMS shows the <i>Threshold</i> monitors.	



Table 159. Test Case Id: TC\_N\_02\_CSMS

<b>Test case name</b>	<b>Get Monitoring Report - with component/variable</b>	
<b>Test case Id</b>	TC_N_02_CSMS	
<b>Use case Id(s)</b>	N02	
<b>Requirement(s)</b>	N02.FR.05, N02.FR.10	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a report of monitors that match the the given list of components and variables.	
<b>Purpose</b>	To test that CSMS supports requesting a monitoring report for a given component and variable and that it handles an empty result set.	
<b>Prerequisite(s)</b>	CS has implemented device model monitoring and MonitoringCtrlr.Enabled = true.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> CSMS requests ClearVariableMonitoring ItemsPerMessage from CS.	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<i>Manually instruct CSMS to get a report of monitors for:</i> - the variable Power of ChargingStation	
	2. OCTT responds with: <b>GetMonitoringReportResponse</b> with: <b>Status EmptyResultSet</b>	1. CSMS sends <b>GetMonitoringReportRequest</b>
	<i>Manually instruct CSMS to get a report of monitors for:</i> - the variable AvailabilityState of EVSE #1.	
	4. OCTT responds with: <b>GetMonitoringReportResponse</b> with: <b>Status Accepted</b>	3. CSMS sends <b>GetMonitoringReportRequest</b>
	5. OCTT responds with: <b>NotifyMonitoringReportRequest</b>	6. CSMS sends <b>NotifyMonitoringReportResponse</b>
	<i>Step 5 and 6 are repeated as often as needed to report all configuration variables.</i>	
<b>Tool validations</b>	* Step 1: Message: <b>GetMonitoringReportRequest</b> - <b>componentVariable[0].component.name</b> = "ChargingStation" - <b>componentVariable[0].variable.name</b> = "Power"	
	* Step 3: Message: <b>GetMonitoringReportRequest</b> - <b>componentVariable[1].component.name</b> = "EVSE" - <b>componentVariable[1].component.evse.id</b> = 1 - <b>componentVariable[1].variable.name</b> = "AvailabilityState"	
	<b>Post scenario validations:</b> Check that CSMS shows the monitor for AvailabilityState for EVSE #1.	

Table 160. Test Case Id: TC\_N\_03\_CSMS

<b>Test case name</b>	<b>Get Monitoring Report - with component criteria and component/variable</b>	
<b>Test case Id</b>	TC_N_03_CSMS	
<b>Use case Id(s)</b>	N02	
<b>Requirement(s)</b>	N02.FR.05, N02.FR.10	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS requests a report of monitors that match both the component criteria and the given list of components and variables.	
<b>Purpose</b>	To test that CSMS supports requesting a monitoring report for both the component criteria and a given component and variable and that it handles an empty result set.	
<b>Prerequisite(s)</b>	CS has implemented device model monitoring and MonitoringCtrlr.Enabled = true.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> CSMS requests ClearVariableMonitoring ItemsPerMessage from CS.	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	Manually instruct CSMS to get a report of monitors for: - all <i>DeltaMonitoring</i> - and the variable <i>AvailabilityState</i> for EVSE #1.	
	2. OCTT responds with: <b>GetMonitoringReportResponse</b> with: <b>Status</b> <i>EmptyResultSet</i>	1. CSMS sends <b>GetMonitoringReportRequest</b>
	Manually instruct CSMS to get a report of monitors for: - all <i>ThresholdMonitoring</i> - and the variable <i>Power</i> of <i>ChargingStation</i> .	
	4. OCTT responds with: <b>GetMonitoringReportResponse</b> with: <b>Status</b> <i>Accepted</i>	3. CSMS sends <b>GetMonitoringReportRequest</b>
	5. OCTT responds with: <b>NotifyMonitoringReportRequest</b>	6. CSMS sends <b>NotifyMonitoringReportResponse</b>
	Step 5 and 6 are repeated as often as needed to report all configuration variables.	
<b>Tool validations</b>	* Step 1: Message: <b>GetMonitoringReportRequest</b> - <b>monitoringCriteria</b> = <i>DeltaMonitoring</i> - <b>componentVariable[0].component.name</b> = "EVSE" - <b>componentVariable[0].component.evse.id</b> = <configured evseld> - <b>componentVariable[0].variable.name</b> = "AvailabilityState" * Step 3: Message: <b>GetMonitoringReportRequest</b> - <b>monitoringCriteria</b> = <i>ThresholdMonitoring</i> - <b>componentVariable[0].component.name</b> = "ChargingStation" - <b>componentVariable[0].variable.name</b> = "Power"	
	<b>Post scenario validations:</b> Check that CSMS shows the <i>Threshold</i> monitors for <i>Power</i> for <i>ChargingStation</i> .	

Table 161. Test Case Id: TC\_N\_05\_CSMS

<b>Test case name</b>	<b>Set Monitoring Base – success</b>	
<b>Test case Id</b>	TC_N_05_CSMS	
<b>Use case Id(s)</b>	N03	
<b>Requirement(s)</b>	N03.FR.03, N03.FR.04, N03.FR.05	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS sends a SetMonitoringBaseRequest for <i>All</i> , <i>FactoryDefault</i> and <i>HardWiredOnly</i> .	
<b>Purpose</b>	To test that CSMS supports all three monitoring base types.	
<b>Prerequisite(s)</b>	CS has implemented device model monitoring and MonitoringCtrlr.Enabled = true.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. OCTT responds with: <b>SetMonitoringBaseResponse</b>	<i>Instruct CSMS to set a monitoring base of _All._</i> <b>1. CSMS sends SetMonitoringBaseRequest</b>
	4. OCTT responds with: <b>SetMonitoringBaseResponse</b>	<i>Instruct CSMS to set a monitoring base of _FactoryDefault._</i> <b>3. OCTT sends SetMonitoringBaseRequest</b>
	6. The OCTT responds with: <b>SetMonitoringBaseResponse</b>	<i>Instruct CSMS to set a monitoring base of _HardWiredOnly._</i> <b>5. OCTT sends SetMonitoringBaseRequest</b>
<b>Tool validations</b>	* Step 1 Message: <b>SetMonitoringBaseRequest</b> - <b>monitoringBase</b> = <i>All</i>	
	* Step 3 Message: <b>SetMonitoringBaseRequest</b> - <b>monitoringBase</b> = <i>FactoryDefault</i>	
	* Step 6 Message: <b>SetMonitoringBaseRequest</b> - <b>monitoringBase</b> = <i>HardWiredOnly</i>	
	<b>Post scenario validations:</b> N/A	

Table 162. Test Case Id: TC\_N\_16\_CSMS

<b>Test case name</b>	<b>Set Monitoring Level – Success</b>	
<b>Test case Id</b>	TC_N_16_CSMS	
<b>Use case Id(s)</b>	N05	
<b>Requirement(s)</b>	N05.FR.01	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS sets a monitoring level.	
<b>Purpose</b>	To test that CSMS supports setting of a monitoring level.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	2. OCTT responds with: <b>SetMonitoringLevelResponse</b> with <b>Status</b> is <i>Accepted</i>	1. <i>Instruct CSMS to set a monitoring level with <b>severity</b> = _4</i>
<b>Tool validations</b>	* Step 1: Message: <b>SetMonitoringLevelRequest</b> with: <b>severity</b> = 4	
	<b>Post scenario validations:</b> N/A	

Table 163. Test Case Id: TC\_N\_17\_CSMS

<b>Test case name</b>	<b>Set Monitoring Level – Out of range</b>	
<b>Test case Id</b>	TC_N_17_CSMS	
<b>Use case Id(s)</b>	N05	
<b>Requirement(s)</b>	N05.FR.02	
<b>System under test</b>	CSMS	
<b>Description</b>	CSMS sets a monitoring level.	
<b>Purpose</b>	To test that CSMS supports the rejection of setting of a monitoring level.	
<b>Prerequisite(s)</b>	The OCTT will always reject the message, but normally this would only occur if the set severity level is out of range.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	2. OCTT responds with: <b>SetMonitoringLevelResponse</b> with <b>Status</b> is <i>Rejected</i>	1. Instruct CSMS to set a monitoring level with <b>severity</b> = _4

Table 164. Test Case Id: TC\_N\_20\_CSMS

<b>Test case name</b>	<b>Alert – HardWiredMonitor</b>	
<b>Test case Id</b>	TC_N_20_CSMS	
<b>Use case Id(s)</b>	N07	
<b>Requirement(s)</b>	N07.FR.03	
<b>System under test</b>	CSMS	
<b>Description</b>	Charging Station sends an NotifyEventRequest for a HardWiredMonitor.	
<b>Purpose</b>	To test that the CSMS is able to handle a HardWiredMonitor.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. OCTT sends <b>NotifyEventRequest</b> message with <b>eventNotificationType = HardWiredMonitor</b>	2. CSMS returns <b>NotifyEventResponse</b> message.
<b>Tool validations</b>	* Step 2: Message: <b>NotifyEventResponse</b> with empty body.	
	<b>Post scenario validations:</b> N/A	

Table 165. Test Case Id: TC\_N\_24\_CSMS

Test case name	Periodic events	
Test case Id	TC_N_24_CSMS	
Use case Id(s)	N08	
Requirement(s)	N08.FR.02	
System under test	CSMS	
Description	Charging Station sends a periodic NotifyEventRequest.	
Purpose	To test that CSMS returns a NotifyEventResponse. <i>Note: this is identical to TC_N_20_CSMS, only with a periodic event.</i>	
Prerequisite(s)	N/a	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	Tester makes OCTT send a NotifyEventRequest message.	
	1. OCTT sends <b>NotifyEventRequest</b> message.	2. CSMS returns <b>NotifyEventResponse</b> message.
	<u>Note(s):</u> - Step 1 and 2 will be repeated <i>n</i> times	
Tool validations	* Step 2: Message: <b>NotifyEventResponse</b> with empty body.	
	<b>Post scenario validations:</b> N/A	

Table 166. Test Case Id: TC\_N\_25\_CSMS

<b>Test case name</b>	<b>Retrieve Log Information - Diagnostics Log - Success</b>	
<b>Test case Id</b>	TC_N_25_CSMS	
<b>Use case Id(s)</b>	N01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers the functionality of getting log information from a Charging Station. The CSMS can request a Charging Station to upload a file with log information to a given location (URL). The format of this log file is not prescribed. The Charging Station successfully uploads a log file and gives information about the status of the upload by sending status notifications to the CSMS.	
<b>Purpose</b>	To verify if the CSMS is able to request a charging station to successfully upload a log as described at the OCPP specification.	
<b>Prerequisite(s)</b>	Charging Station has log information available.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetLogResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>GetLogRequest</b>
	3. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploading</b> - <b>requestId Same Id as the GetLogRequest</b>	4. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
	5. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploaded</b> - <b>requestId Same Id as the GetLogRequest</b>	6. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>GetLogRequest</b> - <b>logType DiagnosticsLog</b>	
	<b>Post scenario validations:</b> - N/a	



Table 167. Test Case Id: TC\_N\_27\_CSMS

<b>Test case name</b>	<b>Get Customer Information - Accepted + data</b>	
<b>Test case Id</b>	TC_N_27_CSMS	
<b>Use case Id(s)</b>	N09	
<b>Requirement(s)</b>	N09.FR.01, N09.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sends a message to the Charging Station to retrieve IdToken customer information, for example to be compliant with local privacy laws. The Charging Station notifies the CSMS by sending one or more reports.	
<b>Purpose</b>	To verify if the CSMS sends the request correctly and responds on the notifies as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>CustomerInformationResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>CustomerInformationRequest</b>
	3. The OCTT sends a <b>NotifyCustomerInformationRequest</b>	4. The CSMS responds with a <b>NotifyCustomerInformationResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>CustomerInformationRequest</b> - <b>report true</b> - <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> - <b>idToken.type</b> <Configured valid_idtoken_type>	
	<b>Post scenario validations:</b> - N/a	

Table 168. Test Case Id: TC\_N\_28\_CSMS

<b>Test case name</b>	<b>Get Customer Information - Accepted + no data</b>	
<b>Test case Id</b>	TC_N_28_CSMS	
<b>Use case Id(s)</b>	N09	
<b>Requirement(s)</b>	N09.FR.01, N09.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sends a message to the Charging Station to retrieve IdToken customer information, for example to be compliant with local privacy laws. The Charging Station notifies the CSMS by sending one or more reports.	
<b>Purpose</b>	To verify if the CSMS sends the request correctly and responds on the notifies as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>CustomerInformationResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>CustomerInformationRequest</b>
	3. The OCTT sends a <b>NotifyCustomerInformationRequest</b>	4. The CSMS responds with a <b>NotifyCustomerInformationResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>CustomerInformationRequest</b> - <b>report true</b> - <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> - <b>idToken.type</b> <Configured valid_idtoken_type>	
	<b>Post scenario validations:</b> - N/a	

Table 169. Test Case Id: TC\_N\_29\_CSMS

<b>Test case name</b>	<b>Get Customer Information - Rejected</b>	
<b>Test case Id</b>	TC_N_29_CSMS	
<b>Use case Id(s)</b>	N09	
<b>Requirement(s)</b>	N09.FR.01, N09.FR.04	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sends a message to the Charging Station to retrieve IdToken customer information, but the Charging Station rejects the request.	
<b>Purpose</b>	To verify if the CSMS sends the request correctly as described at the OCPP specification, and can handle the Charging Station rejecting the request.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>CustomerInformationResponse</b> with <b>status Rejected</b>	1. The CSMS sends a <b>CustomerInformationRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>CustomerInformationRequest</b> - <b>report true</b> - <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> - <b>idToken.type</b> <Configured valid_idtoken_type>	
	<b>Post scenario validations:</b> - N/a	

Table 170. Test Case Id: TC\_N\_62\_CSMS

<b>Test case name</b>	<b>Clear Customer Information - Clear and report - customerIdIdentifier</b>	
<b>Test case Id</b>	TC_N_62_CSMS	
<b>Use case Id(s)</b>	N10	
<b>Requirement(s)</b>	N10.FR.08	
<b>System under test</b>	CSMS	
<b>Description</b>	The CSMS sends a message to the Charging Station to clear (and retrieve) raw customer information, for example to be compliant with local privacy laws. The Charging Station notifies the CSMS by sending one or more reports.	
<b>Purpose</b>	To verify if the CSMS sends the request correctly and responds on the notifies as described at the OCPP specification.	
<b>Prerequisite(s)</b>	The CSMS supports retrieving / deleting CustomerInformation - CustomerIdentifier	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>CustomerInformationResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>CustomerInformationRequest</b>
	3. The OCTT sends a <b>NotifyCustomerInformationRequest</b>	4. The CSMS responds with a <b>NotifyCustomerInformationResponse</b>
<b>Tool validations</b>	* Step 1: Message <b>CustomerInformationRequest</b> - <b>report true</b> - <b>clear true</b> - <b>customerIdIdentifier "OpenChargeAlliance"</b>	
	<b>Post scenario validations:</b> - N/a	

Table 171. Test Case Id: TC\_N\_34\_CSMS

<b>Test case name</b>	<b>Retrieve Log Information - Rejected</b>	
<b>Test case Id</b>	TC_N_34_CSMS	
<b>Use case Id(s)</b>	N01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers the functionality of getting log information from a Charging Station. The CSMS can request a Charging Station to upload a file with log information to a given location (URL). The format of this log file is not prescribed. The Charging Station successfully uploads a log file and gives information about the status of the upload by sending status notifications to the CSMS.	
<b>Purpose</b>	To verify if the CSMS is able to request a charging station to successfully upload a log as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetLogResponse</b> with <b>status Rejected</b>	1. The CSMS sends a <b>GetLogRequest</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 172. Test Case Id: TC\_N\_35\_CSMS

<b>Test case name</b>	<b>Retrieve Log Information - Security Log - Success</b>	
<b>Test case Id</b>	TC_N_35_CSMS	
<b>Use case Id(s)</b>	N01	
<b>Requirement(s)</b>		
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers the functionality of getting log information from a Charging Station. The CSMS can request a Charging Station to upload a file with log information to a given location (URL). The format of this log file is not prescribed. The Charging Station successfully uploads a log file and gives information about the status of the upload by sending status notifications to the CSMS.	
<b>Purpose</b>	To verify if the CSMS is able to request a charging station to successfully upload a log as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> Charging Station has log information available.	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetLogResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>GetLogRequest</b>
	3. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploading</b> - <b>requestId Same Id as the GetLogRequest</b>	4. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
	5. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploaded</b> - <b>requestId Same Id as the GetLogRequest</b>	6. The OCTT responds with a <b>LogStatusNotificationResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>GetLogRequest</b> - <b>logType SecurityLog</b>	
	<b>Post scenario validations:</b> - N/a	

Table 173. Test Case Id: TC\_N\_36\_CSMS

<b>Test case name</b>	<b>Retrieve Log Information - Second Request</b>	
<b>Test case Id</b>	TC_N_36_CSMS	
<b>Use case Id(s)</b>	N01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case covers the functionality of getting log information from a Charging Station. The CSMS can request a Charging Station to upload a file with log information to a given location (URL). The format of this log file is not prescribed. The Charging Station successfully uploads a log file and gives information about the status of the upload by sending status notifications to the CSMS.	
<b>Purpose</b>	To verify if the CSMS is able to request a second request while the charging station is uploading a log as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> Charging Station has log information available.	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
		1. The CSMS sends a <b>GetLogRequest</b>
	2. The OCTT responds with a <b>GetLogResponse</b> with <b>status Accepted</b>	
	3. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploading</b> - <b>requestId Same Id as the GetLogRequest from Step 1</b>	4. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
	6. The OCTT responds with a <b>GetLogResponse</b> with <b>status AcceptedCanceled</b>	5. The CSMS sends a <b>GetLogRequest</b>
	7. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status AcceptedCanceled</b> - <b>requestId Same Id as the GetLogRequest from Step 1</b>	8. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
	9. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploading</b> - <b>requestId Same Id as the GetLogRequest from Step 5</b>	10. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
	11. The OCTT sends a <b>LogStatusNotificationRequest</b> with - <b>status Uploaded</b> - <b>requestId Same Id as the GetLogRequest from Step 5</b>	12. The CSMS responds with a <b>LogStatusNotificationResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 174. Test Case Id: TC\_N\_44\_CSMS

<b>Test case name</b>	<b>Clear / Remove Monitoring - Rejected</b>	
<b>Test case Id</b>	TC_N_44_CSMS	
<b>Use case Id(s)</b>	N06	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	A monitoring setting can be cleared (removed) by sending a <code>ClearVariableMonitoringRequest</code> with the id of the monitoring setting.	
<b>Purpose</b>	To verify if the CSMS is able to correctly read the respond from a charging station on a request to clear a monitor that cannot be cleared as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <code>ClearVariableMonitoringResponse</code> with <code>clearMonitoringResult[0].status</code> <i>Rejected</i>	1. The CSMS sends a <code>ClearVariableMonitoringRequest</code>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	



Table 175. Test Case Id: TC\_N\_47\_CSMS

<b>Test case name</b>	<b>Get Monitoring report - Report all</b>	
<b>Test case Id</b>	TC_N_47_CSMS	
<b>Use case Id(s)</b>	N02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS requests the Charging Station to send a report about configured monitoring settings per component and variable. Optionally, this list can be filtered on monitoringCriteria and componentVariables.	
<b>Purpose</b>	To verify if the CSMS is able to send a get monitor request omitting the monitoringCriteria and componentVariable as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetMonitoringReportResponse</b>	1. The CSMS sends a <b>GetMonitoringReportRequest</b>
	3. The OCTT sends a <b>NotifyMonitoringReportRequest</b>	4. The CSMS responds with a <b>NotifyMonitoringReportResponse</b> .
	<u>Note(s):</u> - If <b>tbc</b> is True at Step 3 then step 3 and 4 will be repeated	
<b>Tool validations</b>	* Step 1: Message <b>GetMonitoringReportRequest</b> - <b>monitoringCriteria</b> omitted AND - <b>componentVariable</b> omitted.	
	<b>Post scenario validations:</b> - N/a	

Table 176. Test Case Id: TC\_N\_48\_CSMS

<b>Test case name</b>	<b>Alert Event - Variable monitoring on write only</b>	
<b>Test case Id</b>	TC_N_48_CSMS	
<b>Use case Id(s)</b>	N07	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	NotifyEventRequest reports every Component/Variable for which a VariableMonitoring setting was triggered. Only the VariableMonitoring settings that are responsible for triggering an event are included.	
<b>Purpose</b>	To verify if the CSMS is able to read a request from a trigger from a variablemonitor which is write only as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>NotifyEventRequest</b> with <b>eventData.actualValue</b> empty	2. The CSMS responds with a <b>NotifyEventResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 177. Test Case Id: TC\_N\_49\_CSMS

<b>Test case name</b>	<b>Alert Event - LowerThreshold/UpperThreshold cleared after reboot</b>	
<b>Test case Id</b>	TC_N_49_CSMS	
<b>Use case Id(s)</b>	N07	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	NotifyEventRequest reports every Component/Variable for which a VariableMonitoring setting was triggered. Only the VariableMonitoring settings that are responsible for triggering an event are included.	
<b>Purpose</b>	To verify if the CSMS is able to read a request when a trigger is cleared after a reboot as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>NotifyEventRequest</b> with <b>eventData.cleared true</b>	2. The CSMS responds with a <b>NotifyEventResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 178. Test Case Id: TC\_N\_50\_CSMS

<b>Test case name</b>	<b>Alert Event - Periodic Triggered</b>	
<b>Test case Id</b>	TC_N_50_CSMS	
<b>Use case Id(s)</b>	N07	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	NotifyEventRequest reports every Component/Variable for which a VariableMonitoring setting was triggered. Only the VariableMonitoring settings that are responsible for triggering an event are included.	
<b>Purpose</b>	To verify if the CSMS is able to read a request when a trigger reason is periodic after a reboot as described at the OCPP specification.	
<b>Prerequisite(s)</b>	n/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>NotifyEventRequest</b> with <b>eventData.trigger</b> <i>Periodic</i>	2. The CSMS responds with a <b>NotifyEventResponse</b> .
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

## 16. 0 Display Message

Table 179. Test Case Id: TC\_O\_01\_CSMS

<b>Test case name</b>	<b>Set Display Message - Success</b>	
<b>Test case Id</b>	TC_O_01_CSMS	
<b>Use case Id(s)</b>	O01	
<b>Requirement(s)</b>	O01_FR_04	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send an SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send the request according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a SetDisplayMessageRequest.	
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id> - <b>message.priority</b> <Configured Priority> - <b>message.message.format</b> <Configured Format>	
	<b>Post scenario validations:</b> - N/a	

Table 180. Test Case Id: TC\_O\_02\_CSMS

<b>Test case name</b>	<b>Get all Display Messages - Success</b>	
<b>Test case Id</b>	TC_O_02_CSMS	
<b>Use case Id(s)</b>	O03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can request all the installed DisplayMessages configured via OCPP in a Charging Station. The Charging Station can remove messages when they are out-dated, or transactions have ended. It can be very useful for a CSO to be able to view to current list of messages, so the CSO knows which messages are (still) configured.	
<b>Purpose</b>	To verify if the CSMS is able to send the request to get the DisplayMessages according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> A display message is configured.	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetDisplayMessagesResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>GetDisplayMessagesRequest</b>
	3. The OCTT sends a <b>NotifyDisplayMessagesRequest</b>	4. The CSMS responds with a <b>NotifyDisplayMessagesResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>GetDisplayMessagesRequest</b> - <b>requestId</b> <Generated Id> - <b>id</b> <Omitted> - <b>priority</b> <Omitted> - <b>state</b> <Omitted>	
	<b>Post scenario validations:</b> - N/a	

Table 181. Test Case Id: TC\_O\_03\_CSMS

<b>Test case name</b>	<b>Get all Display Messages - No DisplayMessages configured</b>	
<b>Test case Id</b>	TC_O_03_CSMS	
<b>Use case Id(s)</b>	O03	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can request all the installed DisplayMessages configured via OCPP in a Charging Station. The Charging Station can remove messages when they are out-dated, or transactions have ended. It can be very useful for a CSO to be able to view to current list of messages, so the CSO knows which messages are (still) configured.	
<b>Purpose</b>	To verify if the CSMS can request to get all display messages according to the DisplayMessage mechanism as described in the OCPP specification when no messages are configured.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetDisplayMessagesResponse</b> with <b>status</b> <i>Unknown</i>	1. The CSMS sends a <b>GetDisplayMessagesRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>GetDisplayMessagesRequest</b> - <b>requestId</b> <Generated request id>	
	<b>Post scenario validations:</b> - N/a	

Table 182. Test Case Id: TC\_O\_04\_CSMS

<b>Test case name</b>	<b>Clear a Display Messages - Success</b>	
<b>Test case Id</b>	TC_O_04_CSMS	
<b>Use case Id(s)</b>	O05	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can remove a specific message, configured via OCPP in a Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to clear a message according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> A display message is configured.	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<i>Note: As a help method, a <b>GetDisplayMessagesRequest</b> is requested first for CSMS's that implemented their <b>ClearDisplayMessage</b> as a combined feature.</i>	
	2. The OCTT responds with a <b>ClearDisplayMessageResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>ClearDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>ClearDisplayMessageRequest</b> - <b>id</b> <Generated Id from set display message>	
	<b>Post scenario validations:</b> - N/a	



Table 183. Test Case Id: TC\_O\_05\_CSMS

<b>Test case name</b>	<b>Clear a Display Messages - Unknown Key</b>	
<b>Test case Id</b>	TC_O_05_CSMS	
<b>Use case Id(s)</b>	O05	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can remove a specific message, configured via OCPP in a Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to request the Charging Station to clear a message according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	If the CSMS supports sending a ClearDisplayMessageRequest with an unknown id.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>ClearDisplayMessageResponse</b> with <b>status Unknown</b>	1. The CSMS sends a <b>ClearDisplayMessageRequest</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> - N/a	

Table 184. Test Case Id: TC\_O\_06\_CSMS

<b>Test case name</b>	<b>Set Display Message - Specific transaction - Success</b>	
<b>Test case Id</b>	TC_O_06_CSMS	
<b>Use case Id(s)</b>	O02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can set a message to be displayed on a Charging Station for a specific transaction. Depending on the given parameters the message shall be displayed a certain way on the Charging Station.	
<b>Purpose</b>	To verify if the CSMS is able to send a display message correctly according the mechanism as described in the OCPP specification for a specific transaction.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a display message for a specific transaction.	
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
	3. Execute <b>Reusable State</b> <i>EVDIsconnected</i>	
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.transactionId</b> Same ID as previously returned by the Charging Station AND - <b>message.priority</b> <Configured Priority>	
	<b>Post scenario validations:</b> - N/a	

Table 185. Test Case Id: TC\_O\_07\_CSMS

<b>Test case name</b>	<b>Get a Specific Display Message - Id</b>	
<b>Test case Id</b>	TC_O_07_CSMS	
<b>Use case Id(s)</b>	O04	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can request specific installed DisplayMessages configured via OCPP in a Charging Station. The Charging Station can remove messages when they are out-dated, or transactions have ended. It can be very useful for a CSO to be able to view to current list of messages, so the CSO knows which messages are (still) configured.	
<b>Purpose</b>	To verify if the CSMS is able to request a specific id message from the charging station according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> A display message is configured.	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetDisplayMessagesResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>GetDisplayMessagesRequest</b>
	3. The OCTT sends a <b>NotifyDisplayMessagesRequest</b>	4. The CSMS responds with a <b>NotifyDisplayMessagesResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>GetDisplayMessagesRequest</b> - <b>id</b> <Configured_Id> - <b>priority</b> <Omitted> - <b>state</b> <Omitted> - <b>requestId</b> <Generated Id>	
	<b>Post scenario validations:</b> - N/a	

Table 186. Test Case Id: TC\_O\_08\_CSMS

<b>Test case name</b>	<b>Get a Specific Display Message - Priority</b>	
<b>Test case Id</b>	TC_O_08_CSMS	
<b>Use case Id(s)</b>	O04	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can request specific installed DisplayMessages configured via OCPP in a Charging Station. The Charging Station can remove messages when they are out-dated, or transactions have ended. It can be very useful for a CSO to be able to view to current list of messages, so the CSO knows which messages are (still) configured.	
<b>Purpose</b>	To verify if the CSMS is able to request specific priority messages from the charging station according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> A message with <Configured_Priority> is configured	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetDisplayMessagesResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>GetDisplayMessagesRequest</b>
	3. The OCTT sends a <b>NotifyDisplayMessagesRequest</b>	4. The CSMS responds with a <b>NotifyDisplayMessagesResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>GetDisplayMessagesRequest</b> - <b>priority</b> <Configured_Priority> - <b>id</b> <Omitted> - <b>state</b> <Omitted> - <b>requestId</b> <Generated Id>	
	<b>Post scenario validations:</b> - N/a	

Table 187. Test Case Id: TC\_O\_09\_CSMS

<b>Test case name</b>	<b>Get a Specific Display Message - State</b>	
<b>Test case Id</b>	TC_O_09_CSMS	
<b>Use case Id(s)</b>	O04	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSO can request specific installed DisplayMessages configured via OCPP in a Charging Station. The Charging Station can remove messages when they are out-dated, or transactions have ended. It can be very useful for a CSO to be able to view to current list of messages, so the CSO knows which messages are (still) configured.	
<b>Purpose</b>	To verify if the CSMS is able to request specific state messages from the charging station according to the mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> A message with <Configured_State> is configured	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>GetDisplayMessagesResponse</b> with <b>status Accepted</b>	1. The CSMS sends a <b>GetDisplayMessagesRequest</b>
	3. The OCTT sends a <b>NotifyDisplayMessagesRequest</b>	4. The CSMS responds with a <b>NotifyDisplayMessagesResponse</b> .
<b>Tool validations</b>	* Step 1: Message <b>GetDisplayMessagesRequest</b> - <b>state</b> <Configured_State> - <b>priority</b> <Omitted> - <b>id</b> <Omitted> - <b>requestId</b> <Generated Id>	
	<b>Post scenario validations:</b> - N/a	

Table 188. Test Case Id: TC\_O\_10\_CSMS

<b>Test case name</b>	<b>Set Display Message - Specific transaction - Unknown Transaction Id</b>	
<b>Test case Id</b>	TC_O_10_CSMS	
<b>Use case Id(s)</b>	O02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how a CSMS can attempt to set a DisplayMessage for a transactionId that the CS does not know. The CS will respond with a SetDisplayMessageResponse status of UnknownTransaction.	
<b>Purpose</b>	To verify if the CSMS is able to send a display message correctly according the mechanism as described in the OCPP specification for a specific transaction.	
<b>Prerequisite(s)</b>	If the CSMS supports sending a SetDisplayMessageRequest with a transactionId for a transaction that does not exist.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a display message for a specific transaction.	
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with status <i>UnknownTransaction</i>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.transactionId</b> not omit AND - <b>message.priority</b> <Configured Priority>	
	<b>Post scenario validations:</b> - N/a	

Table 189. Test Case Id: TC\_O\_14\_CSMS

<b>Test case name</b>	<b>Set Display Message - Remove message after EndTime</b>	
<b>Test case Id</b>	TC_O_14_CSMS	
<b>Use case Id(s)</b>	O01	
<b>Requirement(s)</b>	O01_FR_05	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send a SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send the request with a endTime according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a SetDisplayMessageRequest with a endTime.	
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id> - <b>message.endTime</b> <Configured endTime>	
	<b>Post scenario validations:</b> - N/a	

Table 190. Test Case Id: TC\_O\_17\_CSMS

<b>Test case name</b>	<b>Set Display Message - NotSupportedPriority</b>	
<b>Test case Id</b>	TC_O_17_CSMS	
<b>Use case Id(s)</b>	O01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send a SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send a display message with a specific priority, on which the Charging station responds not supported, according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with <b>status NotSupportedPriority</b>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id> - <b>message.priority</b> <Configured priority>	
	<b>Post scenario validations:</b> - N/a	



Table 191. Test Case Id: TC\_O\_18\_CSMS

<b>Test case name</b>	<b>Set Display Message - NotSupportedState</b>	
<b>Test case Id</b>	TC_O_18_CSMS	
<b>Use case Id(s)</b>	O01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send a SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send a display message with a specific state, on which the Charging station responds not supported, according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with <b>status NotSupportedState</b>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id> - <b>message.state</b> <Configured state>	
	<b>Post scenario validations:</b> - N/a	

Table 192. Test Case Id: TC\_O\_19\_CSMS

<b>Test case name</b>	<b>Set Display Message - NotSupportedMessageFormat</b>	
<b>Test case Id</b>	TC_O_19_CSMS	
<b>Use case Id(s)</b>	O01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to sent an SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send a display message with a specific MessageFormat, on which the Charging station responds not supported, according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with <b>status NotSupportedMessageFormat</b>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id>	
	<b>Post scenario validations:</b> - N/a	

Table 193. Test Case Id: TC\_O\_26\_CSMS

<b>Test case name</b>	<b>Set Display Message - Rejected</b>	
<b>Test case Id</b>	TC_O_26_CSMS	
<b>Use case Id(s)</b>	O01	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send a SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send the request according to the DisplayMessage mechanism as described in the OCPP specification which gets rejected.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a SetDisplayMessageRequest with a Normal Cycle priority.	
	2. The OCTT responds with a SetDisplayMessageResponse with status Rejected	1. The CSMS sends a SetDisplayMessageRequest
<b>Tool validations</b>	* Step 1: Message SetDisplayMessageRequest - message.id <Generated Id> - message.priority <Configured Priority>	
	<b>Post scenario validations:</b> - N/a	

Table 194. Test Case Id: TC\_O\_27\_CSMS

<b>Test case name</b>	<b>Set Display Message - Specific transaction - Display message at StartTime</b>	
<b>Test case Id</b>	TC_O_27_CSMS	
<b>Use case Id(s)</b>	O02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send a SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send the request with a startTime for a specific transaction according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a1	
	<b>Charging State:</b> State is <i>EnergyTransferStarted</i>	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id> - <b>message.startDateTime</b> <Configured startDateTime> - <b>message.transactionId</b> is present	
	<b>Post scenario validations:</b> - N/a	

Table 195. Test Case Id: TC\_O\_28\_CSMS

<b>Test case name</b>	<b>Set Display Message - Specific transaction - Remove message after EndTime</b>	
<b>Test case Id</b>	TC_O_28_CSMS	
<b>Use case Id(s)</b>	O02	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	This test case describes how the CSMS can be requested to send a SetDisplayMessageRequest to the charging station. Depending on the given parameters the message shall be displayed a certain way and at a certain moment on the Charging Station. These messages are displayed additionally on a Charging Station and are not part of the firmware.	
<b>Purpose</b>	To verify if the CSMS is able to send the request with a endTime for a specific transaction according to the DisplayMessage mechanism as described in the OCPP specification.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Charging State:</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	2. The OCTT responds with a <b>SetDisplayMessageResponse</b> with status <i>Accepted</i>	1. The CSMS sends a <b>SetDisplayMessageRequest</b>
<b>Tool validations</b>	* Step 1: Message <b>SetDisplayMessageRequest</b> - <b>message.id</b> <Generated Id> - <b>message.priority</b> <Configured Priority> - <b>message.endDateTime</b> <Configured endDateTime> - <b>message.state</b> <Configured State> - <b>message.transactionId</b> is present	
	<b>Post scenario validations:</b> - N/a	

## 17. P DataTransfer

Table 196. Test Case Id: TC\_P\_02\_CSMS

<b>Test case name</b>	<b>Data Transfer to the CSMS - Rejected / Unknown VendorId / Unknown Messageld</b>	
<b>Test case Id</b>	TC_P_02_CSMS	
<b>Use case Id(s)</b>	P02	
<b>Requirement(s)</b>	P02.FR.06, P02.FR.07	
<b>System under test</b>	CSMS	
<b>Description</b>	The DataTransfer message to send information for functions that are not supported by OCPP.	
<b>Purpose</b>	To verify whether the CSMS is able to handle receiving a DataTransferRequest, even if it does not support any vendor-specific implementations.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>DataTransferRequest</b> with <b>vendorId</b> <Configured vendorId> <b>messageld</b> <Configured messageld>	2. The CSMS responds with a <b>DataTransferResponse</b>
<b>Tool validations</b>	* Step 2: Message: <b>DataTransferResponse</b> - <b>status</b> must be <i>UnknownVendorId</i> OR <i>UnknownMessageld</i> OR <i>Rejected</i> (Rejected will also be allowed, because there are implementers that like to just reject the message when the Charging Station does not support any vendor-specific features.	
	<b>Post scenario validations:</b> N/a	

Table 197. Test Case Id: TC\_P\_03\_CSMS

<b>Test case name</b>	<b>Able to receive custom data</b>	
<b>Test case Id</b>	TC_P_03_CSMS	
<b>Use case Id(s)</b>	N/a	
<b>Requirement(s)</b>	N/a	
<b>System under test</b>	CSMS	
<b>Description</b>	Checks if the CSMS is able to receive custom data.	
<b>Purpose</b>	To verify whether the CSMS is able to handle receiving custom data.	
<b>Prerequisite(s)</b>	N/a	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>StatusNotificationRequest</b> with <b>customData</b> <customData>	2. The CSMS responds with a <b>StatusNotificationResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> with <b>customData</b> customData <b>transactionInfo.customData</b> <customData>	4. The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
	<b>Post scenario validations:</b> N/a	

## 18. Reusable states

Testcases can refer to a reusable state at the before or main stage. The steps described at the reusable state will be executed and then it will return to the testcase that called the reusable state.

Table 198. Reusable State: Booted

State	Booted	
System under test	CSMS	
Description	This state will simulate that the Charging Station is completely power cycled. The OCTT end in a state where it is "booted" back up and is in idle mode.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<ol style="list-style-type: none"> <li>The OCTT sends a <b>BootNotificationRequest</b> with <b>reason</b> <i>PowerUp</i> <b>chargingStation.model</b> &lt;Configured model&gt; <b>chargingStation.vendorName</b> &lt;Configured vendorName&gt;</li> </ol>	<ol style="list-style-type: none"> <li>The CSMS responds with a <b>BootNotificationResponse</b></li> </ol>
	<ol style="list-style-type: none"> <li>The OCTT notifies the CSMS about the current state of all connectors.             Message: <b>StatusNotificationRequest</b>            with <b>connectorStatus</b> <i>Available</i>            Message: <b>NotifyEventRequest</b>            with <b>trigger</b> <i>Delta</i>  <b>actualValue</b> "Available"  <b>component.name</b> "Connector"  <b>variable.name</b> "AvailabilityState"         </li> </ol>	<ol style="list-style-type: none"> <li>The CSMS responds accordingly.</li> </ol>
Tool validations	* Step 2: Message: <b>BootNotificationResponse</b> - <b>status</b> <i>Accepted</i>	
Post condition	State is <i>Booted</i>	



Table 199. Reusable State: Reserved

<b>State</b>	<b>Reserved</b>	
<b>System under test</b>	CSMS	
<b>Description</b>	This state will simulate a reservation for a specified evse.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to send a ReserveNowRequest for specific EVSE.	
	<b>2.</b> The OCTT responds with a <b>ReserveNowResponse</b> With status <i>Accepted</i>	<b>1.</b> The CSMS sends a <b>ReserveNowRequest</b>
	<b>3.</b> The OCTT notifies the CSMS about the current state of the connector(s) of the Specified EVSE  Message: <b>StatusNotificationRequest</b> with <b>connectorStatus</b> <i>Reserved</i> Message: <b>NotifyEventRequest</b> with <b>trigger</b> <i>Delta</i> <b>actualValue</b> <i>"Reserved"</i> <b>component.name</b> <i>"Connector"</i> <b>variable.name</b> <i>"AvailabilityState"</i>	<b>4.</b> The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message: <b>ReserveNowRequest</b> - <b>evseld</b> must be <i>&lt;Specified evseld&gt;</i> - <b>connectorType</b> must be omitted - <b>idToken.idToken</b> <i>&lt;Configured valid_idtoken_idtoken&gt;</i> - <b>idToken.type</b> <i>&lt;Configured valid_idtoken_type&gt;</i>	
<b>Post condition</b>	<b>State</b> is <i>Reserved</i>	

Table 200. Reusable State: Unavailable

<b>State</b>	<b>Unavailable</b>	
<b>System under test</b>	Charging Station	
<b>Description</b>	This state will simulate that Charging Station / EVSEs / connectors are set to AvailabilityState Unavailable.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to change the availability of the specified components to Inoperative.	
	2. The OCTT responds with a <b>ChangeAvailabilityResponse</b> with <b>status</b> <i>Accepted</i>	1. The CSMS sends a <b>ChangeAvailabilityRequest</b>
	3. The OCTT notifies the CSMS about the current state of all connectors belonging to the specified EVSE (and optionally also from the EVSE itself). Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> <i>Unavailable</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> <i>Delta</i> - <b>actualValue</b> <i>"Unavailable"</i> - <b>component.name</b> <i>"ChargingStation" / EVSE / Connector</i> - <b>variable.name</b> <i>"AvailabilityState"</i>	4. The CSMS responds accordingly.
<b>Tool validations</b>	* Step 1: Message <b>ChangeAvailabilityRequest</b> - <b>operationalStatus</b> <i>Inoperative</i> - <b>evse</b> <i>&lt;Specified evseld&gt;</i> - <b>connectorId</b> <i>omitted</i>	
<b>Post condition</b>	<b>State</b> is <i>Unavailable</i>	

Table 201. Reusable State: EVConnectedPreSession

State	EVConnectedPreSession	
System under test	CSMS	
Description	This state will simulate that the EV and EVSE of the simulated Charging Station are connected.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<p>1. The OCTT notifies the CSMS about the status change of the connector</p> <p>Message: <b>StatusNotificationRequest</b></p> <ul style="list-style-type: none"> <li>- <b>connectorStatus</b> is <i>Occupied</i></li> </ul> <p>Message: <b>NotifyEventRequest</b></p> <ul style="list-style-type: none"> <li>- <b>trigger</b> is <i>Delta</i></li> <li>- <b>actualValue</b> is <i>Occupied</i></li> <li>- <b>component.name</b> is <i>Connector</i></li> <li>- <b>variable.name</b> is <i>AvailabilityState</i></li> </ul>	<p>2. The CSMS responds accordingly.</p>
	<p>3. The OCTT sends a <b>TransactionEventRequest</b></p> <p>With <b>triggerReason</b> is <i>CablePluggedIn</i></p> <p><b>transactionInfo.chargingState</b> is <i>EVConnected</i></p> <p><b>evse.id</b> &lt;Configured evseld&gt;</p> <p><b>evse.connectorId</b> &lt;Configured connectorId&gt;</p> <p>If <b>State</b> is <i>Authorized</i> then</p> <p><b>eventType</b> is <i>Updated</i></p> <p>else</p> <p><b>eventType</b> is <i>Started</i></p>	<p>4. The CSMS responds with a <b>TransactionEventResponse</b></p>
Tool validations	N/a	
Post condition	<b>State</b> is <i>EVConnectedPreSession</i>	

Table 202. Reusable State: Authorized

State	Authorized	
System under test	CSMS	
Description	This state will simulate that the EV Driver is locally authorizing to start a transaction on the simulated Charging Station.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends an <b>AuthorizeRequest</b> With <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type>	2. The CSMS responds with an <b>AuthorizeResponse</b>
	3. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>Authorized</i> <b>idToken.idToken</b> <Configured valid_idtoken_idtoken> <b>idToken.type</b> <Configured valid_idtoken_type> If <b>State</b> is <i>EVConnectedPreSession</i> then <b>eventType</b> is <i>Updated</i> else <b>eventType</b> is <i>Started</i>	4. The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	* Step 2: Message: <b>AuthorizeResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i> * Step 4: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i>	
Post condition	<b>State</b> is <i>Authorized</i>	

Table 203. Reusable State: EnergyTransferStarted

State	EnergyTransferStarted	
System under test	CSMS	
Description	This state will simulate that there is transferring energy between the EV and EVSE of the simulated Charging Station.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> If <b>State</b> is NOT <i>Authorized</i> then execute <b>Reusable State</b> <i>Authorized</i> If <b>EVConnected</b> is <i>true</i> , then proceed to part 2 Else proceed to part 1.	
Main (Part 1) (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT notifies the CSMS about the status change of the connector.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> is <i>Occupied</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> is <i>Delta</i> - <b>actualValue</b> is <i>Occupied</i> - <b>component.name</b> is <i>Connector</i> - <b>variable.name</b> is <i>AvailabilityState</i>	<b>2.</b> The CSMS responds accordingly.
	<b>3.</b> The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>CablePluggedIn</i> <b>transactionInfo.chargingState</b> is <i>EVConnected</i> <b>evse.id</b> <Configured evseld> <b>evse.connectorId</b> <Configured connectorId> <b>eventType</b> is <i>Updated</i>	<b>4.</b> The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	N/a	
Main (Part 2) (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>5.</b> The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>Charging</i> <b>eventType</b> is <i>Updated</i>	<b>6.</b> The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	N/a	
Post condition	<b>State</b> is <i>EnergyTransferStarted</i> <b>EVConnected</b> is <i>true</i>	

Table 204. Reusable State: EnergyTransferSuspended

State	EnergyTransferSuspended	
System under test	CSMS	
Description	This state will simulate that the Charging Station is in a state where the energy transfer is suspended by the EV.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> If <b>State</b> is NOT <i>EnergyTransferStarted</i> then execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Notes(s):</u> The tool will wait for <Configured Transaction Duration> seconds	
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>SuspendedEV</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	N/a	
Post condition	<b>State</b> is <i>EnergyTransferSuspended</i>	

Table 205. Reusable State: StopAuthorized

State	StopAuthorized	
System under test	CSMS	
Description	This state will simulate that the Charging Station is in a state where the charging session is authorized to stop.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> If <b>State</b> is NOT <i>EnergyTransferStarted</i> then execute <b>Reusable State</b> <i>EnergyTransferStarted</i>	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Notes(s):</u> The tool will wait for <Configured Transaction Duration> seconds	
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>StopAuthorized</i> <b>eventType</b> is <i>Updated</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	* Step 2: Message: <b>TransactionEventResponse</b> - <b>idTokenInfo.status</b> must be <i>Accepted</i>	
Post condition	<b>State</b> is <i>StopAuthorized</i>	

Table 206. Reusable State: EVConnectedPostSession

State	EVConnectedPostSession	
System under test	CSMS	
Description	This state will simulate that the Charging Station is in a state where the energy transfer has been stopped and the transaction is NOT authorized to resume energy transfer without re-authorization.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> If <b>State</b> is NOT <i>StopAuthorized</i> then execute <b>Reusable State</b> <i>StopAuthorized</i>	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>ChargingStateChanged</i> <b>transactionInfo.chargingState</b> is <i>EVConnected</i> <b>eventType</b> is <i>Updated</i>	2. The CSMS responds with a <b>TransactionEventResponse</b>
Tool validations	N/a	
Post condition	<b>State</b> is <i>EVConnectedPostSession</i>	



Table 207. Reusable State: EVDisconnected

<b>State</b>	<b>EVDisconnected</b>	
<b>System under test</b>	CSMS	
<b>Description</b>	This state will simulate that the EV and EVSE of the simulated Charging Station are disconnected, after the charging session is authorized to stop.	
<b>Before</b> (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> If <b>State</b> is NOT <i>EVConnectedPostSession</i> then execute <b>Reusable State</b> <i>EVConnectedPostSession</i>	
<b>Main</b> (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<b>1.</b> The OCTT notifies the CSMS about the status change of the connector.  Message: <b>StatusNotificationRequest</b> - <b>connectorStatus</b> is <i>Available</i> Message: <b>NotifyEventRequest</b> - <b>trigger</b> is <i>Delta</i> - <b>actualValue</b> is <i>Available</i> - <b>component.name</b> is <i>Connector</i> - <b>variable.name</b> is <i>AvailabilityState</i>	<b>2.</b> The CSMS responds accordingly.
	<b>3.</b> The OCTT sends a <b>TransactionEventRequest</b> With <b>triggerReason</b> is <i>EVCommunicationLost</i> <b>transactionInfo.chargingState</b> is <i>Idle</i> <b>transactionInfo.stoppedReason</b> is <i>EVDisconnected</i> <b>eventType</b> is <i>Ended</i>	<b>4.</b> The CSMS responds with a <b>TransactionEventResponse</b>
<b>Tool validations</b>	N/a	
<b>Post condition</b>	<b>State</b> is <i>EVDisconnected</i>	

Table 208. Reusable State: GetInstalledCertificates

State	GetInstalledCertificates	
System under test	CSMS	
Description	The hashData from installed certificates of the specified type will be retrieved from the Charging Station	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to send a <code>GetInstalledCertificateIdsRequest</code> with <code>certificateType</code> _<Specified certificateType>	
	<p>2. The OCTT responds with a <b>GetInstalledCertificateIdsResponse</b> With <b>status</b> is <i>Accepted</i> <b>certificateHashDataChain</b> contains an entry with following values:  <b>certificateHashDataChain[0].certificateType</b> is &lt;Specified certificateType&gt;  <b>certificateHashDataChain[0].certificateHashData</b> contains &lt;HashData from the configured certificate of the specified certificateType&gt;</p>	<p>1. The CSMS sends a <b>GetInstalledCertificateIdsRequest</b></p>
Tool validations	<p>* Step 1:  Message: <b>GetInstalledCertificateIdsRequest</b>  - <b>certificateType</b> must be &lt;Specified certificateType&gt;</p>	
Post condition	Certificate of the specified certificateType is retrieved from the Charging Station.	

Table 209. Reusable State: CertificateInstalled

State	CertificateInstalled	
System under test	CSMS	
Description	A pre configured certificate of the specified certificateType will be installed.	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Trigger the CSMS to send an InstallCertificateRequest with certificateType <Specified certificateType>	
	2. The OCTT responds with a <b>InstallCertificateResponse</b> With <b>status</b> is <i>Accepted</i>	1. The CSMS sends a <b>InstallCertificateRequest</b>
Tool validations	* Step 1: Message: <b>InstallCertificateRequest</b> - <b>certificateType</b> must be <Specified certificateType> - <b>certificate</b> must be <The configured certificate of the specified certificateType.>	
Post condition	Certificate of the specified certificateType is stored at the Charging Station.	

Table 210. Reusable State: ISO15118SmartCharging

State	ISO15118SmartCharging	
System under test	CSMS	
Description		
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Scenario)	<b>Charging Station</b>	<b>CSMS</b>
	1. The OCTT sends a <b>NotifyEVChargingNeedsRequest</b> with <b>evseld</b> <Configured evseld> <b>maxScheduleTuples</b> & <b>chargingNeeds</b> <Configured values from mock EV>+	2. The CSMS responds with a <b>NotifyEVChargingNeedsResponse</b> .
	4. The OCTT responds with a <b>SetChargingProfileResponse</b> with: <b>status</b> <i>Accepted</i>	3. The CSMS sends a <b>SetChargingProfileRequest</b> <u>Note(s):</u> - If <b>NotifyEVChargingNeedsResponseStatus</b> was <i>Processing</i> , the OCTT will wait 60 seconds for the request
	5. The OCTT sends a <b>NotifyEVChargingScheduleRequest</b> with <b>evseld</b> <Configured evseld> <b>chargingSchedule</b> <ChargingSchedule provided at step 3>	6. The CSMS responds with a <b>NotifyEVChargingScheduleResponse</b> .
	7. The OCTT sends a <b>TransactionEventRequest</b> with <b>triggerReason</b> <ChargingStateChanged> <b>transactionInfo.chargingState</b> <Charging>	8. The CSMS responds with a <b>TransactionEventResponse</b> .

State	ISO15118SmartCharging
Tool validations	<ul style="list-style-type: none"><li>* Step 1: Message: <b>NotifyEVChargingNeedsResponse</b> - <b>Status</b> <i>Accepted or Processing</i></li><li>* Step 3: Message: <b>SetChargingProfileRequest</b> - <b>chargingProfilePurpose</b> <i>&lt;TxProfile&gt;</i> - <b>transactionId</b> <i>&lt;Provided transactionId from before&gt;</i></li><li>* Step 4: Message: <b>NotifyEVChargingScheduleResponse</b> - <b>status</b> <i>&lt;Accepted&gt;</i></li></ul>
Post condition	N/a

Table 211. Memory State: RenewChargingStationCertificate

State	RenewChargingStationCertificate	
System under test	Charging Station	
Description	The ChargingStationCertificate is renewed using A02/A03	
Before (Preparations)	<b>Configuration State:</b> N/a	
	<b>Memory State:</b> N/a	
	<b>Reusable State(s):</b> N/a	
Main (Test scenario)	<b>Charging Station</b>	<b>CSMS</b>
	<u>Manual Action:</u> Request the CSMS to send a Trigger Message Request with <b>requestedMessage</b> <i>SignChargingStationCertificate</i>	
	<b>2.</b> The OCTT sends a <b>TriggerMessageResponse</b> with <b>status</b> <i>Accepted</i>	<b>1.</b> The CSMS sends a <b>TriggerMessageRequest</b> With <b>requestedMessage</b> <i>SignChargingStationCertificate</i>
	<b>3</b> The OCTT sends a <b>SignCertificateRequest</b>	<b>4.</b> The CSMS responds with a <b>SignCertificateResponse</b> with <b>status</b> <i>Accepted</i>
	<b>6.</b> The OCTT sends a <b>CertificateSignedResponse</b> with <b>status</b> <i>Accepted</i>	<b>5.</b> The CSMS sends a <b>CertificateSignedRequest</b> With <b>certificateChain</b> <i>&lt;Certificate generated from the received CSR from step 3 and signed by the provided CSMS Root certificate&gt;</i> <b>certificateType</b> <i>ChargingStationCertificate</i>
Tool validations	<p>* Step 2: Message: <b>TriggerMessageResponse</b> - <b>status</b> must be <i>Accepted</i></p> <p>* Step 3: Message: <b>SignCertificateRequest</b> - <b>csr</b> must contain <i>&lt;An CSR that meets the following requirements: When using RSA or DSA the key must be at least 2048 bits long. and when using elliptic curve cryptography the key must be at least 224 bits long. The received CSR must be transmitted as described in RFC 2986 and then encoded in Privacy-Enhanced Mail (PEM) format.&gt;</i></p> <p>* Step 6: Message: <b>CertificateSignedResponse</b> - <b>status</b> must be <i>Accepted</i></p>	
	<b>Post scenario validations:</b> N/a	