



ALLEGATO 1
TEST PROTOCOL

Content

1	DESCRIPTION.....	3
1.1	PURPOSE	3
1.2	TEST ENVIRONMENT	3
1.3	ABBREVIATIONS USED	3
2	VERIFICATION.....	4
2.1	TEST CASES	4
2.1.1	<i>Hard Reset (make charger ready for new test session).....</i>	<i>4</i>
2.1.2	<i>Get Configuration</i>	<i>5</i>
2.1.3	<i>ChangeConfiguration</i>	<i>5</i>
2.1.4	<i>Clear cache (make charger ready for new test session).....</i>	<i>6</i>
2.1.5	<i>Clear Whitelist (make charger ready for new test session).....</i>	<i>6</i>
2.1.6	<i>Authorize with Valid new key</i>	<i>6</i>
2.1.7	<i>Authorize with Invalid new key</i>	<i>7</i>
2.1.8	<i>Authorize with existing key and LocalPreAuthorize is set to false.....</i>	<i>7</i>
2.1.9	<i>Authorize with existing key and LocalPreAuthorize is set to true</i>	<i>8</i>
2.1.10	<i>Starttransaction with key that is valid in cache but invalid in CDMW.....</i>	<i>9</i>
2.1.11	<i>Starttransaction with car NOT connected when authenticating RFID</i>	<i>10</i>
2.1.12	<i>Starttransaction with car connected when authenticating RFID</i>	<i>10</i>
2.1.13	<i>Reset while charging</i>	<i>11</i>
2.1.14	<i>Metervalues</i>	<i>12</i>
2.1.15	<i>Emergency Stop.....</i>	<i>13</i>
2.1.16	<i>Power Failure in Online Mode</i>	<i>14</i>
2.1.17	<i>Offline testing (with powerfailure).....</i>	<i>14</i>
2.1.18	<i>Offline testing (without powerfailure).....</i>	<i>16</i>
2.1.19	<i>Remote Start / Remote Stop</i>	<i>17</i>
2.1.20	<i>Unlock connector/Stop transaction from car</i>	<i>18</i>
2.1.21	<i>Freemode</i>	<i>18</i>
2.1.22	<i>Smart Charging</i>	<i>20</i>
2.1.23	<i>Heartbeat.....</i>	<i>21</i>
2.1.24	<i>Get Diagnostics.....</i>	<i>21</i>
2.1.25	<i>Firmware update</i>	<i>22</i>
2.1.26	<i>Firmware update failure</i>	<i>Errore. Il segnalibro non è definito.</i>
3	OBSERVATIONS	22
4	QUESTIONS TO MANUFACTURER	22
5	OFF	ERRORE. IL SEGNALIBRO NON È DEFINITO.

1 Description

1.1 Purpose

This is a test procedure created to check integration communication between chargers (CP) and a back-office (CS) based on OCPP standard.

This document is a guideline that we developed for making testing simpler and more effective.

1.2 Test environment

Please fill in the following details

Date of testing	
Charge point	
Tester(email, phone number)	
Name on Testing tool	
Manufacturer of the charger	
Charger model	
FW Version	
HW Version	
Connector type(s)	
Number of connectors	
Unit ID (OCPP ID)	
Central System (CDMW)	
OCPP Endpoint URL	
RFID Tag used	
Test environment	
CP-CS communication link Eg.: LAN(Ethernet cable, behind NAT and firewall) – WAN - CDMW	

1.3 Abbreviations used

Acronym	Meaning
CB	Charge Box
CP	Charge Point
CS	Central System
OCPP	Open Charge Point Protocol
WL	White-list
CTC	Cloud to Cloud
PDU	Protocol Data Unit

CAC	Cloud And Cloud
CDMW	Charge and Drive Web

2 Verification

One of these codes should be put in the status field for each test case

Status codes	
PASS	Test result valid
FAIL	Test result invalid
CHECK	Possible improvements and comments
CP NOT SUPPORTED	Not supported at charge point side

2.1 Test Cases

Before starting the testing

1. Add CP in CDMW stage environment.
2. Please connect from internet and use the endpoint <wss://ocppext-stage.chargedrive.com/websocket/>
Encryption is a requirement
3. Make sure that the charger is connected to CDMW and you can see the "Online" status in our system.
4. Have at least 3 RFID tags ready and add 2 of those in CDMW, both 7 and 4 byte keys.
Idtag1 = 7 Byte RFID
Idtag2 = 4 Byte RFID
Idtag3 = 4 Byte RFID (not added in CDMW)

Do the below test cases in the same order as listed in this document, i.e. start with first test case and end with last test case.

If there is a test case charger can't do, then put **CP NOT SUPPORTED** in status field for that test case.

The date and time should be put in format **hour:minutes:seconds (UTC TIME)**

2.1.1 Hard Reset (make charger ready for new test session)		Status
MANDATORY REQUIREMENT		
Requirement	CP shall always be able to do a reset when hard reset request is sent from CS, Hard reset should do a complete powercycle and restart of OS on charger but local list and cache with all RFIDs should be saved.	
Procedure	<div>1. Detach charging cables connected to connectors.</div> <div>2. Go to the charge point page in CDMW and press hard reset</div>	
Start date UTC time		
End date UTC time		
Expected results	<div>1. When pressing Hard reset CS shall send Hard reset request to CP</div>	

	<ol style="list-style-type: none"> 2. CP shall respond with accepted 3. CP should send Status unavailable on all connectors 4. CP should do a complete powercycle on charger 5. CP shall send bootnotification request to CS 6. If CS accepts the boot then charger shall send status available on all connectors, including ID0 connector 	
Actual Results/Comments		
Questions		
Answers		

2.1.2 Get Configuration		Status
MANDATORY REQUIREMENT		
Requirement	CS will send an empty getconfiguration request to CP after CP sends bootnotification Charger must send all charger keys.	
Procedure	<ol style="list-style-type: none"> 1. CS sends an empty getconfigurationrequest to CP asking it for all Config Keys after a bootnotification been sent from CP 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP Shall send a response with a list of ALL its config keys 2. In page Configuration keys on CP page in CDMW all keys charger has sent shall be there. 	
Actual Results/Comments		
Question		
Answer		

2.1.3 ChangeConfiguration		Status
MANDATORY REQUIREMENT		
Requirement	Chargepoint must send accepted on keys that is changed to correct values Chargepoint must send rejected on keys that is changed to faulty values Chargepoint must send reboot required on settings that requires a reset in order to apply, for an example if you change backend endpoint	
Procedure	<ol style="list-style-type: none"> 1. Change a setting to a value you know charger will return rejected on 2. Change a setting to a value you know charger will return reboot required on 3. Change back the settings you just changed as they were before changing them 	Check
Start date UTC time		

End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. Charger shall send rejected 2. Charger shall send reboot required 3. Charger shall accept the old values 	
Actual Results/Comments		

2.1.4 Clear cache (make charger ready for new test session)		Status
PREFERENTIAL REQUIREMENT		
Requirement	Charger shall be able to clear its cache memory on local RFIDs	
Procedure	<ol style="list-style-type: none"> 1. Go to CP page in CDMW and press clear cache 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CS sends a clear cache request 2. CP should respond with accepted and delete cached keys in memory 	
Actual Results/Comments		

2.1.5 Clear Whitelist (make charger ready for new test session)		Status
MANDATORY REQUIREMENT		
Requirement	Charger shall be able to delete its local list when receiving an empty local list request from CS	
Procedure	<ol style="list-style-type: none"> 1. Go to CP page in CDMW and press clear whitelist 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CS sends an empty local list to CP. 2. CP should respond with accepted and remove all saved keys In the local list 	
Actual Results/Comments		

2.1.6 Authorize with Valid new key		Status
MANDATORY REQUIREMENT		
Requirement	Charger should send authorize when it reads a key that's not saved in memory	
Procedure	<ol style="list-style-type: none"> 1. Change config key ConnectionTimeout to 60 2. Change config key AuthorizationCacheEnabled to true 3. Change config key LocalPreAuthorize to false 4. Detach charging cables connected to connectors, charger shall be in status available 	

	5. Read Idtag1 on charger's RFID reader (IF there is one reader per connector then use the RFID reader on connector 1) 6. Don't connect charging cable.	
Start date UTC time		
End date UTC time		
Expected results	1. CS shall send Changeconfig and change the config key ConnectionTimeOut to 60 2. CP shall accept 3. CS shall send Changeconfig and change the config key AuthorizationCacheEnabled to true 4. CS shall send changeconfig and change config key LocalPreAuthorize to false 5. CP shall accept 6. CP should send Authorize request with Idtag1 7. CS should respond with accepted (key must be added as valid in CDMW) 8. CP should send status message preparing on connector1 OR if the charger only has one RFID reader for all connectors then it should send preparing on all connectors that is available 9. CP should timeout after 60 seconds. When charger has timed out, status available should be sent on the connector/s	
Actual Results/Comments		

2.1.7 Authorize with Invalid new key		Status
MANDATORY REQUIREMENT		
Requirement	Charger should send authorize when it reads a key that's not saved in memory	
Procedure	1. Detach charging cables connected to connectors, charger shall be in status available 2. Read Idtag3 on charger's RFID reader (IF there is one reader per connector then use the RFID reader on connector 1)	
Start date UTC time		
End date UTC time		
Expected results	1. CP should send Authorize request with Idtag3 2. CS should respond with Invalid	
Actual Results/Comments		

2.1.8 Authorize with existing key and LocalPreAuthorize is set to false		Status
PREFERENTIAL REQUIREMENT		
Requirement	According to OCPP guidelines, LocalPreAuthorize is a required config key (go to 9.1.11 on OCPP for more info) Charger should send authorize when it reads a key that's saved in memory and config key preauthorize is set to false	
Procedure	1. Detach cables connected to connectors, charger shall be in available status	

	<ol style="list-style-type: none"> 2. Change config key LocalPreAuthorize to false (if it is already false then do not change) 3. Read Idtag1 on chargers RFID reader (IF there is reader/connector then use reader on connector 1) 4. Don't connect charging cable. 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. If you changed LocalPreAuthorize then CS shall send a changeconfig, charger shall respond with accepted. 2. CP should send Authorize request with Idtag1 3. CS should respond with accepted 4. CP should send status message preparing on connector1 OR if the charger only has one RFID reader for all connectors then it should send preparing on all connectors that is available 5. CP should timeout after 60 seconds. When charger has timed out it shall send status available 	
Actual Results/Comments		

2.1.9 Authorize with existing key and LocalPreAuthorize is set to true		Status
PREFERENTIAL REQUIREMENT		
Requirement	Charger should send authorize when it reads a key that's not saved in memory (go to 9.1.11 on OCPP for more info)	
Procedure	<ol style="list-style-type: none"> 1. Detach cables connected to connectors, charger shall be in available status 2. Change config key LocalPreAuthorize to true 3. Read Idtag1 on chargers RFID reader (IF there is reader/connector then use reader on connector 1) 4. Don't connect cable. 5. Send Clear Cache 6. Read Idtag1 on chargers RFID reader (IF there is reader/connector then use reader on connector 1) 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CS shall send a changeconfig for the change made on config key LocalPreAuthorize, charger shall respond with accepted. 2. CP should NOT send an Authorize request with Idtag1 3. CP should send status message preparing on connector1 OR if the charger only has one RFID reader for all connectors then it should send preparing on all connectors that is available 4. CP should timeout after 60 seconds. When charger has timed out status available should be sent on available connectors to CS 5. CP should answer accepted to clear cache command 6. CP should send authorize because the idtag was not in cache. 	
Actual Results/Comments		

2.1.10 Start transaction with key that is valid in cache but invalid in CDMW		Status
PREFERENTIAL REQUIREMENT		
Requirement	Charger shall work with the required configuration key StopTransactionOnInvalidId (9.1.24) If that key is true then charger shall stop a started transaction that received invalid in startresponse	
Procedure	<ol style="list-style-type: none"> 1. Change StopTransactionOnInvalidId to True 2. Make Idtag1 Invalid in CDMW. Go to the Idtag page for idtag1 and uncheck enable 3. Connect car (testtool) to connector1 4. Read Idtag1 on charger's RFID reader, start charging on connector1 5. Wait 120 seconds 6. Remove car (testtool) from connector 7. Connect car (testtool) to connector1 8. Read Idtag1 on charger's RFID reader, start charging on connector1 9. Wait 60 seconds 10. Remove car (testtool) from connector 11. Change StopTransactionOnInvalidId to False 12. Make Idtag1valid in CDMW 13. Connect car (testtool) to connector1 14. Read Idtag1 on charger's RFID reader, start charging on connector1 15. Read Idtag1 on charger's RFID reader, stop charging on connector1 16. Remove car (testtool) from connector 17. Make Idtag1invalid in CDMW 18. Connect car (testtool) to connector1 19. Read Idtag1 on charger's RFID reader, start charging on connector1 20. Wait 120 seconds 21. Read Idtag1 on charger's RFID reader, stop charging on connector1 22. Remove car (testtool) from connector 23. Make Idtag1valid in CDMW 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CS shall send changeconfig to charger with new value True on key StopTransactionOnInvalidId 2. No authorize shall be sent when starting transaction 3. Because key StopTransactionOnInvalidId is set to true. And Idtag1 is valid in cache for charger but invalid in CS then charger shall send stoptransaction request after received Invalid in startresponse 4. Key shall be removed from charger cache or be flagged as invalid in charger's cache 5. Because key is invalid in charger's cache, charger shall send authorize next time Idtag1 is used. Authorize response shall be invalid and no start be sent 6. After Idtag1 is made valid in CS a successful transaction shall happen and key is now back as valid in charger's cache memory 7. After Idtag1 is again made invalid on CS side and StopTransactionOnInvalidId is set to False. Charger shall starttransaction without authorize but ignore invalid response from CS, it shall not stop transaction. 	

	8. It shall still be possible to stop transaction normally with Idtag1 or remotestop or stop from car	
Actual Results/Comments		

2.1.11 Start transaction with car NOT connected when authenticating RFID		Status
MANDATORY REQUIREMENT		
Requirement	Charger shall be able to start and stop transaction on all connectors with RFID	
Procedure	<ol style="list-style-type: none"> 1. Read Idtag1 on charger's RFID reader, start charging on connector1 2. Connect car to connector1 3. Read Idtag1 on charger's RFID reader, stop charging on connector1 4. Remove Car (simulator) from connector 5. Repeat step 1-4 on all connectors 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP shall send status preparing to CS 2. CP shall send start transaction on Connector1 with Idtag1 3. CP shall send status charging on Connector1 (and status Unavailable on connectors that is not able to charge while connector1 is charging) 4. CP shall send stop transaction to CS 5. CP shall send status Finishing and unlock cable 6. CP shall send status Available on Connector1 (and all connectors that connector1 blocked) after car (testtool) has been removed 	
Actual Results/Comments		

2.1.12 Start transaction with car connected when authenticating RFID		Status
MANDATORY REQUIREMENT		
Requirement	Charger shall be able to start and stop transaction on all connectors with RFID	
Procedure	<ol style="list-style-type: none"> 1. Change config key LocalPreAuthorize to false 2. Connect car to connector1 3. Wait 120 seconds 4. Read Idtag1 on charger's RFID reader, start charging on connector1 5. Read Idtag1 on charger's RFID reader, stop charging on connector1 6. Remove car (testtool) from connector 7. Repeat step 2-5 on all connectors 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CS shall send changeconfig to charger with new value false on key LocalPreAuthorize 2. CP shall send accepted 	

	<ol style="list-style-type: none"> 3. CP shall send status preparing to CS and it shall NOT change status to available after 60 seconds 4. CP shall send Authorize with idtag1 5. CS shall respond with accepted 6. CP shall send start transaction on Connector1 with Idtag1 7. CP shall send status charging on Connector1 (and status Unavailable on connectors that is not able to charge while connector1 is charging) 8. CP shall send stop transaction to CS 9. CP shall send status Finishing and unlock cable 10. CP shall send status Available on Connector1 (and on all connectors that connector1 blocked) after car (testtool) has been removed 	
Actual Results/Comments		

2.1.13 Reset while charging		Status
MANDATORY REQUIREMENT		
Requirement	When resetting while connector is charging then charger shall stop charging send status unavailable and send stop transaction	
Procedure	<ol style="list-style-type: none"> 1. Read Idtag1 on charger's RFID reader, start charging on connector1 2. Connect car to connector1 3. Wait until start transaction response is sent. 4. Send soft reset to charger 5. Wait until charger have booted 6. Remove car (testtool) from connector 7. Read Idtag1 on charger's RFID reader, start charging on connector1 8. Connect car to connector1 9. Wait until start transaction response is sent. 10. Send hard reset to charger 11. Wait until charger have booted 12. Remove car (testtool) from connector 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP shall send status preparing to CS 2. CP shall send start transaction on Connector1 with Idtag1 3. CP shall send status charging on Connector1 (and status Unavailable on connectors that is not able to charge while connector1 is charging) 4. CP shall send stop transaction to CS 5. CP shall send status Unavailable and unlock cable 6. CP shall do a soft reset 7. (CP shall send bootnotification IF new Websocket was used to establish connection) 8. CP should send status Finish on Connector1 (and current status on other connectors) 9. CP shall send status Available after test tool has been removed 10. CP shall send status preparing to CS 11. CP shall send start transaction on Connector1 with Idtag1 	

	12. CP shall send status charging on Connector1 (and status Unavailable on connectors that is not able to charge while connector1 is charging) 13. CP shall send stop transaction to CS 14. CP shall send status Unavailable and unlock cable 15. CP shall do a hard reset 16. CP shall send bootnotification 17. CP should send status Finish on Connector1 (and current status on other connectors) 18. CP shall send status Available after test tool has been removed	
Actual Results/Comments		

2.1.14 Metervalues		Status
MANDATORY REQUIREMENT		
Requirement	Charger shall be able to send Metervalues with correct information to CS	
Procedure	<ol style="list-style-type: none"> 1. Change key MeterValueSampleInterval to 60 2. Change key MeterValuesSampledData to all measurands that charger support 3. Change key StopTxnAlignedData to all measurands that charger support 4. Change key StopTxnSampledData to all measurands that charger support 5. Read Idtag1 on charger's RFID reader, start charging on connector1 6. Let it charge for 600 Seconds 7. Go to CP page in CDMW and press stop 8. Wait 120 seconds 9. Remove car (testtool) from connector 10. Change key MeterValueSampleInterval to 0 11. Read Idtag1 on charger's RFID reader, start charging on connector1 12. Let it charge for 600 Seconds 13. Go to CP page in CDMW and press stop 14. Wait 120 seconds 15. Remove car (testtool) from connector 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. Changeconfig shall be sent to charger and set value 60 on config key MeterValueSampleInterval 2. CP shall accept 3. Changeconfig shall be sent to charger and set all supported measurands on config key MeterValuesSampledData 4. CP shall accept 5. Changeconfig shall be sent to charger and set all supported measurands on config key StopTxnAlignedData 6. CP shall accept 7. Changeconfig shall be sent to charger and set all supported measurands on config key StopTxnSampledData 8. CP shall accept 9. CP shall send Authorize with idtag1 10. CS shall respond with accepted 	

	11. CP shall send start transaction on Connector1 with Idtag1 12. CP shall send status charging on Connector1 (and status Unavailable on connectors that is not able to charge while connector1 is charging) 13. CP shall send 10 Metervalues during these 600 seconds while charging 14. CS shall send remotestop with correct TransactionID 15. CP shall accept, stop charging and send stop transaction with that TransactionID 16. CP shall send status Finishing and unlock cable 17. CP shall not send any more Metervalues during the wait time of 120 seconds 18. CP shall send status available on connector1 after testtool was removed (and all connectors that was blocked by charging on connector1) after car (testtool) was removed 19. Changeconfig shall be sent to charger and set value 0 on config key MeterValueSampleInterval 20. CP shall accept 21. CP shall send Authorize with idtag1 22. CS shall respond with accepted 23. CP shall send start transaction on Connector1 with Idtag1 24. CP shall send status charging on Connector1 (and status Unavailable on connectors that is not able to charge while connector1 is charging) 25. CP shall NOT send any Metervalues during these 600 seconds while charging 26. CS shall send remotestop with correct TransactionID 27. CP shall accept, stop charging and send stop transaction with that TransactionID 28. CP shall send status Finishing and unlock cable 29. CP shall not send any more Metervalues during the wait time of 120 seconds 30. CP shall send status available on connector1 (and all connectors that was blocked by charging on connector1) after car (testtool) was removed	
Actual Results/Comments		

2.1.15 Emergency Stop		Status
MANDATORY REQUIREMENT		
Requirement	If charger has an emergency stop and user press it CP shall always stop ongoing transaction and unlock the connector	
Procedure	1. Connect car to connectorID1 2. Go to the charge point page in CDMW and press start transaction. 3. Wait for meter value. 4. Press Emergency button on charger. 5. Release the button. 6. Remove the testtool. 13.	
Start date UTC time		
End date UTC time		
Expected results	1. When emergency button pressed CP must stop ongoing transaction.	

	<ol style="list-style-type: none"> 2. Stop transaction must contains the reason for stop ongoing transaction. 3. CP shall unlock the Connectors. 4. CP should send Status unavailable on all connectors. 5. On releasing emergency button CP must update status to available on all connectors. 	
Actual Results/Comments		

2.1.16 Power Failure in Online Mode		Status
MANDATORY REQUIREMENT		
Requirement	CP shall always stop ongoing transaction and unlock the connector.	
Procedure	<ol style="list-style-type: none"> 1. Connect car(s) to all available connectors. 2. Go to the charge point page in CDMW and press start transaction on all connectors 3. Wait for metre value. 4. Switch off the mains, simulate grid powerfailure (or tripp the RCBO if available) 5. Connector lock should be in Unlock state. Disconnect car 6. Wait for 30 sec and switch on power supply. 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. When switch off input mains CP must stop ongoing transactions 2. Stop transaction must contains the reason for the stop 3. CP shall unlock the Connectors. 4. On powering up CP shall send bootnotification 	
Actual Results/Comments		

2.1.17 Offline testing (with powerfailure)		Status
PREFERENTIAL REQUIREMENT		
Requirement	<p>CP shall be able to be in standalone mode without the help of CS in cases where CS is unavailable.</p> <p>When CP is online again then all transaction messages made offline shall be sent to CS in chronological order</p>	
Procedure	<ol style="list-style-type: none"> 1. Change config key AllowOfflineTxForUnknownId to false 2. Change config key LocalAuthorizeOffline to true 3. Change config key MeterValueSampleInterval to 100 4. Change config key StopTransactionOnEVSideDisconnect to true 5. Change config key UnlockConnectorOnEVSideDisconnect to true 6. Connect car (testtool) to connector1 7. Read Idtag2 on charger's RFID reader 8. Wait until charging starts 9. Wait 30 seconds 10. Disconnect Internet from charger by example removing ethernet cable 	

	<ol style="list-style-type: none"> 11. Try to stop the ongoing transaction by reading Idtag1 12. Try to stop the ongoing transaction by reading Idtag3 13. Try to stop the ongoing transaction by reading Idtag2 14. Start new Transaction with Idtag1 on connector1 15. IF possible, Start new transaction on connector2 with Idtag2 16. Try to stop transaction on connector1 with Idtag2 17. Try to stop transaction on connector2 with Idtag2 18. Stop transaction on connector1 from car (testtool) 19. Remove Car (testtool) from both connectors 20. Insert Car (testtool) to connector1 21. Try to start transaction on connector1 with Idtag3 22. Try to start transaction on connector1 with Idtag1 23. Try to stop transaction on connector1 with Idtag1 24. Insert Car (testtool) to connector2 25. Try to start transaction on connector2 with Idtag1 26. Simulate grid power failure, remove power connector from charger, charger should die. 27. Insert powercable to charger and let charger boot up in offline mode 28. Try to start transaction on connector1 with Idtag1 29. Wait 600 seconds 30. Make charger online 31. Wait for boot and transaction messages sent to CS 32. Try to stop transaction on connector1 with idtag2 33. Try to stop transaction on connector1 with idtag1 34. Remove car (testtool) 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. Config key AllowOfflineTxForUnknownId is set to false 2. Config key LocalAuthorizeOffline is set to true 3. Config key MeterValueSampleInterval is set to 100 4. Config key StopTransactionOnEVSideDisconnect is set to true 5. Config key UnlockConnectorOnEVSideDisconnect is set to true 6. CP shall send status Preparing on connector1 to CS 7. CP shall send authorize with Idtag2, CS sends accepted 8. CP shall send starttransaction on connector1 with Idtag2, CS send accepted 9. CP shall send status charging on connector1 10. Ongoing transaction should only be able to stop by reading Idtag2 11. Idtag1 should be able to start new transaction (because it was authorized in an earlier testcase) on connector1 (connector1) 12. If charger supports charging on several connectors at the same time then charging shall start on another connector (connector2) with Idtag2 13. Connector2 should only be able to stop with Idtag2 14. It shall always be possible to stop charging from car. Connector1 should stop transaction when stopping from car 15. You shall not be able to start new transaction with idtag3 16. Connector1 shall be able to start with Idtag1 17. Transaction on Connector1 shall be able to stop with Idtag1 18. Idtag1 shall be able to start new transaction on connector2 19. If power failure occur, charging shall stop, cable SHALL unlock, then when power is restored charger shall be able to continue operation as normal. Write in comments, does this charger support continuing of transaction after power failure? 20. After charger has restored power and has booted in offline mode, Idtag1 and Idtag2 should still be able to start new transactions on connectors 	

	21. When charger is online, then the first thing it shall do is to send a bootnotification 22. After that CP shall send all transaction messages with correct information that happen while offline in chronological order (at least 6 Metervalues shall be sent) 23. After this is done it shall send status Charging on connector1 and the status on the other connectors 24. Transaction on connector1 shall only be able to stop with Idtag1 25. After stop charger shall send status Finishing on connector 1 and then send Available after car (testtool) is removed	
Actual Results/Comments		

2.1.18 Offline testing (without powerfailure)		Status
PREFERENTIAL REQUIREMENT		
Requirement	CP shall be able to be in standalone mode without the help of CS in cases where CS is unavailable. When CP is online again then all transaction messages made offline shall be sent to CS in chronological order	
Procedure	<ol style="list-style-type: none"> 1. Connect car (testtool) to connector1 2. Read Idtag2 on charger's RFID reader 3. Wait until charging starts 4. Wait 30 seconds 5. Disconnect Internet from charger by example removing ethernet cable 6. Try to stop the ongoing transaction by reading Idtag2 7. Start new Transaction with Idtag1 on connector1 8. Try to stop transaction on connector1 with Idtag1 9. Try to start transaction on connector1 with Idtag2 10. Wait 30 seconds 11. Make charger online 12. Wait for bootnotification and transaction messages sent to CS (bootnotification may not be required if charger use the same websocket as before) 13. Try to stop transaction on connector1 with idtag1 14. Try to stop transaction on connector1 with idtag2 15. Remove car (testtool) 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP shall send status Preparing on connector1 to CS 2. CP shall send authorize with Idtag2, CS sends accepted 3. CP shall send starttransaction on connector1 with Idtag2, CS send accepted 4. CP shall send status charging on connector1 5. Ongoing transaction should only be able to stop by reading Idtag2 6. Idtag2 shall be able to start new transaction on Connector1 7. After CP is online then CP shall send all transaction messages with correct information that happen while offline in chronological order (new bootnotification may not be required if charger use the same websocket as when it was online) 8. After this is done it shall send status Charging on connector1 and the current status on the other connectors 9. Transaction on Connector1 should only be able to stop by Idtag2 10. CP shall send status Finishing and unlock cable 	

	11. CP shall send status Available on connector1 after Car (testtool) is removed from Connector1	
Actual Results/Comments		

2.1.19 Remote Start / Remote Stop		Status
MANDATORY REQUIREMENT		
Requirement	It shall be possible to send remote start and stop to charger. Both before car has connected and after car has connected to connector. If charger has more than one connector then it should be possible to charge with remotestart on more then one connector (DC chargers might not support that)	
Procedure	<ol style="list-style-type: none"> 1. Change configkey AuthorizeRemoteTxRequests to true 2. Change configkey LocalPreAuthorize to false 3. Go to CP page in CDMW, press start on connector1 4. Wait until timeout occurs 5. Press start on connector1 6. Connect car (testtool) to connector1 7. IF possible, press start on connector2 (or another connector that support simultaneous charging) 8. Connect car (testtool) to that connector 9. Press stop on connector1 10. Press stop on the other connector 11. Remove car (testtool) from the other connector 12. Press start on connector1 13. IF started then press stop 14. Remove car (testtool) from connector1 15. Connect car (testtool) to connector1 16. Press start on connector1 17. Press stop on connector1 18. Remove car (testtool) from connector1 19. Do a remote start / remote stop and charge from all connectors (one at a time) 20. Reapeat steps 3-18 but with the difference that configkey AuthorizeRemoteTxRequests is set to false 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP shall accept remote start and send status preparing on connector1 2. CP shall timeout after 60 seconds and send status available on connector1 3. CP shall accept remotestart and send status preparing on connector1 4. CP shall send starttransaction with Idtag declared in remotestart and send status charging (and send status unavailable on connectors that transaction on this connector blocks) 5. IF remotestart was initialized for another connector then same thing shall happen to that connector as connector1 6. CP shall send accepted on remote stop 7. CP shall send stop transaction with correct TransactionID 8. CP shall send Status Finishing after stop 9. IF CP receives a remotestart on a connector that is in Finishing status then expected is that charger sends rejected on that request. 	

	10. After car (testtool) is removed from connector CP should send status Available	
Actual Results/Comments		

2.1.20 Unlock connector/Stop transaction from car		Status
MANDATORY REQUIREMENT		
Requirement	If connector has lock, then it should always be possible to free a car that's trapped in the lock.	
Procedure	<ol style="list-style-type: none"> 1. Press unlock on connector1 from CP page in CDMW 2. Connect EV car to connector1 (or a connector with lock functionality) 3. Press unlock on connector from CP page in CDMW 4. Press Start on connector from CP page in CDMW 5. Press Unlock 6. Remove car (testtool) from connector 7. Connect EV car to connector 8. Start by using Idtag1 9. Press unlock on that connector 10. Remove car (testtool) from connector 11. Connect EV car to connector 12. Start by using Idtag1 13. Stop transaction from car 14. Remove car (testtool) from connector 15. Repeat step 1-13 on all connectors 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP should reply UnlockFailed because there was no car connected that could be unlocked or if charger doesn't support Unlock it shall reply NotSupported 2. CP shall NOT lock cable to connector in status preparing 3. CP should lock cable in status charging or status SuspendedEV 4. CP shall stop ongoing transaction if unlock is supported and do standard stop procedure (send stop and go to status Finishing) and if charger was able to unlock then reply with Unlocked or if failed to unlock send UnlockFailed 5. When transaction is stopped from car then CP shall unlock the connector car was connected to 	
Actual Results/Comments		
Questions		
Answers		

2.1.21 Freemode		Status
MANDATORY REQUIREMENT		
Requirement	When freemode is activated, then CP should work as normal with only small different behaviours	

	<p>1, all starttransactions CP sends shall use a default IDtag that you can set in a RW config key.</p> <p>2, CP should start a new transaction as soon car (testtool) is connected</p> <p>There shall be a config key that you can change to set if charger locks cable in freemode or not. If config key doesn't exist then it shall not lock cable in freemode</p> <p>In freemode charger shall ignore invalid response in starttransaction regardless of what the StopTransactionOnInvalidId is in for value</p>	
Procedure	<ol style="list-style-type: none"> 1. Turn on freemode with the function that charger does not lock connector in a new transaction by setting the correct values to config keys (set a default idtag that is valid in CDMW). 2. Connect car (testtool) to connector1 3. Stop transaction from car 4. Set StopTransactionOnInvalid to true. 5. Change the config key values to lock the cable in freemode. 6. Change the default idtag to an invalid tag. 7. Send remote start to charger 8. Plug in cable (testtool) to connector1(or if this charger have more then 1 connector plug it in another connector) 9. Send remote stop to charger 10. Wait 60 seconds 11. Remove cable (testtool) 12. Read (if you can) idtag1 on chargers rfid reader 13. Connect cable (testtool) 14. Read (if you can) idtag1 on chargers rfid reader 15. Stop charging and remove cable testtool 16. Repeat testcase 2.1.16 Offline testing (without powerfailure) in freemode 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. Config keys shall be accepted 2. Charger shall send starttransaction with default IDtag on connector1 and NOT lock cable 3. Charger shall send stoptransaction request containing same RFID tag and same transactionID that was in starttransaction 4. Change of config key StopTransactionOnInvalid to True should be accepted by charger 5. Config keys should be accepted 6. Starttransaction shall be sent to CS with the default invalid idtag set and cable shall lock 7. CS shall send invalid 8. Charger shall ignore invalid response and continue charging 9. Remote stop should be accepted and charger should go to finishing mode (not starting new transaction) 10. When removing cable charger shall go to available 11. Charger shall send starttransaction with default idtag 12. Charger should only be able stop transaction from car (testtool) or remotestop not with read RFID tag (exception being if the RFID tag read is the same as the default idtag) 13. Offline in freemode shall be a success 	
Actual Results/Comments		

Questions		
Answers		

2.1.22 Smart Charging		Status
MANDATORY REQUIREMENT		
Requirement	Charger should be able to handle smart chargings TxProfile and TxDefaultProfile Fortum also requires that Measurand Current.Offered to be supported in MeterValues	
Procedure	<ol style="list-style-type: none"> 1. Enable Current.Offer to be sent in metervalues 2. Set MeterValueSampledInterval to 30 or 60 Seconds 3. Using AutoValidator tool, while testing charger, click on command and then click on limit current on connector. 4. Send a limit at 6 A to all connectors while there is no charging ongoing 5. Start charging on all available connectors with IDtag1 6. Wait for 2 metervalues 7. Set limit to 0 A on all Connectors 8. Wait 2 metervalues 9. Set limit to 8 A on all Connectors 10. Wait 1 metervalue 11. Using your simulator tool, be in state C and draw more power then 8 A until CurrentOverload is detected, do this on all connectors 12. Start new transaction on all connectors with IDtag1 13. Wait 2 metervalue 14. Set limit to 0 A on all Connectors 15. Stop transactions with IDtag1 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. Charger shall accept Current.Offered 2. Charger shall accept MeterValueSampledInterval 3. 4. TxDefaultProfile shall be accepted with value 6 5. Starttransaction shall be sent on all available connectors 6. Charger shall send (2 * number of charging connectors) metervalues containing Current.Offered with value 6 7. TxProfile will be sent and charger shall accept limit 0, this will cause charger to release contactor and pause transaction with status SuspendedEVSE 8. Charger shall send (2 * number of charging connectors) metervalues containing Current.Offered with value 0 9. TxProfile will be sent and charger shall accept limit 8, this will cause charger to release contactor and pause transaction with status SuspendedEVSE 10. 11. Charger shall stop the transactions because transaction is drawing more power then the limit allows 12. Transaction shall start 13. Current.Offer shall report 6 14. Charger shall accept limit 0 15. Transactions shall stop 	
Actual Results/Comments		

2.1.23 Heartbeat		Status
MANDATORY REQUIREMENT		
Requirement	Charger SHALL send a HeartBeat pdu to CS to ensure that the charger is still alive.	
Procedure	16. Keep the charger online 24 hours 17. Charger might utilize heartbeat skip and then we want to see that charger sends heartbeats while metervalues is off and charger is in a charging state. Change metervalues to 0. 18. Heartbeat shall be send in ongoing transaction, so start a transaction on all connectors that you can do it on. And let transaction stay on for around 2 hours. 19. Stop ongoing transactions	
Start date UTC time		
End date UTC time		
Expected results	16. See if the heartbeat was sent in correct time, if the heartbeat interval was set to 240 then it should send every 4 minutes	
Actual Results/Comments		

2.1.24 Get Diagnostics		Status
MANDATORY REQUIREMENT		
Requirement	Charger shall be able to send diagnostic information to a FTP server	
Procedure	1. Press Get Diagnostics in CP page in CDMW	
Start date UTC time		
End date UTC time		
Expected results	1. Charger shall reply with a filename, the filename must have a unique identifier in it, like the current date and time 2. Charger shall send DiagnosticsStatusNotification and let CS know what happens It shall use these statuses: "Idle", "Uploaded", "UploadFailed", "Uploading"	
Actual Results/Comments		

2.1.25 Firmware update MANDATORY REQUIREMENT		Status
Requirement	Charger shall be able to update its FW. It shall also be possible to downgrade software to a previous version	
Procedure	<ol style="list-style-type: none"> 1. Upgrade to a newer FW version by using FW upgrade from CP page in CDMW 2. Downgrade to the current FW version that we are testing 3. Downgrade to a previous FW version 4. Upgrade to the current FW version that we are testing 	
Start date UTC time		
End date UTC time		
Expected results	<ol style="list-style-type: none"> 1. CP shall send FirmwareStatusNotificationRequest and use these statuses "Downloaded", "DownloadFailed", "Downloading", "Idle", "InstallationFailed", "Installing", "Installed" 2. firmwareVersion in Bootnotification shall change to the FW name we updated/downgraded to 3. CP should do a hard reset after update is complete 	
Actual Results/Comments		

3 Observations

Please note down any observation made during testing that might not fit on testcases comments:

Observations

4 Questions to manufacturer

Questions