



OCPP 2.0.1 Edition 3
Errata 2025-04

Table of Contents

Disclaimer	1
Scope	2
Terminology and Conventions	2
0. Part 0	3
1. Part 1	4
1.1. Page 9 - (2025-01) - 3.4. Monitoring	4
2. Part 2	5
2.1. Page 5 - (2025-01) - Updated limitations on BasicAuthPassword to increase security	5
2.2. Page 14 - (2025-04) - Section 2.7. ISO 15118 support - Aligned ISO 15118 timeout table with updated version in OCPP	
2.1	6
2.3. Page 15 - (2025-01) - Improved text FR.04	6
2.4. Page 19 - (2025-01) - Removed requirement A00.FR.003 as the precondition never occurs	6
2.5. Page 19/41 - (2025-01) - Allow downgrading security profile from 3 to 2	7
2.6. Page 26 - (2025-01) - 1.3.7. TLS with Client Side Certificates Profile - Requirements	9
2.7. Page 28 - (2025-01) - Certificate Hierarchy	9
2.8. Page 31 - (2025-01) - A02/A03 - Updated error handling	10
2.9. Page 34 - (2025-01) - Disallow client certificates future validity date	10
2.10. Page 58/60 - (2025-01) - Missing requirement information about omitting the value for WriteOnly variables	13
2.11. Page 69 - (2025-01) - B11 - Reset without Ongoing Transaction - Requirements	14
2.12. Page 73 - (2025-01) - B12 - Reset with Ongoing Transaction - Requirements	14
2.13. Page 77 - (2025-01) - 1.6 Relationship between authorization and transaction	15
2.14. Page 80 - (2025-01) - C01 - EV Driver Authorization using RFID - Requirements	15
2.15. Page 80 - (2025-01) - C01 - EV Driver Authorization using RFID - Requirements	16
2.16. Page 99 - (2025-01) - C09- Authorization by GroupId - Requirements	16
2.17. Page 101 - (2025-01) - Updated requirements related to clarify the relation between AuthCacheLifeTime and cacheExpiryDateTime	16
2.18. Page 113 - (2025-01) - C16 - Stop Transaction with a Master Pass - Requirements	17
2.19. Page 129 - (2025-01) - Updated sequence diagram E01 S5	17
2.20. Page 151 - (2025-02) - E06.FR.05 for DataSigned as TxStopPoint is invalid	18
2.21. Page 152 - (2025-01) - E07 - Improved scenario description names	18
2.22. Page 154 - (2025-01) - E07 - Transaction locally stopped by IdToken	19
2.23. Page 198 - (2025-01) - G01 - Status Notification - Requirements	20
2.24. Page 197 - (2025-01) - G01 - Status Notification - State transition overview for connecting/disconnecting	21
2.25. Page 208 - (2025-01) - H. Reservation - Introduction	21
2.26. Page 213 - (2025-02) - H02 - Added missing requirements	21
2.27. Page 214/215 - (2025-01) - Improved use case scenario descriptions and added S3	23
2.28. Page 226 - (2025-01) - I06.FR.02 Improved requirement text	26
2.29. Page 231 - (2025-01) - Updated section Multiple Locations/Phases	26
2.30. Page 243 - (2025-01) - Improved section on external Smart Charging Control Signals	27
2.31. Page 248 - (2025-01) - 3.7 Avoiding Phase Conflicts	28
2.32. Page 286 - (2025-01) - K15 - ISO 15118-2 Charging with load leveling - Requirements	28
2.33. Page 286 - (2025-01) - K16 - Renegotiation initiated by CSMS - Requirements	29
2.34. Page 292 - (2025-01) - Use case L01 - Added clarification to step 3 about when to start downloading the firmware	29
2.35. Page 259/260 - (2025-01) - K03 - Updated use case description and sequence diagram	29
2.36. Page 274/275/276/277 - (2025-01) - K11/K12 - Updated use case descriptions and sequence diagrams	32
2.37. Page 278/279 - (2025-01) - K13 - Updated requirement preconditions	34
2.38. Page 279 - (2025-01) - K14 - Updated use case scenario description	35
2.39. Page 306 - (2025-01) - M. ISO 15118 Certificate Management	35
2.40. Page 308 - (2025-01) - Update introduction sequence diagram ISO 15118	35
2.41. Page 330 - (2025-01) - N03 Set Monitoring Base: Improved text of Remark	37
2.42. Page 331 - (2025-01) - N03.FR.04: text improvement	37
2.43. Page 350 - (2025-02) - O01 - Added missing requirements	37
2.44. Page 446 - (2025-01) - ActiveNetworkProfile is incorrectly marked as optional	38
2.45. Page 453 - (2025-01) - References to <i>monitorValue</i> changed to <i>value</i> [354]	38
2.46. Page 454 - (2025-01) - N04.FR.06 Improved limit definition of thresholds [353]	38
2.47. Page 456 - (2025-02) - New configuration variable to allow TLS wildcard certificates	39

2.48. Page 458 - (2025-01) - Added optional variable to allow the Charging Station to report its supported idTokenType.....	39
2.49. Page 462 - (2025-01) - Added note to EnergyTransfer description as TxStartPoint	40
2.50. Page 476 - (2025-01) - Added Connector component to AvailabilityState referenced variable	40
3. Part 3	41
4. Part 4	42
4.1. Page 6 - (2025-01) - 3.1.1. The connection URL.....	42
4.2. Page 7 - (2025-01) - 3.1.2. OCPP version	42
4.3. Page 8 - (2025-01) - 3.3. WebSocket Compression	42
4.4. Page 10 - (2025-01) - 4.1.3. The message type	42
4.5. Page 10 - (2025-01) - 4.1.3. The message type	42
4.6. Page 10 - (2024-09) - 4.1.4. Message ID	43
4.7. Page 13 - (2025-01) - 4.2.3. CALLERROR.....	43
4.8. Page 15 - (2025-01) - 5. Connection.....	43
4.9. TLS fragment length	43
4.10. Page 15 - (2025-01) - 5.3. WebSocket Ping in relation to OCPP Heartbeat.....	44
4.11. Page 15 - (2025-01) - 5 Connection - Added section about TLS fragment length	44
4.12. Page 16 - (2025-04) - 5.3 Reconnecting - reset backoff wait timer	44
4.13. Page 18 - (2025-02) - 6.3 Connection loss - Allow Local Controller to keep connection open.....	44
5. Part 5	46
5.1. General - (2025-02) - Renamed OCTT to Test System.....	46
5.2. Page 7 - (2025-02) - Changed definition of C-01 Support for offline authorization of transactions.....	46
5.3. Page 9 - (2024-11) - Optional feature list for charging station - Change name R-3	46
5.4. Page 9 - (2025-02) - Optional feature list for charging station - Added AQ-10 to make TC_N_48_CS conditional.....	47
5.5. Page 9/11/27/48/52 - (2025-02) - Removed feature ISO-3, added additional question AQ-11	47
5.6. Page 13 - (2025-02) - Updated Id's of the additional questions of CSMSs in the appendix	48
5.7. Page 17/33 - (2025-02) - Test Cases Local Authorization List Management & Authorization Cache - Update conditions.....	48
5.8. Page 19 - (2024-09) - TC_E_04_CS Updated condition for test case to exclude it for MacAddress and ISO 15118 PnC	52
5.9. Page 22 - (2024-09) - TC_E_17_CS Updated condition for test case to correctly specify the applicable TxStopPoint combinations	52
5.10. Page 39 - (2025-02) - TC_N_48_CS Made conditional	53
5.11. Page 40 - (2024-11) - TC_H_13_CS Updated invalid condition for test case	53
5.12. Page 42 - (2025-02) - Removed TC_O_15_CS from certification program	53
5.13. Test Cases Advanced User Interface.....	54
5.14. Page 44 - (2025-02) - Removed TC_A_13_CS and TC_A_13_CSMS from certification program	54
5.15. Page 48 - (2024-06) - Added additional questions to appendix	54
5.16. Page 48 - (2025-02) - Updated Id's of the additional questions for CSMSs in the appendix	54
5.17. Page 48 - (2025-04) - Duplicate AQ-11 id	55
6. Part 6	56
6.1. General.....	56
6.1.1. Page XX - (2024-11) - All testcases - Updated table structure of all testcases.....	56
6.1.2. Page XX - (2025-02) - Renamed OCTT to Test System	56
6.2. Charging Station.....	56
6.2.1. Page 4 - (2025-02) - TC_A_01_CS - Updated old identifierString reference in description.....	56
6.2.2. Page 7 - (2024-11) - TC_A_05_CS - Successfully reconnecting after every failed connection attempt	56
6.2.3. Page 7 - (2025-02) - TC_A_05_CS	57
6.2.4. Page 7 - (2025-04) - TC_A_05_CS - Updated before steps to take into account the AllowCSMSTLSWildcards variable	58
6.2.5. Page 20 - (2025-02) - TC_A_23_CS - CSMS returns a CertificateSigned message for each request	58
6.2.6. Page 22 - (2024-09) - TC_A_19_CS - Fixed references to ConfigurationSlot [020-4762]	59
6.2.7. Page 24 - (2025-02) - TC_A_20_CS - Testcase did not take into account that the used configuration slot could already be set.....	60
6.2.8. Page 24 - (2025-04) - TC_A_20_CS - SetNetworkConnectionProfile is allowed to be rejected	61
6.2.9. Page 25 - (2025-02) - TC_A_21_CS.....	62
6.2.10. Page 26 - (2025-02) - TC_A_22_CS.....	62
6.2.11. Page 51 - (2025-02) - TC_B_16_CS - Correctly validate result of reading WriteOnly component variables	63
6.2.12. Page 56 - (2024-09) - TC_B_20_CS - Added check on omitting evseld [4390]	63
6.2.13. Page 57 - (2024-09) - TC_B_21_CS - Added check on omitting evseld [4390]	64
6.2.14. Page 58 - (2024-09) - TC_B_22_CS - Added check on omitting evseld [4390]	64
6.2.15. Page 63 - (2025-02) - Changed reset to Immediate	64

6.2.16. Page 85 - (2025-04) - TC_B_53_CS - Check if all required values are provided.	65
6.2.17. Page 89 - (2025-02) - Add setting of NetworkProfileConnectionAttempts	65
6.2.18. Page 117/141 - (2025-02) - Added remote support for Authorization Cache & Local Authorization List testcases	66
6.2.19. Page 101 - (2025-04) - TC_C_14_CS - Fixing invalid component variable reference	67
6.2.20. Page 102 - (2025-04) - TC_C_15_CS - Fixing invalid component variable reference	67
6.2.21. Page 104 - (2025-04) - TC_C_16_CS - Fixing invalid component variable reference	67
6.2.22. Page 105 - (2025-04) - TC_C_17_CS - Fixing invalid component variable reference	67
6.2.23. Page 108 - (2025-04) - TC_C_18_CS - Fixing invalid component variable reference	67
6.2.24. Page 120 - (2025-04) - TC_C_34_CS - Making use of the Deauthorized reusable state	68
6.2.25. Page 128 - (2025-04) - TC_C_41_CS - Fixing invalid component variable reference	68
6.2.26. Page 134 - (2025-04) - TC_C_44_CS - Fixing invalid component variable reference	68
6.2.27. Page 137 - (2025-04) - TC_C_46_CS - Fixing invalid component variable reference	68
6.2.28. Page 138 - (2024-09) - TC_C_47_CS - StoppedReason must be validated in Ended event [O20-4467]	69
6.2.29. Page 145 - (2025-04) - TC_C_25_CS - Fixing invalid component variable reference	69
6.2.30. Page 146 - (2024-11) - TC_C_26_CS - Allow StatusNotification status = Occupied	69
6.2.31. Page 147-153 - (2025-04) - TC_C_50_CS, TC_C_51_CS, TC_C_52_CS, TC_C_53_CS, TC_C_54_CS, TC_C_55_CS - Always re-install V2G certificates	70
6.2.32. Page 147 - (2025-02) - TC_C_50_CS	70
6.2.33. Page 148 - (2025-02) - TC_C_51_CS	71
6.2.34. Page 149 - (2025-02) - TC_C_52_CS	71
6.2.35. Page 150 - (2025-02) - TC_C_53_CS	72
6.2.36. Page 151 - (2024-09) - TC_C_54_CS - removed reusable state IdTokenCached [O20-3510]	73
6.2.37. Page 151 - (2025-02) - TC_C_54_CS	73
6.2.38. Page 153 - (2024-09) - TC_C_55_CS - removed reusable state IdTokenCached [O20-3510]	74
6.2.39. Page 153 - (2025-02) - TC_C_55_CS	74
6.2.40. Page 165/169 - (2025-02) - TC_E_01_CS/TC_E_09_CS/TC_E_10_CS/TC_E_12_CS/TC_E_13_CS - Extended the testcases until the Charging Station start charging	75
6.2.41. Page 174 - (2025-04) - TC_E_17_CS - Aligned configuration before steps with updated prerequisites	76
6.2.42. Page 174 - (2024-09) - TC_E_17_CS - Updated prerequisite for test case to correctly specify the applicable TxStopPoint combinations	77
6.2.43. Page 176 - (2024-11) - TC_E_39_CS - Missing StatusNotificationRequest/NotifyEventRequest	77
6.2.44. Page 182 - (2025-04) - TC_E_52_CS - Testcase is not able to determine the authorization is refused in case TxStartPoint is not Authorized	78
6.2.45. Page 185 - (2024-09) - TC_E_35_CS - StoppedReason must be validated in Ended event [O20-4467]	78
6.2.46. Page 188 - (2025-02) - TC_E_22_CS - Stop transaction options - EnergyTransfer stopped - will end transaction	79
6.2.47. Page 199 - (2025-04) - TC_E_27_CS - Remove manual action between step 4/5	79
6.2.48. Page 204 - (2025-02) - TC_E_31_CS - Add steps for when running the testcase in Remote mode	80
6.2.49. Page 214 - (2024-06) TC_E_43_CS Move reusable state TransactionEventsInQueueEnded to Before [768]	81
6.2.50. Page 217 - (2025-04) - TC_E_46_CS - Testcase updated to use the specialized Authorized15118 reusable state	81
6.2.51. Page 221 - (2024-06) TC_F_04_CS Made mandatory in part 5, but prerequisite in part 6 was not updated	82
6.2.52. Page 221 - (2025-02) TC_F_04_CS Prerequisite only if supported	82
6.2.53. Page 295 - (2024-09) - TC_J_XX_CS Meter Values	82
6.2.54. Page 236 - (2025-02) - TC_F_19_CS - The testcase ends while the firmware update is still ongoing	83
6.2.55. Page 272 - (2025-02) - TC_H_08_CS Reserve an unspecified EVSE - Accepted	83
6.2.56. Page 282 - (2025-02) - TC_H_17_CS - made more explicit on what to validate	84
6.2.57. Page 297 - (2025-02) - TC_J_02_CS Clock-aligned Meter Values - reporting multiple phases	85
6.2.58. Page 297 - (2024-09) - TC_J_02_CS Clock-aligned Meter Values	85
6.2.59. Page 306 - (2024-06) - TC_J_10_CS - Remove reference to non-existing requirements [4697]	86
6.2.60. Page 318 - (2025-04) TC_K_09_CS: Removed validFrom/To from test	86
6.2.61. Page 343 - (2025-04) TC_K_23_CS: Removed validFrom/To from test	87
6.2.62. Page 337 - (2025-04) TC_K_28_CS: Removed validFrom/To from test	87
6.2.63. Page 345 - (2025-02) TC_K_XX_CS: Use realistic values for composite schedules	88
6.2.64. Page 345 - (2024-06) TC_K_35_CS Get Charging Profile - Evseld > 0 + chargingProfilePurpose [773]	89
6.2.65. Page 352 - (2025-04) - TC_K_39_CS - Validation of scheduleStart	89
6.2.66. Page 353 - (2025-04) - TC_K_40_CS: startSchedule improvements	90
6.2.67. Page 355 - (2025-04) - TC_K_41_CS: startSchedule improvement	92
6.2.68. Page 354/355 - (2025-02) - TC_K_40_CS & TC_K_41_CS - Updated composite schedule validation	93
6.2.69. Page 355 - (2025-04) - TC_K_41_CS: Added missing EnergyTransferStarted reusable state	95
6.2.70. Page 359 - (2025-04) - TC_K_53_CS: Added missing validations	96

6.2.71. Page 360 - (2025-02) - TC_K_54_CS: EVConnected must be before Authorization	97
6.2.72. Page 362 - (2025-02) - TC_K_56_CS: EVConnected must be before Authorization	98
6.2.73. Page 364 - (2025-02) - TC_K_57_CS: EVConnected must be before Authorization	98
6.2.74. Page 362 - (2024-09) - TC_K_56_CS Removed expecting triggerReason=ChargingRateChanged [776]	98
6.2.75. Page 366 - (2025-02) - TC_K_58_CS.....	99
6.2.76. Page 384 – TC_L_10_CS - Allow Download/InstallationFailed upon AcceptedCanceled	99
6.2.77. Page 387 - TC_L_06_CS - SecurityEventNotification and FirmwareStatusNotification can be sent in any order	100
6.2.78. Page 422 - (2025-02) - TC_M_15_CS - V2GCertificateChain is not installed before being retrieved	101
6.2.79. Page 436 - (2025-02) - TC_N_01_CS - Made used component variable configurable	101
6.2.80. Page 455 - (2025-02) - TC_N_12_CS - Updating test case for using configuration variables	102
6.2.81. Page 456 - (2025-02) - TC_N_13_CS - Updating test case for using more specific configuration variables.....	103
6.2.82. Page 463 - (2025-02) TC_N_20_CS - Updating test case for using more specific configuration variables.....	105
6.2.83. Page 468 - (2024-06) TC_N_23_CS Offline Notification - OfflineMonitoringEventQueuingSeverity set higher than severityLevel of the monitor [772].....	108
6.2.84. Page 470 - (2024-09) - TC_N_24_CS - Referring to incorrect use case and requirements [O20-4793]	109
6.2.85. Page 492 - (2025-02) - TC_N_39_CS - Test case now searches suitable variables to do test with	110
6.2.86. Page 472 - (2024-12) - TC_N_26_CS - Made test case more explicit and more time before ending	112
6.2.87. Page 470 - (2024-12) - TC_N_24_CS - Test case now searches suitable variable to do test with	113
6.2.88. Page 470 - (2025-02) - TC_N_24_CS - Updating test case for using more specific configuration variables.....	115
6.2.89. Page 482 - (2025-02) - TC_N_63_CS - Clear Customer Information - add manual action to stop session	116
6.2.90. Page 487 - (2025-04) - TC_N_36_CS - LogStatusNotification(AcceptedCanceled) allowed before GetLogResponse.....	117
6.2.91. Page 482 - (2025-04) - TC_N_63_CS - Added missing reusable state EnergyTransferStarted at before steps	118
6.2.92. Page 493 - (2024-09) - TC_N_41_CS - Set Variable Monitoring - Return to FactoryDefault.....	119
6.2.93. Page 482 - (2024-09) - TC_N_63_CS - Clear Customer Information - Clear and report - customerCertificate.....	119
6.2.94. Page 482 - (2025-02) - TC_N_63_CS - Added missing configuration state and authorize explicit using Plug and Charge (PnC)	120
6.2.95. Page 493 - (2024-09) - TC_N_41_CS - Set Variable Monitoring - Return to FactoryDefault.....	121
6.2.96. Page 493 - (2025-02) - TC_N_41_CS - Made less dependent on test case configuration variables, enables predefines monitors	121
6.2.97. Page 495 - (2024-11) - TC_N_43_CS - Remove incorrect tool validation StatusInfo	125
6.2.98. Page 495 - (2025-02) - TC_N_43_CS - Updating test case for using more specific configuration variables.....	125
6.2.99. Page 497 - (2025-02) - TC_N_45_CS - Updating test case for using more specific configuration variables.....	128
6.2.100. Page 501 - (2025-02) - TC_N_51_CS - Updating test case for using more specific configuration variables	130
6.2.101. Page 503 - (2025-02) - TC_N_52_CS - Updating test case for using more specific configuration variables	132
6.2.102. Page 504 - (2025-02) - TC_N_53_CS - Updating test case for using more specific configuration variables	134
6.2.103. Page 505 - (2025-02) - TC_N_56_CS - Made test case configurable using configuration variables	135
6.2.104. Page 520 - (2025-02) - TC_O_15_CS - Test case removed	136
6.2.105. Page 530 - (2025-02) - TC_O_28_CS - Transaction id should be specified for DisplayMessage	136
6.2.106. Page 533 - (2025-02) - TC_O_32_CS - Made notes about display behaviour more explicit.	136
6.2.107. Page 544 - (2025-02) - TC_O_39_CS - Wait for StatusNotificationRequest or NotifyEventRequest.	137
6.2.108. Page 555 - (2024-11) - Remove StatusNotificationRequest from Authorized reusable state Main B steps.....	139
6.2.109. Page 560 - (2025-04) - Reusable states StopAuthorized & Deauthorized.....	139
6.2.110. Deauthorized	141
6.2.111. Page 566 - (2025-02) - Reusable state RenegotiateChargingLimits	141
6.2.112. Page 573/151 - (2025-04) - Removed Main steps B from IdTokenCached reusable state and added IdTokenCached15118	142
6.2.113. Page 574 - (2025-04) - IdTokenLocalAuthList memory state - set Enable to true if implemented	144
6.2.114. Page 575 - (2024-09) - Reusable state RenewChargingStationCertificate expects a reconnection [784].....	144
6.2.115. Page 575 - (2025-02) - Reusable state RenewChargingStationCertificate must not do a Reset [5281]	145
6.3. CSMS	147
6.3.1. Page 593 - (2025-04) - TC_A_11_CSMS - Added post scenario validation for clarification.....	147
6.3.2. Page 593 - (2024-09) - TC_A_11_CSMS - Reconnect using new client certificate	147
6.3.3. Page 596 - (2024-09) - TC_A_14_CSMS - Update Charging Station Certificate by request of CSMS - Invalid certificate	147
6.3.4. Page 597 - (2025-04) - TC_A_19_CSMS - Added main steps and clarified tool validations	148
6.3.5. Page 597 - (2024-09) - TC_A_19_CSMS - Added additional information regarding the use of the client certificates .	149
6.3.6. Page 597 - (2024-09) - TC_A_19_CSMS - Removed validation of OcppCsmsUrl [O20-4355]	150
6.3.7. Page 637 - (2024-11) - TC_C_50_CSMS - Changed reference to configured valid idToken to a specific eMAID	

idToken.....	150
6.3.8. Page 639 - (2024-09) - TC_C_52_CSMS - TC does not use <Configured contract_certificate>	151
6.3.9. Page 639 - (2025-02) - TC_C_52_CSMS - Certificate needs at least one subCA	151
6.3.10. Page 640 - (2025-04) - TC_D_01_CSMS - Missing tool validation that the idTokenInfo must be provided for all list entries.....	152
6.3.11. Page 712 - (2024-09) - TC_I_01_CSMS - Show EV Driver running total cost.....	152
6.3.12. Page 715 - (2025-02) - TC_I_02_CSMS - Added explicit information about CSMS tariff configuration and sending in needed metervalues	153
6.3.13. Page 726-760 - (2025-04) - TC_K_XX_CSMS - Improved tool validations to be sure valid Charging Profiles are used	154
6.3.14. Page 728 - (2024-09) - TC_K_03_CSMS - Not requiring validFrom/To fields in charging profile [O20-4592] and chargingProfileKind must be Absolute [O20-4591].....	161
6.3.15. Page 733 - (2024-09) - TC_K_10_CSMS - Not requiring validFrom/To fields in charging profile [O20-4592]	161
6.3.16. Page 734 - (2024-09) - TC_K_15_CSMS - Not requiring validFrom/To fields in charging profile [O20-4592]	162
6.3.17. Page 753 - (2024-09) - TC_K_55_CSMS, TC_K_57_CSMS, TC_K_58_CSMS, TC_K_59_CSMS Removed triggerReason = ChargingRateChanged [776]	163
6.3.18. Page 760 - (2024-11) - TC_K_70_CSMS - Updated tool validation chargingProfiles and added preparation step. . .	163
6.3.19. Page 824 - (2025-04) - TC_N_62_CSMS - Check only single identifier is provided.	163
6.3.20. Page 830 - (2024-11) - TC_N_46_CSMS - Updated tool validation sendLocalListRequest	164
6.3.21. Page 830 - (2025-02) - TC_N_46_CSMS - Aligning configuration variable usage.	165
6.3.22. Page 854 - (2025-02) - TC_O_27_CSMS - Fixing validations to be more specific for test case	166
6.3.23. Page 855 - (2025-02) - TC_O_28_CSMS - Fixing validations to be more specific for test case	166

Disclaimer

Copyright © 2010 – 2025 Open Charge Alliance. All rights reserved.

This document is made available under the *Creative Commons Attribution-NoDerivatives 4.0 International Public License* (<https://creativecommons.org/licenses/by-nd/4.0/legalcode>).

Version History

Version	Date	Description
2025-04	2025-04-30	Includes errata for Part 2, 5 and 6 of OCPP 2.0.1 Edition 3.
2025-02	2025-03-06	Includes errata for Part 2, 5 and 6 of OCPP 2.0.1 Edition 3.
2025-01	2025-01-23	Includes errata for Part 1-4 of OCPP 2.0.1 Edition 3
2024-11	2024-11-14	Includes errata for Part 5 and Part 6 of OCPP 2.0.1 Edition 3
2024-09	2024-09-25	Includes errata for Part 4, Part 5 and Part 6 of OCPP 2.0.1 Edition 3
2024-06	2024-06-27	Includes errata for Part 5 and Part 6.

Scope

This document contains errata on the OCPP 2.0.1 documentation. These errata have to be read as an addition to the release of OCPP 2.0.1 Edition 3.

The errata do not affect any schemas of OCPP messages. Certain errata do contain changes to requirements or even new requirements, but only in cases where a requirement contains an obvious error and would not or could not be implemented literally. New requirements are only added when they were already implicitly there. These changes have been discussed in or were proposed by the Technology Working Group of the Open Charge Alliance.

The appendices of the OCPP specification can be updated without requiring a new OCPP release. This mainly concerns the components and variables of the OCPP device model, which can be extended with new components or variables, as long as they are optional.

Terminology and Conventions

Bold: when needed to clarify differences, bold text might be used.

The errata entries are sorted by page number of the affected section of the specification document. When an errata entry affects multiple parts of the specification, then the various changes are grouped together with subsections referring to the pages affected by those changes.

This is version 2025-04 of the errata. The errata of this version are marked with "(2025-04)" in the section title.

In some cases the issue number by which it was reported, is added in square brackets at the end of the section title, e.g. "[349]". For retrieval of the issue in the issue tracking system prefix the number with "OCPP20M", like "[OCPP20M-349]".

0. Part 0

Currently no new errata for OCPP 2.0.1 Edition 3 part 0.

1. Part 1

1.1. Page 9 - (2025-01) - 3.4. Monitoring

The first sub-bullet is further clarified that it refers to the absolute difference:

	Description
Old	When the dataType of the variable is integer or decimal, this value represents the difference to be reached to trigger the monitor.
New	When the dataType of the variable is integer or decimal, this value represents the absolute difference to be reached to trigger the monitor.

2. Part 2

2.1. Page 5 - (2025-01) - Updated limitations on BasicAuthPassword to increase security

A "passwordString" should not be limited to any set of symbols as this limits its security greatly. There is also no reason for limiting the password size.

Table 1. Primitive Datatypes

Datatype	Description
passwordString	This is a UTF-8 encoded case-sensitive string. that can only contain characters from the following character set: "a-z", "A-Z", "0-9" or any of the following limited set of symbols: * - = ; + ! @ .

	ID	Precondition	Requirement definition
Old text	A00.FR.205		The password SHALL be stored in the BasicAuthPassword configuration variable. It SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by passwordString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64).
New text	A00.FR.205		The password SHALL be stored in the BasicAuthPassword configuration variable. It SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and a maximum as defined by the maxLimit of configuration variable BasicAuthPassword , which must be at least 40 characters and at most 64. The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64).
Old text	A00.FR.304	A00.FR.302	The password SHALL be stored in the BasicAuthPassword Configuration Variable. It SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by passwordString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). (Same as A00.FR.205)
New text	A00.FR.304	A00.FR.302	The password SHALL be stored in the BasicAuthPassword Configuration Variable. It SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and a maximum as defined by the maxLimit of configuration variable BasicAuthPassword , which must be at least 40 characters and at most 64. The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). (Same as A00.FR.205)

BasicAuthPassword

The basic authentication password is used for HTTP Basic Authentication. The configuration value is write-only, so that it cannot be accidentally stored in plaintext by the CSMS when it reads out all configuration values.

Required	no		
Component	componentName	SecurityCtrlr	
Variable	variableName	BasicAuthPassword	
	variableAttributes	mutability	WriteOnly
	variableCharacteristics	dataType	string
		maxLimit	At least 40, at most 64.

Description	The basic authentication password is used for HTTP Basic Authentication. The password SHALL be a randomly chosen passwordString with a sufficiently high entropy, consisting of minimum 16 and a maximum as defined by the <i>maxLimit</i> of <i>BasicAuthPassword</i> , which must be at least 40 characters and at most 64. The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). This configuration variable is write-only, so that it cannot be accidentally stored in plaintext by the CSMS when it reads out all configuration variables. This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.
--------------------	---

2.2. Page 14 - (2025-04) - Section 2.7. ISO 15118 support - Aligned ISO 15118 timeout table with updated version in OCPP 2.1

For reference, the current timing constraints for ISO 15118-2:2014 are:

Table 2. ISO 15118-2 Timing constraints

Timeout	Default
Sequence Timeouts	60 seconds
Sequence Performance Timeouts	40 seconds
PaymentDetailsReq/Res	4.5 seconds
AuthorizationReq/Res	1.5 seconds
CertificateUpdateReq/Res	4.5 seconds
CertificateInstallationReq/Res	4.5 seconds

2.3. Page 15 - (2025-01) - Improved text FR.04

The requirement did not take into account the pending status and should refer to the applicable use cases B02 and B03.

	ID	Precondition	Requirement definition	Note
Old text	FR.04	When the CSMS did not accept the BootNotificationRequest from the Charging Station AND The Charging Station sends a message other than BootNotificationRequest	The CSMS SHALL respond with a RPC Framework: CALLERROR: SecurityError.	
New text	FR.04	When the CSMS rejected the BootNotificationRequest from the Charging Station AND The Charging Station sends a message other than BootNotificationRequest	The CSMS SHALL respond with a RPC Framework: CALLERROR: SecurityError.	See use cases B0 2 and B03 for details.

2.4. Page 19 - (2025-01) - Removed requirement A00.FR.003 as the precondition never occurs

Requirement A00.FR.003 describes a precondition that never occurs in which the CSMS connects to the Charging Station. The requirement is therefore removed.

ID	Precondition	Requirement definition
A00.FR.003	If the CSMS tries to connect with a different profile than the Charging Station is using	The Charging Station SHALL terminate the connection.

2.5. Page 19/41 - (2025-01) - Allow downgrading security profile from 3 to 2

For migration purposes a Charging Station implementer has now the option for their Charging Stations to allow downgrading the security profile from 3 to 2, because not all CSMSs support security profile 3. ENCS was consulted to identify if this would pose any security risks. This was not the case as security profile 2 still uses TLS, which provides sufficient security.

Table 3. A05 - Upgrade Charging Station Security Profile

No.	Type	Description
1	Name	Upgrade Charging Station Security Profile
2	ID	A05
	Functional block	A. Security
3	Objective(s)	The CSO wants to increase change the security of the OCPP connection between CSMS and a Charging Station.
4	Description	Use case when migrating from OCPP 1.6 without security profiles to OCPP 2.0.1. Before migrating to a security profile, the prerequisites, like installed certificates or password, need to be configured.
	Actors	CSMS, Charging Station
	Scenario description	<ol style="list-style-type: none"> 1. The CSMS sets a new value for the NetworkConfigurationPriority Configuration Variable via SetVariablesRequest, such that the NetworkConnectionProfile for the new (higher) security profile becomes first in the list and the existing connection profile becomes second in the list. 2. The Charging Station responds with a SetVariablesResponse with status Accepted 3. The CSMS sends a ResetRequest(OnIdle) 4. The Charging Station reboots and connects via the new primary NetworkConnectionProfile
5	Prerequisite(s)	The CSO ensures that a NetworkConnectionProfile has been set using (higher) an allowed security profile AND that the prerequisite(s) for going to a higher the new security profile are met before sending the command to change to a higher the new security profile.
6	Postcondition(s)	The Charging Station was successfully upgraded to a higher new security profile.
7	Error handling	n/a
8	Remark(s)	<p>For security reasons it is by default not allowed to revert to a lower Security Profile using OCPP.</p> <p>Only when the variable AllowSecurityProfileDowngrade is implemented and is set to true, it is allowed to downgrade from profile 3 to profile 2. Even in that case, it is not allowed to revert from profile 2 or profile 3 to security profile 1 using OCPP.</p>

Changed A00 requirement:

	ID	Precondition	Requirement definition
Old text	A00.FR.005		Lowering the security profile that is used, to a less secure profile, is for security reasons, not part of the OCPP specification, and MUST be done through another method, not via OCPP. OCPP messages SHALL NOT be used for this (e.g. SetVariablesRequest or DataTransferRequest).
New text	A00.FR.005		<p>Lowering the security profile that is used, to a less secure profile, is for security reasons, not recommended.</p> <p>The Charging Station SHALL only allow to lower the security profile if the variable AllowSecurityProfileDowngrade is implemented and set to true. In that case, the Charging Station SHALL only allow to downgrade from profile 3 to profile 2. The Charging Station SHALL NOT allow to downgrade from profile 2 or profile 3 to profile 1 using the OCPP protocol.</p>

Added A05 requirements:

	ID	Precondition	Requirement definition
New requirement	A05.FR.08	The variable <code>AllowSecurityProfileDowngrade</code> is implemented and set to true AND The currently active 'SecurityProfile' is 3 AND The Charging Station receives <code>SetVariablesRequest</code> for <code>NetworkConfigurationPriority</code> containing profile slots for <code>NetworkConnectionProfiles</code> with a 'securityProfile' value equal to 2.	The Charging Station SHALL respond with <code>SetVariablesResponse</code> (Accepted)
New requirement	A05.FR.09	The variable <code>AllowSecurityProfileDowngrade</code> is implemented and set to true AND The currently active 'SecurityProfile' is higher than 1 AND The Charging Station receives <code>SetVariablesRequest</code> for <code>NetworkConfigurationPriority</code> containing profile slots for <code>NetworkConnectionProfiles</code> with a 'securityProfile' value equal to 1.	The Charging Station SHALL respond with <code>SetVariablesResponse</code> (Rejected)
New requirement	A05.FR.10	The variable <code>AllowSecurityProfileDowngrade</code> is not implemented or implemented and set to false AND The Charging Station receives <code>SetVariablesRequest</code> for <code>NetworkConfigurationPriority</code> containing profile slots for <code>NetworkConnectionProfiles</code> with a 'securityProfile' value lower than the currently active security profile	The Charging Station SHALL respond with <code>SetVariablesResponse</code> (Rejected)

Changed/added B09 requirements:

	ID	Precondition	Requirement definition
Old text	B09.FR.04	On receipt of the <code>SetNetworkProfileRequest</code> AND the <code>NetworkConnectionProfile</code> contains a lower securityProfile than stored at the configuration variable <code>SecurityProfile</code>	The Charging Station SHALL respond by sending a <code>SetNetworkProfileResponse</code> message, with status Rejected
New text	B09.FR.04	The variable <code>AllowSecurityProfileDowngrade</code> is not implemented or implemented and set to false AND The Charging Station receives a <code>SetNetworkProfileRequest</code> AND the <code>NetworkConnectionProfile</code> contains a lower securityProfile than the currently active security profile	The Charging Station SHALL respond by sending a <code>SetNetworkProfileResponse</code> message, with status Rejected

	ID	Precondition	Requirement definition
New requirement	B09.FR.31	The variable <code>AllowSecurityProfileDowngrade</code> is implemented and set to true AND The currently active 'SecurityProfile' is 3 AND The Charging Station receives a <code>SetNetworkProfileRequest</code> AND the <code>NetworkConnectionProfile</code> contains a securityProfile with a value of 2.	The Charging Station SHALL respond with <code>SetVariablesResponse</code> (Accepted)
New requirement	B09.FR.32	The variable <code>AllowSecurityProfileDowngrade</code> is implemented and set to true AND The currently active 'SecurityProfile' is higher than 1 AND The Charging Station receives a <code>SetNetworkProfileRequest</code> AND the <code>NetworkConnectionProfile</code> contains a securityProfile with a value of 1.	The Charging Station SHALL respond with <code>SetVariablesResponse</code> (Rejected)

Added referenced variable:

AllowSecurityProfileDowngrade

Required	no		
Component	componentName	SecurityCtrlr	
Variable	variableName	AllowSecurityProfileDowngrade	
	variableAttributes	mutability	ReadWrite/ReadOnly
	variableCharacteristics	dataType	boolean
Description	If this variable is implemented and set to true, then the Charging Station allows downgrading the security profile from 3 to 2. For security reasons it is not allowed to revert from profile 2 or profile 3 to security profile 1 using OCPP.		

2.6. Page 26 - (2025-01) - 1.3.7. TLS with Client Side Certificates Profile - Requirements

A new requirement is added to support A00.FR.429:

New requirement

Table 4. Security Profile 3 - TLS with Client Side Certificates - Requirements

ID	Precondition	Requirement definition
A00.FR.430	If the Charging Station certificate has expired	The Charging Station SHOULD still attempt to establish a connection with the CSMS and leave the decision to accept the connection up to the CSMS.

2.7. Page 28 - (2025-01) - Certificate Hierarchy

A note has been added to the specification to warn OCPP implementers about the potential risks involved with the installation of CA bundles.

NOTE

It is not recommended to have preinstalled well-known root CA certificates on a Charging Station like in operating systems or browsers, like for example a CA bundle. Only root and intermediate certificates part of the Charging Station Operator hierarchy should be used for the OCPP connection, as described by section [Certificate Hierarchy](#). Trusting many additional well-known root CA certificates creates security risks.

2.8. Page 31 - (2025-01) - A02/A03 - Updated error handling

NOT RECOMMENDED in the error handling is confusing as there is already a requirement that explicitly states that the Charging Station SHALL retry after "CertSigningWaitMinimum".

7	Error handling	The CSMS accepts the CSR request from the Charging Station, before forwarding it to the CA. But when the CA cannot be reached, or rejects the CSR, the Charging Station will never be known. The CSMS may do some checks on the CSR, but cannot do all the checks that a CA does, and it does not prevent connection timeout to the CA. When something like this goes wrong, either the CA is offline or the CSR send by the Charging Station is not correct, according to the CA. In both cases this is something an operator at the CSO needs to be notified of. The operator then needs to investigate the issue. When resolved, the operator can re-run A02. It is NOT RECOMMENDED to let the Charging Station retry when the certificate is not send within X minutes or hours. When the CSR is incorrect, that will not be resolved automatically. It is possible that only a new firmware will fix this.
---	----------------	--

2.9. Page 34 - (2025-01) - Disallow client certificates future validity date

Currently the specification is unclear on whether accepting client certificates with a future validity date is intended behavior. If it would be allowed behavior, several problems were identified. For example, there would be no way for the CSMS to manage these (ghost) client certificates using OCPP. ENCS was consulted if there would be any security risks when not allowing accepting client certificates with a future validity date. ENCS identified no security risks, therefore it was decided to not allow accepting client certificates with a future validity date. In addition clarifications have been made to describe how to handle non-happy flow scenarios in which the Charging Station is unable to successfully connect using the new client certificate in combination with the NetworkConnectionProfile mechanism described at use case B10.

Changed/added A02 requirements:

	ID	Precondition	Requirement definition	Note
Old text	A02.FR.06		The Charging Station SHALL verify the validity of the signed certificate in the CertificateSignedRequest message, checking at least the period when the certificate is valid, the properties in Certificate Properties , and that it is part of the Charging Station Operator certificate hierarchy as described in Certificate Hierarchy .	
New text	A02.FR.06		The Charging Station SHALL verify the validity of the signed certificate in the CertificateSignedRequest message, checking that the current date (at the time of the update) is within the certificate's validity period, the properties in Certificate Properties , and that it is part of the Charging Station Operator certificate hierarchy as described in Certificate Hierarchy .	When providing a newly signed client certificate with a start period that equals the current time, the CSMS should take into account that there might be a slight discrepancy in the time between the Charging Station and CSMS. This could cause the Charging Station to reject the new certificate, because in case a small time difference exists, the validity period might (just) be in the future for the device.
Old text	A02.FR.08		The Charging Station SHALL switch to the new certificate as soon as the current date and time is after the 'Not valid before' field in the certificate (e.g. by closing the websocket and TLS connection and reconnecting with the new certificate).	

	ID	Precondition	Requirement definition	Note
New text	A02.FR.08	If the certificate is valid.	The Charging Station SHALL respond to the CertificateSignedRequest with status Accepted AND the Charging Station SHALL switch to the new certificate by reconnecting the websocket and TLS connection.	
Old text	A02.FR.09	If the Charging Station contains more than one valid certificate of the <i>ChargingStationCertificate</i> type.	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period.	
New text	A02.FR.09		<Requirement removed>	
Old text	A02.FR.10	A02.FR.09 AND When the Charging Station has validated that the new certificate works	The Charging Station MAY discard the old certificate. It is RECOMMENDED to store old certificates for one month, as fallback.	
New text	A02.FR.10	(A02.FR.08 OR A02.FR.28) AND The Charging Station successfully connected to the CSMS using either one of the certificates.	The Charging Station SHALL discard the client certificate that is NOT in use.	This is to prevent having multiple client certificates installed at the Charging Station, which the CSMS is unable to manage.
New requirement	A02.FR.28	A02.FR.08 AND the charging station was not able to successfully connect to any of the configured entries of NetworkConfigurationPriority using the new certificate AND The Charging Station supports either one or both reconnection mechanisms described at requirements; B10.FR.07 and B10.FR.08.	The Charging Station SHALL for the reconnection mechanism described at B10.FR.07 fallback to the old client certificate AND for the reconnection mechanism described at B10.FR.08 alternate between using the old and new client certificate after all NetworkConfigurationPriority entries.	As described by requirement B10.FR.09, the Charging Station SHOULD NOT stop trying to reconnect to the CSMS. This is to prevent the Charging Station from becoming a stranded asset.
New requirement	A02.FR.29	A02.FR.10 AND The Charging Station discarded the new client certificate.	The Charging Station SHOULD send a SecurityEventNotification DiscardedRenewedClientCertificate to the CSMS.	Otherwise the CSMS is not aware that the Charging Station discarded the new client certificate and the CSMS should again trigger a client certificate renewal.

Changed/added A03 requirements:

	ID	Precondition	Requirement definition	Note
Old text	A03.FR.06		The Charging Station SHALL verify the validity of the signed certificate in the CertificateSignedRequest message, checking at least the period when the certificate is valid, the properties in Certificate Properties , and that it is part of the Charging Station Operator certificate hierarchy as described in Certificate Hierarchy . (Same as A02.FR.06)	

	ID	Precondition	Requirement definition	Note
New text	A03.FR.06		The Charging Station SHALL verify the validity of the signed certificate in the CertificateSignedRequest message, checking that the current date (at the time of the update) is within the certificate's validity period, the properties in Certificate Properties , and that it is part of the Charging Station Operator certificate hierarchy as described in Certificate Hierarchy . (Same as A02.FR.06)	When providing a newly signed client certificate with a start period that equals the current time, the CSMS should take into account that there might be a slight discrepancy in the time between the Charging Station and CSMS. This could cause the Charging Station to reject the new certificate, because in case a small time difference exists, the validity period might (just) be in the future for the device.
Old text	A03.FR.08		The Charging Station SHALL switch to the new certificate as soon as the current date and time is after the 'Not valid before' field in the certificate (e.g. by closing the websocket and TLS connection and reconnecting with the new certificate). (Same as A02.FR.08)	
New text	A03.FR.08	If the certificate is valid.	The Charging Station SHALL respond to the CertificateSignedRequest with status Accepted AND the Charging Station SHALL switch to the new certificate by reconnecting the websocket and TLS connection. (Same as A02.FR.08)	
Old text	A03.FR.09	If the Charging Station contains more than one valid certificate of the ChargingStationCertificate type.	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period. (Same as A02.FR.09)	
New text	A03.FR.09		<Requirement removed>	
Old text	A03.FR.10	A03.FR.09 AND When the Charging Station has validated that the new certificate works	The Charging Station MAY discard the old certificate. It is RECOMMENDED to store old certificates for one month, as fallback. (Same as A02.FR.10)	
New text	A03.FR.10	(A03.FR.08 OR A03.FR.24) AND The Charging Station successfully connected to the CSMS using either one of the certificates.	The Charging Station SHALL discard the client certificate that is NOT in use. (Same as A02.FR.10)	This is to prevent having multiple client certificates installed at the Charging Station, which the CSMS is unable to manage.
New requirement	A03.FR.24	A03.FR.08 AND the charging station was not able to successfully connect to any of the configured entries of NetworkConfigurationPriority using the new certificate AND The Charging Station supports either one or both reconnection mechanisms described at requirements; B10.FR.07 and B10.FR.08.	The Charging Station SHALL for the reconnection mechanism described at B10.FR.07 fallback to the old client certificate AND for the reconnection mechanism described at B10.FR.08 alternate between using the old and new client certificate after all NetworkConfigurationPriority entries. (Same as A02.FR.28)	As described by requirement B10.FR.09, the Charging Station SHOULD NOT stop trying to reconnect to the CSMS. This is to prevent the Charging Station from becoming a stranded asset.

	ID	Precondition	Requirement definition	Note
New requirement	A03.FR.25	A03.FR.10 AND The Charging Station discarded the new client certificate.	The Charging Station SHOULD send a SecurityEventNotification DiscardedRenewedClientCertificate to the CSMS. (Same as A02.FR.29)	Otherwise the CSMS is not aware that the Charging Station discarded the new client certificate and the CSMS may need to trigger a new client certificate renewal.

Changed/added B10 requirements:

	ID	Precondition	Requirement definition	Note
Old text	B10.FR.07	B10.FR.03 AND All NetworkProfileConnectionAttempts for every entry of NetworkConfigurationPriority failed.	The Charging Station SHOULD fallback and start 'reconnecting' to the NetworkConnectionProfile for which the last successful connection was made.	'reconnecting' in this requirement, refers to the reconnection mechanism described at section 5.3. Reconnecting from "Part 4 - JSON over WebSockets implementation guide".
New text	B10.FR.07	B10.FR.09	The Charging Station SHOULD fallback and start 'reconnecting' to the NetworkConnectionProfile for which the last successful connection was made.	'reconnecting' in this requirement, refers to the reconnection mechanism described at section 5.3. Reconnecting from "Part 4 - JSON over WebSockets implementation guide".
New requirement	B10.FR.08	B10.FR.09	The Charging Station SHOULD restart connecting with all configured entries of the NetworkConfigurationPriority	
New requirement	B10.FR.09	B10.FR.03 AND All NetworkProfileConnectionAttempts for every entry of NetworkConfigurationPriority failed.	The Charging Station SHOULD NOT stop trying to reconnect to the CSMS. The Charging Station SHOULD implement either one or both reconnecting mechanisms described at requirements; B10.FR.07 and B10.FR.08.	This is to prevent the Charging Station from becoming a stranded asset.

2.10. Page 58/60 - (2025-01) - Missing requirement information about omitting the value for WriteOnly variables

The specification does describe already that the value for WriteOnly variables need to be omitted, however this information is not part of a requirement yet.

Changed B07 requirement:

	ID	Precondition	Requirement definition	Note
Old text	B07.FR.03	B07.FR.01	The Charging Station SHALL send the requested information via one or more NotifyReportRequest messages to the CSMS.	It is good practice to send the report data in as few messages as possible in order to limit data overhead.
New text	B07.FR.03	B07.FR.01	The Charging Station SHALL send the requested information, excluding the value of WriteOnly variables, via one or more NotifyReportRequest messages to the CSMS.	It is good practice to send the report data in as few messages as possible in order to limit data overhead.

Changed B08 requirement:

	ID	Precondition	Requirement definition
Old text	B08.FR.03	B08.FR.01	The Charging Station SHALL send the requested information via one or more NotifyReportRequest messages to the CSMS.
New text	B08.FR.03	B08.FR.01	The Charging Station SHALL send the requested information, excluding the value of WriteOnly variables , via one or more NotifyReportRequest messages to the CSMS.

2.11. Page 69 - (2025-01) - B11 - Reset without Ongoing Transaction - Requirements

The definition of B11.FR.06 has been improved:

	ID	Precondition	Requirement definition
Old text	B11.FR.06	B11.FR.01 AND For example there is a firmware update ongoing that cannot be interrupted.	The Charging Station SHALL respond with a status <i>Rejected</i> .
New text	B11.FR.06	B11.FR.01 AND Charging Station is at this moment not able to perform a reset	The Charging Station SHALL respond with a status <i>Rejected</i> .

2.12. Page 73 - (2025-01) - B12 - Reset with Ongoing Transaction - Requirements

The requirement B12.FR.04, 08 and added B12.FR.10 have been improved:

At the (2025-04) version, the old text requirement definitions have been corrected, because they already contained the revision instead of the original text from the edition 3 specification. In addition, requirement B12.FR.10 was newly added to edition 3, not updated.

	ID	Precondition	Requirement definition
Old text (Corrected in 2025-04)	B12.FR.04	If no <i>evselId</i> is supplied AND If any transaction is in progress and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate any transaction in progress and send a TransactionEventRequest (eventType = Ended) message before performing a reboot.
New text	B12.FR.04	If no <i>evselId</i> is supplied AND If any transaction is in progress and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate any transaction in progress and send a TransactionEventRequest (eventType = Ended) message with triggerReason = ResetCommand and transactionInfo.stoppedReason = ImmediateReset for each terminated transaction before performing a reboot.
Old text (Corrected in 2025-04)	B12.FR.08	If an <i>evselId</i> is supplied AND If a transaction is in progress on the EVSE and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate the transaction in progress on the EVSE and send a TransactionEventRequest (eventType = Ended) message before performing a reset.
New text	B12.FR.08	If an <i>evselId</i> is supplied AND If a transaction is in progress on the EVSE and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate the transaction in progress on the EVSE and send a TransactionEventRequest (eventType = Ended) message with triggerReason = ResetCommand and transactionInfo.stoppedReason = ImmediateReset before resetting the EVSE.
Added (Corrected in 2025-04)	B12.FR.10	B12.FR.02 AND Charging Station is at this moment not able to perform an Immediate reset for a reason other than the fact that a transaction is in progress	The Charging Station SHALL return a ResetResponse Rejected

2.13. Page 77 - (2025-01) - 1.6 Relationship between authorization and transaction

A new section has been added after 1.5 Unknown Offline Authorization

This section is informative.

The purpose of authorization is twofold. It ensures in the first place, that energy is only offered to a known user (represented by the *idToken*), which is essential for billing. In the second place, it ensures that only the user who was authorized in the first place (or a member of the same group of users) is allowed to unplug the cable. This is an important safeguard against cable theft in situations where the charging station does not have a fixed cable and the user brings its own charging cable.

Authorization and the duration of the authorization period are not strictly tied to a transaction: it is possible to have transactions without explicit authorization, e.g. in the case of a charging station that can be started with a push button. In that case one could say that there is a permanent authorization for anyone to charge.

The start of the authorization period:

- can take place before a transaction is started (e.g. when a cable is not yet connected), or
- can cause a transaction to be started (e.g. when authorization is defined as the start of a transaction by setting TxStartPoint = Authorized), or
- can happen after a transaction has already started (e.g. when connection of the cable is defined as the start of a transaction by setting TxStartPoint = EVConnected).

(See chapter E.1.1 "Flexible transaction start/stop" for a description of transaction start and stop points.)

In any case, authorization (or authorization period) ends when the same *idToken* is presented again for authorization, or when the transaction ends. This means that ending of the authorization period:

- can happen during a transaction without ending the transaction (e.g. when *idToken* is presented again, but TxStopPoint = EVConnected), or
- can cause the transaction to end (e.g. when *idToken* is presented again and TxStopPoint = Authorized), or
- can be caused by the end of the transaction (e.g. when *idToken* is not presented for authorization, but the cable is disconnected and TxStopPoint = EVConnected), or
- can be caused by cable plug-out if no transaction was started.

A Charging Station defines when authorization starts (i.e. upon receiving the AuthorizeResponse, or when authorizing locally via authorization cache or local authorization list) and when authorization ends (i.e. when *idToken* is presented a second time, or when the transaction ends). Charging Station notifies CSMS about this, as follows:

- If authorization occurs before start of the transaction, Charging Station tells CSMS that authorization has taken place, by including the *idToken* in the first TransactionEventRequest of the transaction.
- If authorization occurs within a transaction or at the start of a transaction, Charging Station reports this by including the *idToken* in TransactionEventRequest together with a *triggerReason* = Authorized.
- The end of authorization is reported in a TransactionEventRequest with a *triggerReason* = StopAuthorized or Deauthorized, or by reporting the end of the transaction.

2.14. Page 80 - (2025-01) - C01 - EV Driver Authorization using RFID - Requirements

A requirement has been added to define when two *idTokens* are considered equal:

New requirement

ID	Precondition	Requirement definition	Note
C01.FR.25		Two <code>[idtoken.type]</code> elements are considered to be equal when they have the same value for the fields <code>idToken.idToken</code> and <code>idToken.type</code>	<code>additionalInfo</code> is not taken into account when comparing. See C01.FR.02, C01.FR.03, C01.FR.05 for <code>idToken</code> requirements where <code>idTokens</code> are compared.

2.15. Page 80 - (2025-01) - C01 - EV Driver Authorization using RFID - Requirements

A requirement has been added to make explicit that authorization ends after `EVConnectionTimeout`:

New requirement

ID	Precondition	Requirement definition	Note
C01.FR.26	When an <code>idToken</code> has been authorized and the EV Driver does not plug in the charging cable before the timeout set by the Configuration Variable: <code>EVConnectionTimeOut</code>	Charging Station SHALL end the authorization of <code>idToken</code>	See also E03.FR.05 and F02.FR.07/08 for additional behavior in case a transaction had already been started.

2.16. Page 99 - (2025-01) - C09- Authorization by GroupId - Requirements

A `GroupIdToken` is controlled by CSMS. This should be reflected in the type.

New requirement

ID	Precondition	Requirement definition
C09.FR.13		The field <code>idToken.type</code> of a <code>GroupIdToken</code> SHOULD be <code>Central</code>

2.17. Page 101 - (2025-01) - Updated requirements related to clarify the relation between `AuthCacheLifeTime` and `cacheExpiryDateTime`

The OCPP specification already describes the relation between `AuthCacheLifeTime` and `cacheExpiryDateTime` at section 1.3 Authorization Cache, however the requirements are missing this information.

Changed C10 requirements:

	ID	Precondition	Requirement definition	Note
Old text	C10.FR.10	When the validity of an Authorization Cache entry expires.	The Authorization Cache entry SHALL be removed from the cache or changed to <code>Expired</code> .	
New text	C10.FR.10	<code>NOT C10.FR.13 AND when more than AuthCacheLifeTime seconds have passed since idTokenInfo was last updated</code>	The Authorization Cache entry SHALL be removed from the cache or changed to <code>Expired</code> .	A <code>cacheExpiryDateTime</code> in the past will prevent an <code>idToken</code> from being stored in the authorization cache, or remove it from authorization cache if it was already present. This is used e.g. for prepaid accounts that should not be kept in authorization cache.

	ID	Precondition	Requirement definition	Note
Old text	C10.FR.13	When IdTokenInfoType contains a value for <i>cacheExpiryDateTime</i>	The time a token is considered to be present in the cache is determined by <i>cacheExpiryDateTime</i> . This variable indicates the date and time after which a token expires in the Authorization Cache.	This expiry of the cache is not the same as the expiration date that is set for the IdToken (e.g. RFID card expiry date).
New text	C10.FR.13	When IdTokenInfoType contains a value for <i>cacheExpiryDateTime</i> and current time is greater than <i>idTokenInfo.cacheExpiryDateTime</i>	The Authorization Cache entry SHALL be removed from the cache or changed to <i>Expired</i> .	This expiry of the cache is not the same as the expiration date that is set for the IdToken (e.g. RFID card expiry date).

2.18. Page 113 - (2025-01) - C16 - Stop Transaction with a Master Pass - Requirements

Requirements about actual stopping of transactions have been added/updated:

Updated requirement

Table 5. C16 - Stop Transaction with a Master Pass - Requirements

	ID	Precondition	Requirement definition
Old text	C16.FR.01	User presents an IdToken that has a groupId equal to <i>MasterPassGroupId</i> AND The Charging Station has a UI with input capabilities.	The Charging Station SHALL "show" the Master Pass UI to let user select which transaction to stop.
New text	C16.FR.01	User presents an IdToken that has a groupId equal to <i>MasterPassGroupId</i> AND The Charging Station has a UI with input capabilities.	The Charging Station SHALL "show" the Master Pass UI to let user select which transaction to stop.
Old text	C16.FR.02	User presents an IdToken that has a groupId equal to <i>MasterPassGroupId</i> AND the Charging Station does NOT have a UI.	The Charging Station SHALL stop all ongoing transactions as described in use case E07.
New text	C16.FR.02	User presents an IdToken that has a groupId equal to <i>MasterPassGroupId</i> AND the Charging Station does NOT have a UI.	The Charging Station SHALL stop all ongoing transactions as described in use case E07.

New requirements

ID	Precondition	Requirement definition
C16.FR.07	C16.FR.01 OR C16.FR.02	Charging Station SHALL stop the transaction as described in use case E07.
C16.FR.07	C16.FR.07	Charging Station SHALL set <i>transactionInfo.stoppedReason</i> = <i>MasterPass</i> in <i>TransactionEventRequest</i> with <i>eventType</i> = <i>Ended</i> .

2.19. Page 129 - (2025-01) - Updated sequence diagram E01 S5

After all added clarifications on the PowerPathClosed TxStartPoint the corresponding sequence diagram was not updated to reflect this.

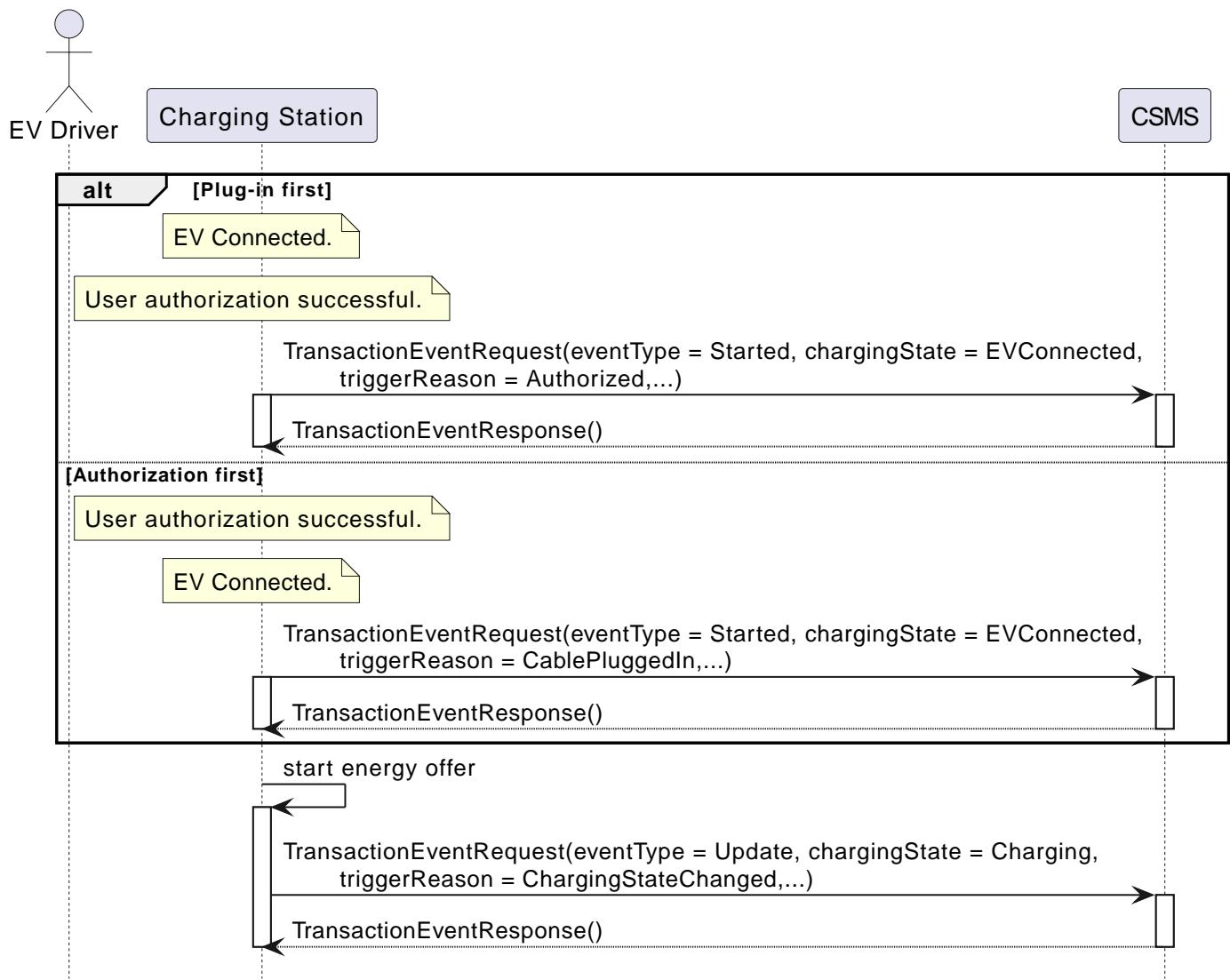


Figure 43. Sequence Diagram: Start Transaction options - PowerPathClosed

2.20. Page 151 - (2025-02) - E06.FR.05 for DataSigned as TxStopPoint is invalid

DataSigned cannot be used as a TxStopPoint. This requirement is therefore invalid and confusing when present.

Deleted requirement

	ID	Precondition	Requirement definition
Delete	E06.FR.05	<code>TxStopPoint</code> contains: <code>DataSigned</code> AND Charging Station can no longer retrieve signed meter values.	The Charging Station SHALL stop the transaction and send a <code>TransactionEventRequest (eventType = Ended)</code> to the CSMS.

2.21. Page 152 - (2025-01) - E07 - Improved scenario description names

The use case did not clearly indicate when the scenario and the alternative scenario are applicable.

No.	Type	Description
1	Name	Transaction locally stopped by IdToken
2	ID	E07
3	Objective(s)	The EV Driver wants to stop an ongoing transaction, by locally presenting his IdToken.

No.	Type	Description
4	Description	This use case covers how the EV Driver can stop a transaction when he wants to leave the charging station.
	Actors	Charging Station, CSMS, EV Driver
	Scenario description Reporting StopAuthorized with end of transaction	<p><i>TxStopPoint = Authorized (or PowerPathClosed)</i></p> <ol style="list-style-type: none"> 1. The EV Driver presents IdToken a second time to end charging. 2. The Charging Station stops the energy transfer and if the cable is not permanently attached, the Charging Station unlocks the cable. 3. The Charging Station sends a <i>TransactionEventRequest (eventType = Ended)</i> with <i>triggerReason = StopAuthorized</i> and <i>stoppedReason = Local</i>. 4. The CSMS responds with a <i>TransactionEventResponse</i>.
	Alternative scenario(s) Reporting StopAuthorized in Update event first, then end transaction	<p><i>TxStopPoint = Authorized (or PowerPathClosed)</i></p> <ol style="list-style-type: none"> 1. The EV Driver presents IdToken a second time to end charging. 2. The Charging Station sends a <i>TransactionEventRequest (eventType = Updated)</i> with <i>triggerReason = StopAuthorized</i> 3. The CSMS responds with a <i>TransactionEventResponse</i>. 4. The Charging Station stops the energy transfer and if the cable is not permanently attached, the Charging Station unlocks the cable. 5. The Charging Station sends a <i>TransactionEventRequest (eventType = Ended)</i> with <i>triggerReason = ChargingStateChanged, transactionInfo.chargingState = EVConnected</i> 6. The CSMS responds with a <i>TransactionEventResponse</i>.
...		

2.22. Page 154 - (2025-01) - E07 - Transaction locally stopped by IdToken

Sequence diagram Fig. 55 is updated to fix an error.

The fourth TransactionEventRequest needs to have: *triggerReason = ChargingStateChanged, chargingState = EVConnected*

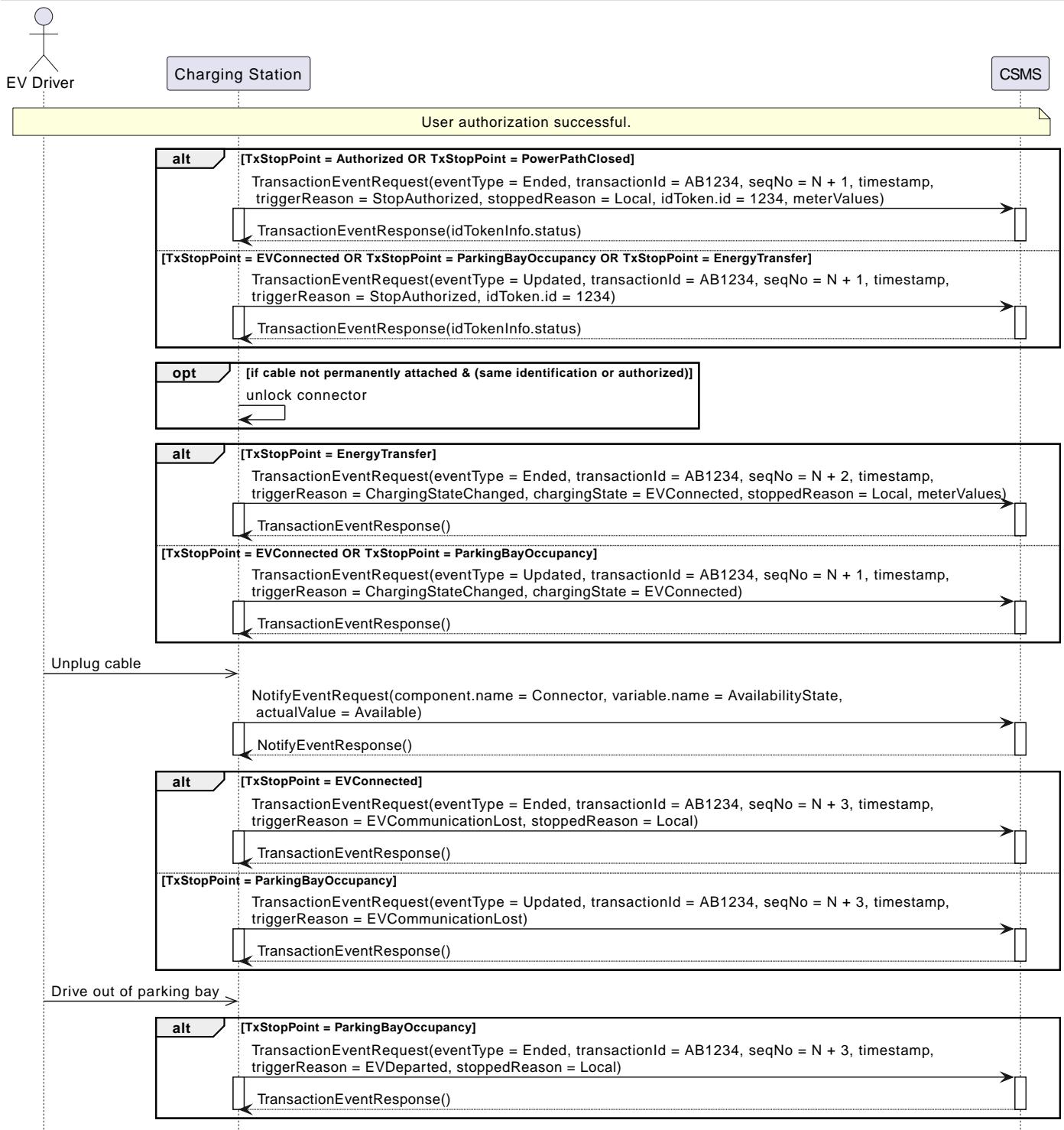


Figure 55. Sequence Diagram: Transaction locally stopped by IdToken with TransactionEventRequest reported strictly by TxStopPoint configuration

2.23. Page 198 - (2025-01) - G01 - Status Notification - Requirements

A requirement is added to make explicit that a plug-in on reserved connector does not automatically change status to Occupied.

ID	Precondition	Requirement definition
G01.FR.09	The connector is Reserved when an EV is connecting AND (No IdToken is presented OR EV driver presents an IdToken not matching the reservation)	Connector status SHALL not change.

2.24. Page 197 - (2025-01) - G01 - Status Notification - State transition overview for connecting/disconnecting

The following text from the table has to be removed:

Initial	Cable plugin	Cable unplug
Available	→ Occupied	-
Occupied	-	→ Available (→ Unavailable, if scheduled to become Unavailable)
Reserved	(→ Occupied, only if authorized for reserved IdToken)	-
Unavailable	-	-
Faulted	-	-

2.25. Page 208 - (2025-01) - H. Reservation - Introduction

The Introduction text has been updated:"

Old text	<p>This Functional Block describes the reservation functionality of OCPP. The reservation functionality enables an EV Driver to make a reservation of a Charging Station/EVSE, ensuring an available Connector at a Charging Station when he arrives.</p> <p>With Charging Stations not being abundantly available, and EVs having limited range, EV Drivers plan their trips from Charging Station to Charging Station. They need to know for sure they can use a Charging Station they plan to go to. They don't like it when another EV Driver has started using the Charging Station in the time they were traveling to the Charging Station.</p> <p>For the EV Driver it is useful to be able to reserve a specific Type of Connector, or, when the EV Driver has no preference, an unspecified EVSE at a Charging Station. So he knows for sure he can charge at the Charging Station when he arrives.</p>
New text	<p>This Functional Block describes the reservation functionality of OCPP. The reservation functionality enables an EV Driver to reserve an EVSE at a Charging Station until a certain time in order to ensure that this EVSE cannot be occupied by another user.</p> <p>OCPP allows to reserve a specific EVSE at a Charging Station or a specific connector type. The EV Driver can also reserve an unspecified EVSE, in which case the Charging Station will make sure that at least one EVSE remains available for the EV Driver.</p> <p>Only available EVSEs can be reserved, since a Charging Station cannot know in advance when an occupied EVSE will become available again. This makes it impossible to guarantee a reservation for an EVSE that is currently occupied.</p> <p>NOTE: A CSMS would still be able to support the reservation functionality for occupied EVSEs by delaying the sending of the reservation message to the Charging Station until the EVSE becomes available, but there is no guarantee that it is available in time.</p>

2.26. Page 213 - (2025-02) - H02 - Added missing requirements

Added missing requirements explicitly specifying behaviour of Charging Station when a reservation is cancelled.

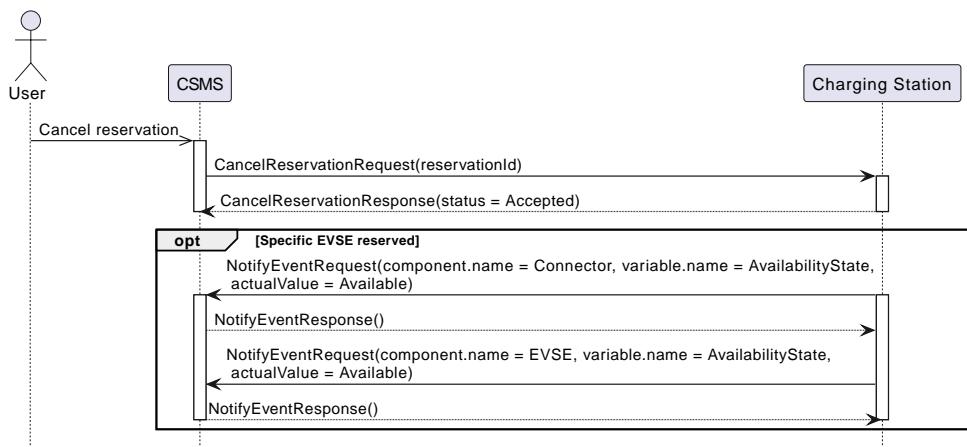
Removed details from scenario description:

No.	Type	Description
		[...]

No.	Type	Description
	Scenario description	<p>1. EV Driver asks the CSMS to cancel a reservation.</p> <p>2. To cancel a reservation, the CSMS sends CancelReservationRequest to the Charging Station.</p> <p>3. If the Charging Station has a reservation matching the reservationId in the request PDU, it returns the status Accepted.</p> <p>4. If a specific EVSE was reserved for this reservation, the Charging Station sends StatusNotificationRequest with the status Available or a NotifyEventRequest with AvailabilityState set to Available for all the Connectors of that EVSE.</p> <p>5. The CSMS responds with StatusNotificationResponse or NotifyEventResponse to the Charging Station.</p> <p>6. The reservation is canceled.</p> <p>[...]</p>

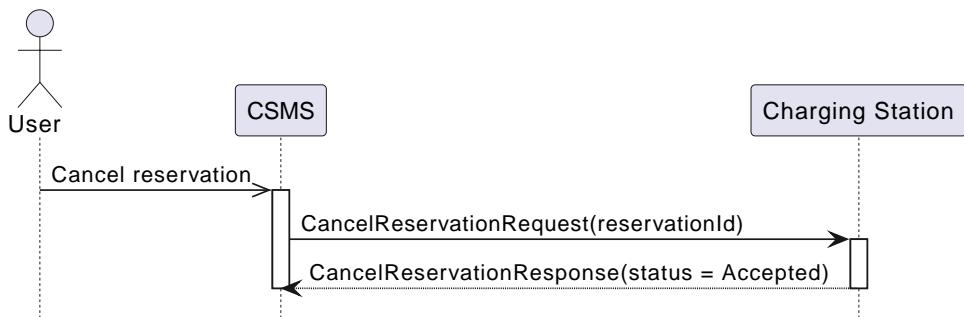
Removed details from sequence diagram

Old:



Sequence Diagram: Cancel Reservation

New:



Sequence Diagram: Cancel Reservation

New requirements

ID	Precondition	Requirement definition
H02.FR.03	H02.FR.02 AND If a specific EVSE was reserved for this reservation	The Charging Station SHALL allow charging again on this EVSE.
H02.FR.04	H02.FR.03	The Charging Station SHALL send a StatusNotificationRequest with status Available or a NotifyEventRequest with AvailabilityState set to Available to the CSMS for each connector, notifying the CSMS that all the connectors of this EVSE are available again for any EV Driver.

ID	Precondition	Requirement definition
H02.FR.05	H02.FR.02 AND If no specific EVSE was reserved for this reservation	The Charging Station SHALL allow charging on all EVSE which were not reserved explicitly.
H02.FR.06	H01.FR.05 AND before cancelling the reservation the amount of EVSEs reserved was equal to the amount of reservations	The Charging Station SHALL send for all connectors of all EVSEs which were not reserved explicitly: - a NotifyEventRequest with component = "Connector", variable = "AvailabilityState", trigger = "Delta", actualValue = "Available", OR - a StatusNotificationRequest with connectorStatus = Available.

2.27. Page 214/215 - (2025-01) - Improved use case scenario descriptions and added S3

The scenario descriptions have been updated to better explain the actual claiming of a reservation and have been clearly divided based on the configured TxStartPoint(s). In addition, A S3 has been added: 'Use an EVSE when Charging Station has a reservation for idToken, but connector status is Available.' This happens when reservation is for an unspecified EVSE and multiple EVSEs are available.

No.	Type	Description
1	Name	Use a reserved EVSE
2	ID	H03
3	Objective(s)	Use a reserved EVSE
4	Description	This use cases covers how a reserved EVSE can be used based on IdToken and GroupIdToken information.
	Actors	Charging Station, CSMS, EV Driver
S1	Scenario objective	Use an EVSE with connector status Reserved, that is reserved for this IdToken
	Scenario description	<p><i>TxStartPoint = "Authorized"; IdToken presented first</i></p> <ol style="list-style-type: none"> 1. The EV Driver presents an IdTokenType at the Charging Station that is the same as the reservation's IdTokenType. 2. Charging Station matches IdTokenType with the reservation. 3. Connector status becomes Available, since reservation has now been consumed. 4. Charging Station optionally authorizes the IdTokenType via an AuthorizeRequest. 5. If authorization accepted, or authorization step was skipped: <ul style="list-style-type: none"> a. Charging Station starts a transaction as in E03 - Start Transaction - IdToken First. b. Connector status will become Occupied when cable is connected.
	Scenario description #2	<p><i>TxStartPoint = "EVConnected"; Cable plugged in first</i></p> <ol style="list-style-type: none"> 1. The EV Driver connects the cable. 2. Charging Station starts a transaction, but EVSE connector status remains Reserved. 3. The EV Driver presents an IdTokenType at the Charging Station that is the same as the reservation's IdTokenType 4. Charging Station matches IdTokenType with the reservation 5. Connector status becomes Occupied, since reservation has now been consumed 6. Charging Station optionally authorizes the IdTokenType via an AuthorizeRequest 7. If authorization accepted, or authorization step was skipped: <ul style="list-style-type: none"> a. Charging Station starts a transaction as in E02 - Start Transaction - Cable Plugin First

No.	Type	Description
5	Prerequisite(s)	EVSE has been reserved for IdToken and connector status is Reserved.
6	Postcondition(s)	n/a

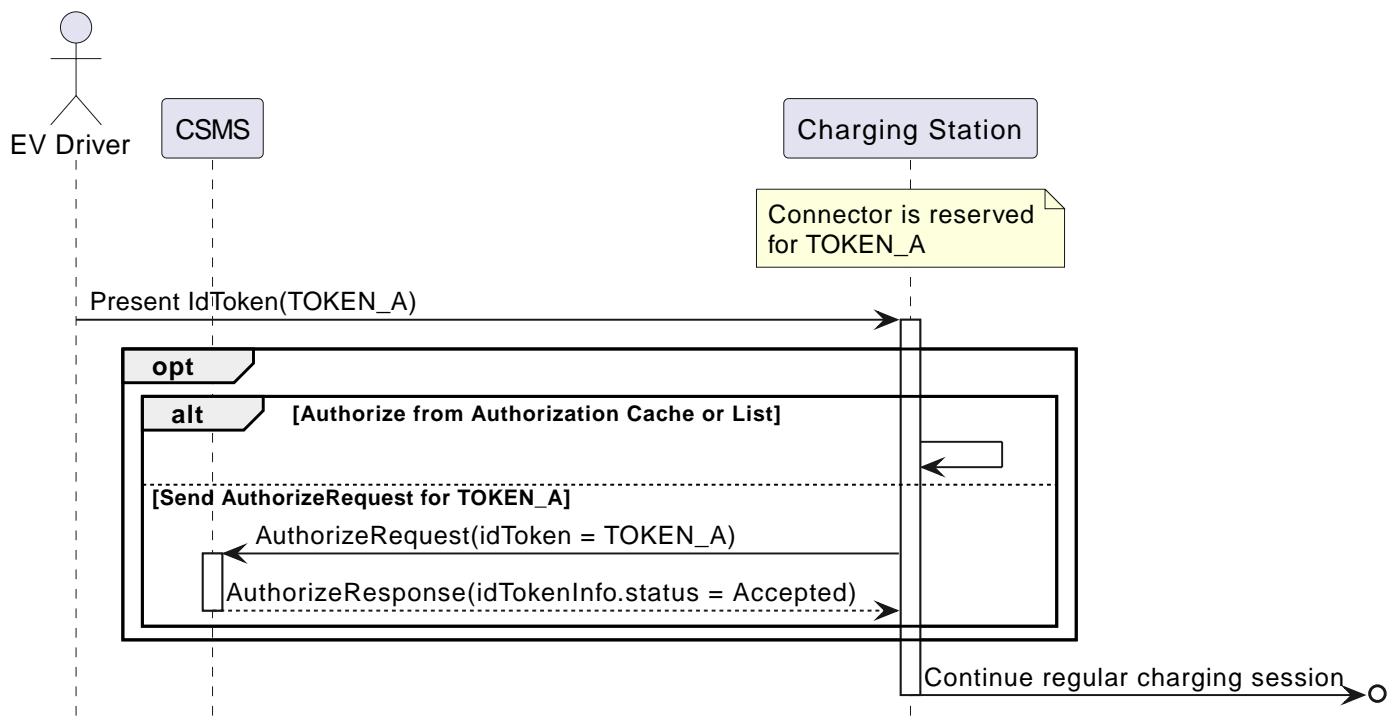


Figure 82. Sequence Diagram: Use a reserved EVSE with IdToken

S2	Scenario objective	Use an EVSE with connector status Reserved, that is reserved for this GroupIdToken
	Scenario description	<p>TxStartPoint = "Authorized"; IdToken presented first</p> <ol style="list-style-type: none"> 1. The EV Driver presents an IdTokenType at the Charging Station that is not the same as the reservation's IdTokenType, but the reservation contains a groupIdToken. 2. Charging Station authorizes the IdTokenType via an AuthorizeRequest, Local Authorization List or Authorization Cache, and checks if the groupIdToken of the IdTokenType matches with the reservation. 3. If groupIdTokens match: <ol style="list-style-type: none"> a. Connector status becomes Available, since reservation has now been consumed. b. Charging Station starts a transaction as in E03 - Start Transaction - IdToken First c. Connector status will become Occupied when cable is connected

	<i>Scenario description #2</i>	<p><i>TxStartPoint = "EVConnected"; Cable plugged in first</i></p> <ol style="list-style-type: none"> 1. The EV Driver connects the cable. 2. Charging Station starts a transaction, but connector status remains Reserved. 3. The EV Driver presents an IdTokenType at the Charging Station that is not the same as the reservation's IdTokenType, but the reservation contains a groupIdToken. 4. Charging Station authorizes the IdTokenType via an AuthorizeRequest, Local Authorization List or Authorization Cache, and checks if the groupIdToken of the IdTokenType matches with the reservation. 5. If groupIdTokens match: <ol style="list-style-type: none"> a. Connector status becomes Occupied, since reservation has now been consumed. b. Charging Station starts a transaction as in E02 - Start Transaction - Cable Plugin First
5	Prerequisite(s)	EVSE has been reserved for GroupIdToken. EVSE connectorStatus = Reserved.
6	Postcondition(s)	n/a

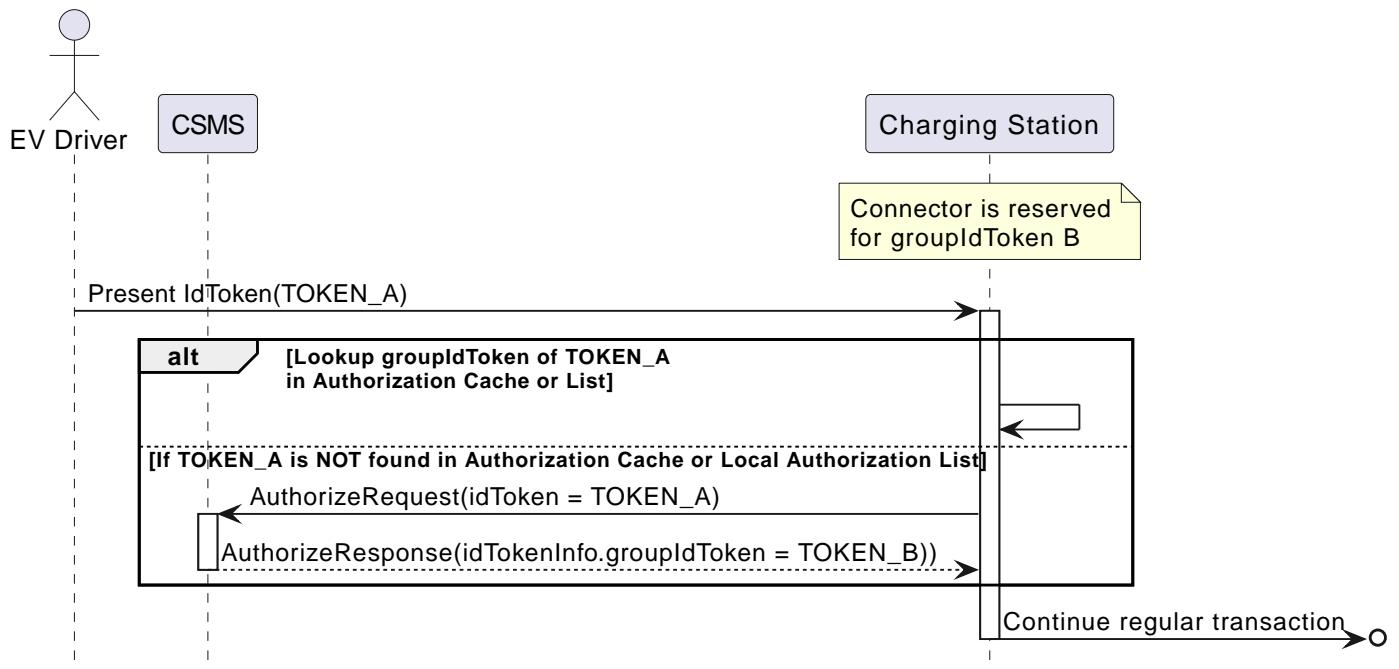


Figure 83. Sequence Diagram: Use a reserved EVSE with GroupId

S3	Scenario objective	Use an EVSE when Charging Station has a reservation for idToken, but connector status is Available. This happens when reservation is for an unspecified EVSE and multiple EVSEs are available.
	Scenario description	<p><i>TxStartPoint = "Authorized"; IdToken presented first</i></p> <p>+ Identical to scenario S1 above.</p>

	Scenario description #2	<p><i>TxStartPoint = "EVConnected"; Cable plugged in first</i></p> <ol style="list-style-type: none"> 1. The EV Driver connects the cable 2. Charging Station reports connector status as Occupied 3. Charging Station starts a transaction 4. The EV Driver presents an IdTokenType at the Charging Station that is the same as the reservation's IdTokenType 5. Charging Station matches IdTokenType with the reservation 6. Charging Station optionally authorizes the IdTokenType via an AuthorizeRequest 7. If authorization accepted, or authorization step was skipped: <ol style="list-style-type: none"> a. Charging Station starts a transaction as in E02 - Start Transaction - Cable Plugin First
5	Prerequisite(s)	Unspecified EVSE has been reserved for idToken. EVSE connector status is Available.
6	Postcondition(s)	n/a

7	Error handling	n/a
8	Remark(s)	<p>It is RECOMMENDED to validate the Identifier with an AuthorizeRequest after reception of ReserveNowRequest and before the start of the transaction.</p> <p>If an <i>idToken</i> is presented that does not match the reservation (and <i>groupIdTokens</i> do not match either), then this <i>idToken</i> is not authorized to charge.</p> <p>If <i>TxStartPoint</i> = Authorized or PowerPathClosed then a transaction would not be started in this case.</p> <p>If <i>TxStartPoint</i> = EVConnected or ParkingBayOccupancy then a transaction would be started by cable plug-in or occupancy of parking bay, but charging would not start. Assuming a <i>TxStopPoint</i> of EVConnected the transaction would be ended at cable plug-out.</p>

Added note:

	ID	Precondition	Requirement definition	Note
Old text	H03.FR.09	When an <i>idToken</i> or <i>groupIdToken</i> is presented that matches a reservation	Charging Station SHALL consider the reservation to be used (consumed)	
New text	H03.FR.09	When an <i>idToken</i> or <i>groupIdToken</i> is presented that matches a reservation	Charging Station SHALL consider the reservation to be used (consumed)	The <i>(group)IdToken</i> can be presented locally at a card reader, but can also be part of a RequestStartTransaction .

2.28. Page 226 - (2025-01) - I06.FR.02 Improved requirement text

Old	I06.FR.02	I06.FR.01 AND When there is updated tariff information available.	The CSMS SHALL respond with a TransactionEventResponse message to the Charging Station, containing the updated tariff information in the PersonalMessage field.
New	I06.FR.02	I06.FR.01 AND When there is updated tariff information available.	The CSMS SHALL respond with a TransactionEventResponse message to the Charging Station, containing the updated tariff information in the updatedPersonalMessage field.

2.29. Page 231 - (2025-01) - Updated section Multiple Locations/Phases

The section now specifies the 'relevant' configuration variables that apply.

Old	When a Charging Station can measure the same measurand on multiple locations or phases, all possible locations and/or phases SHALL be reported when configured in one of the relevant Configuration Variables.
New	When a Charging Station has measurands configured in SampledDataTxStarted/Updated/EndedMeasurands and/or AlignedDataMeasurands /AlignedDataTxEndedMeasurands , that can be measured on multiple locations or phases, then all possible locations and/or phases SHALL be reported.

2.30. Page 243 - (2025-01) - Improved section on external Smart Charging Control Signals

To be inline with the EMS scenario additions to OCPP 2.1, this section also has been improved for OCPP 2.0.1.

Old text:

The OCPP protocol is originally developed for communication between a CSMS and one or more Charging Stations. As described in the above, this means that a Charging Station Operator (CSO) CSMS controls a Charging Station and, based on the charging limits of both the EV and the Charging Station, the CSO determines how fast the EV is charged. However, in some situations / applications of OCPP enabled Charging Stations, these are not the only 2 factors that determine the charging speed. Other inputs that determine charging speed could be DSO signals (e.g. via IEC 61850 [IEC61850-7-420], IEC 60870 [IEC60870-5-104], DNP3 [DNP3] or OpenADR [OPENADR]) or signals from a Building / Home Energy Management System. Although these signals are out of scope for OCPP, it seems clear from an OCPP perspective that the CSMS is to be informed of changes in charging by external signals. However, this also leads to a number of questions, such as how to deal with conflicting signals. The figure below presents an example setup with an Energy Management System, where the external signals are visualized both in a setup with direct communication to the Charging Station as well as a multiple Charging Station setup using a Local Controller:

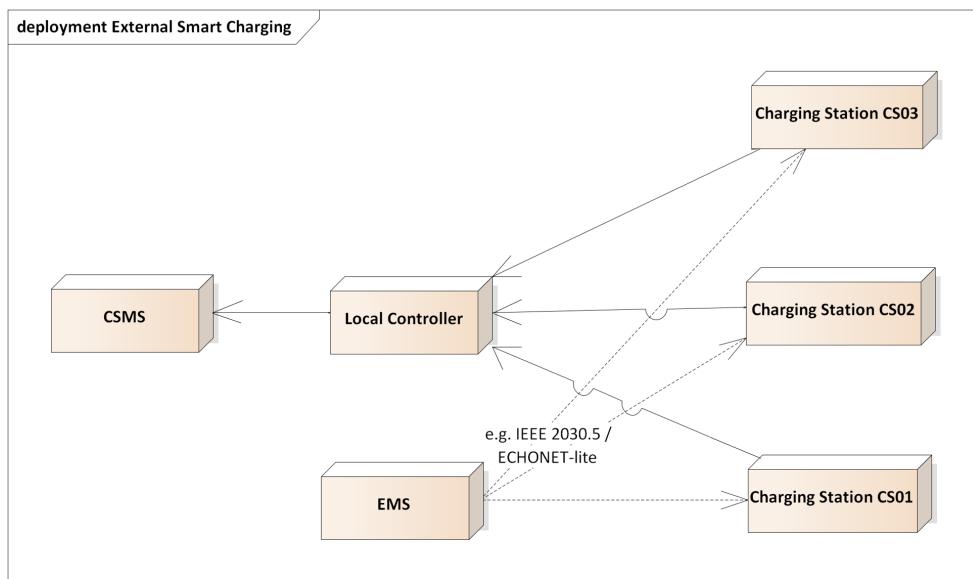


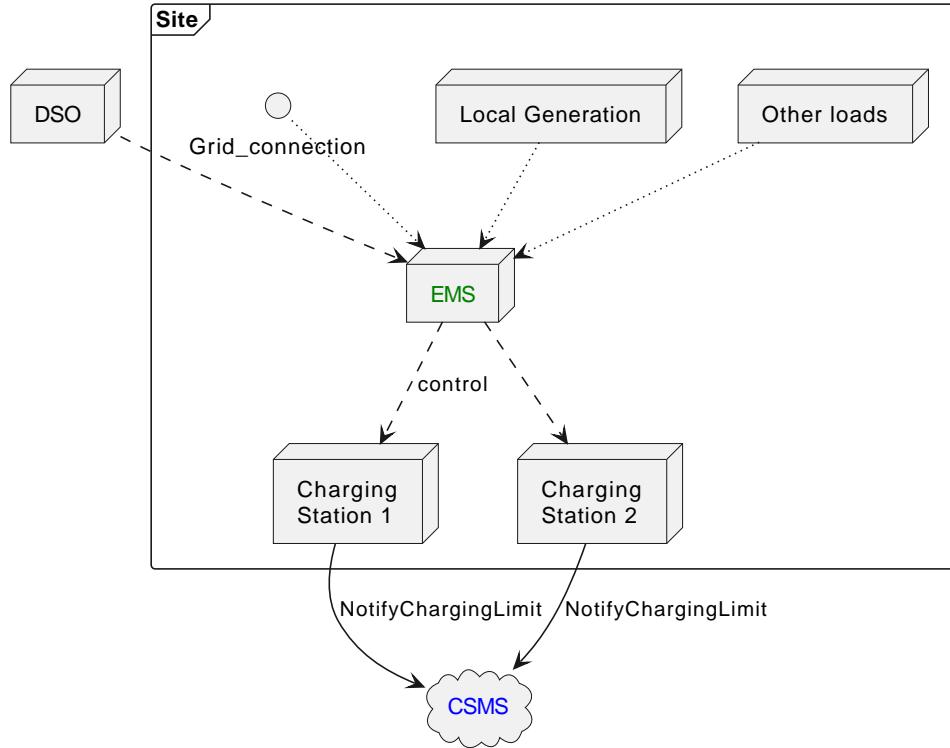
Figure 97. External Smart Charging

New text:

The OCPP protocol is developed for communication between a CSMS and one or more Charging Stations. As described in the above, this means that a CSMS of a Charging Station Operator (CSO) controls a Charging Station and, based on the charging limits of both the EV and the Charging Station, the CSO controls how fast the EV is charged. In some situations there are other factors that might control charging power: A DSO can send signals to change charging power (e.g. via IEC 61850 [IEC61850-7-420], IEC 60870 [IEC60870-5-104], DNP3 [DNP3] or OpenADR [OPENADR]), or a Home Energy Management System or a smart meter may be in place to limit charging power.

An external actor can connect to a Charging Station with any protocol that is supported by the Charging Station for this purpose, like Modbus, EEBUS, and even OCPP. This control signal can be a single limit value or a schedule. In both cases Charging Station will represent the limit internally as a charging profile of purpose ChargingStationExternalConstraints.

A CSMS may need to be informed of changes in charging rate as a result of external signals. OCPP provides a NotifyChargingLimitRequest message to report such changes.



EMS control directly to Charging Stations

2.31. Page 248 - (2025-01) - 3.7 Avoiding Phase Conflicts

The following paragraph about Avoiding Phase Conflicts has been added:

In the situation where a `ChargingStationMaxProfile` or a `ChargingStationExternalConstraints` define a value for `numberPhases` or `phaseToUse`, then a possible conflict might arise if such values are also specified in a `TxDefaultProfile` or `TxProfile`. The following rules apply in that case:

`numberPhases`

The lowest value for a schedule period of all applicable profiles is used for the composite schedule period. If `ChargingStationMaxProfile` has `numberPhases` = 3 and `TxProfile` has `numberPhases` = 1, then the value 1 is used. The same applies to the reverse situation.

`phaseToUse`

When there is a conflicting value of `phaseToUse` between the schedule periods of applicable profiles, then there is no way to create a composite schedule period. For example, a CSMS should not submit a charging profile of purpose `ChargingStationMaxProfile` for `phaseToUse` = 1 and then a `TxProfile` for `phaseToUse` = 3, because the charging station will not know which value has preference. Therefore, a `SetChargingProfileRequest` that causes such a conflict will have to be rejected.

When a relative `TxProfile` is being used and different phases occur in various schedule periods, then it may become difficult to detect if and where such a phase conflict occurs. A charging station should only accept a `SetChargingProfileRequest` when it can be certain, that there is a no risk of a phase conflict. This means, that when the charging station is not able to verify that no phase conflict occurs in any schedule period (which can happen when the `TxProfile` is received for a transaction, but charging has not yet started, so that start time of the first schedule period is not known), that it cannot accept a charging profile if any of the schedule periods contains a value for `phaseToUse` that differs from the value used in the `ChargingStationMaxProfile` or `ChargingStationExternalConstraints`.

NOTE | A value of `phaseToUse` may only be used when `numberOfPhases` = 1.

2.32. Page 286 - (2025-01) - K15 - ISO 15118-2 Charging with load leveling - Requirements

Added recommendation for timestamp when offline:

New / Updated requirements

	ID	Precondition	Requirements	Note
Changed	K15.FR.01	When the Charging Station receives charging needs from the EV	The Charging Station SHALL send a NotifyEVChargingNeedsRequest to the CSMS.	See also K15.FR.20
New	K15.FR.20	K15.FR.01 AND Charging Station is offline	Charging Station SHOULD add timestamp to the NotifyEVChargingNeedsRequest with the time when charging needs were received from EV	This will tell CSMS how old this data is, if it was not immediately sent because of an offline period.
New	K15.FR.21	K15.FR.10	Charging Station SHOULD set <i>selectedScheduleTupleId</i> to the Id of the <i>chargingSchedule</i> that EV selected from the provided ChargingProfileType(s) .	

2.33. Page 286 - (2025-01) - K16 - Renegotiation initiated by CSMS - Requirements

The following requirement was added:

New requirement

ID	Precondition	Requirements	NOTE
K16.FR.14	K16.FR.05	Charging Station SHOULD set <i>selectedScheduleTupleId</i> to the Id of the <i>chargingSchedule</i> that EV selected from the provided ChargingProfileType(s) .	

2.34. Page 292 - (2025-01) - Use case L01 - Added clarification to step 3 about when to start downloading the firmware

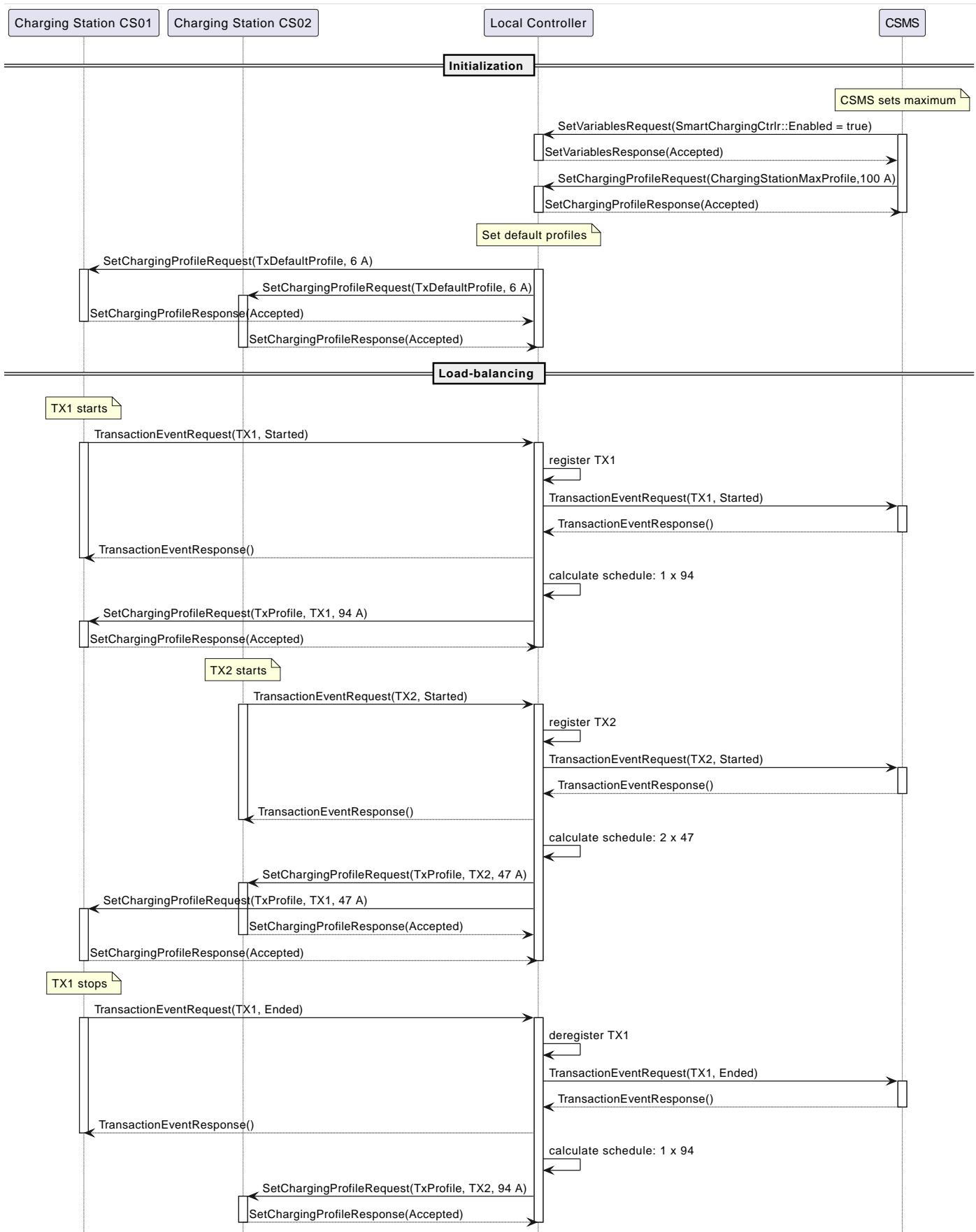
No.	Type	Description
1	Name	Secure Firmware Update
2	ID	L01
3	Objective(s)	Download and install a Secure firmware update.
4	Description	Illustrate how a Charging Station processes a Secure firmware update.
	Actors	CSMS, Charging Station
	Scenario description	<p>1. The CSMS sends an UpdateFirmwareRequest message that contains the location of the firmware, the time after which it should be retrieved, and information on how many times the Charging Station should retry downloading the firmware.</p> <p>2. The Charging Station verifies the validity of the certificate against the Manufacturer root certificate.</p> <p>3. If the certificate is valid AND the retrieveDateTime has passed, the Charging Station starts downloading the firmware, and sends a FirmwareStatusNotificationRequest with status Downloading.</p> <p>If the certificate is not valid or could not be verified, the Charging Station aborts the firmware update process and sends a UpdateFirmwareResponse with status InvalidCertificate and a SecurityEventNotificationRequest with the security event InvalidFirmwareSigningCertificate (See part 2 appendices for the full list of security events).</p> <p>...</p>

2.35. Page 259/260 - (2025-01) - K03 - Updated use case description and sequence diagram

The use case description and sequence diagram have been updated to provide more information on how local load-balancing can be performed. (Requirements for K03 have not changed).

No.	Type	Description
1	Name	Local Smart Charging
2	ID	K03
3	Objective(s)	To illustrate the process of local load-balancing by a Local Cluster.
4	Description	<p>This use case is an example of how local load-balancing can be performed. It does not imply that other approaches would not be correct. The process has been simplified for clarity and should not be regarded as prescriptive.</p> <p>A Local Controller is configured with a value for maximum current for the total cluster by CSMS via a charging profile of type <code>ChargingStationMaxProfile</code> to the Local Controller, or an EMS may have set a <code>ChargingStationExternalConstraints</code> charging profile.</p> <p>The Local Controller divides the maximum current among the active transactions. Whenever a transaction starts or finishes, the Local Controller will update the charging profiles of the remaining transactions to divide the maximum current equally.</p> <p>For simplicity's sake, this use case does not differentiate on departure time or state of charge of vehicles, nor does it take the actual energy consumption of vehicles into account.</p>
	Actors	Charging Station (CS01, CS02), Local Controller (LC), CSMS
	Scenario description	<p>Assume no transactions are active in the local cluster and the maximum current for the local cluster has been configured to be 100 A. The charging stations all have a <code>TxDefaultProfile</code> that allows a current of only 6 A, so that vehicles cannot immediately start charging at full power before the LC had the chance to set a charging profile.</p> <ol style="list-style-type: none"> 1. A transaction starts on charging station CS01. It sends a <code>TransactionEventRequest(Started)</code> to LC. 2. LC is configured to do local load-balancing (i.e. its <code>SmartChargingCtrlr.Enabled</code> = true), so it registers the transaction id TX1 of the transaction that has been started on CS01, before forwarding the message on the websocket for CS01 towards CSMS. 3. LC sends a <code>SetChargingProfileRequest</code> to CS01 with <code>chargingProfilePurpose</code> = <code>TxProfile</code>, <code>chargingProfileKind</code> = <code>Relative</code>, <code>transactionId</code> = TX1 and a <code>chargingSchedule</code> with a <code>chargingRateUnit</code> = A that contains one <code>chargingSchedulePeriod</code> with a <code>limit</code> of 94 A, so that the entire quota is available to this transaction minus the <code>TxDefaultProfile</code> amount for new transactions. 4. Another transaction starts on charging station CS02. It sends a <code>TransactionEventRequest(Started)</code> to LC. 5. LC registers the new transaction id TX2 and forwards the message on the websocket for CS02 to CSMS. 6. LC divides the available quota by allowing each transaction a maximum of 47 A. 7. LC sends a <code>SetChargingProfile</code> message to CS01 that updates the existing <code>TxProfile</code> and sets the <code>limit</code> to 47 A. 8. LC sends new <code>SetChargingProfile</code> to CS02 with <code>chargingProfilePurpose</code> = <code>TxProfile</code>, <code>chargingProfileKind</code> = <code>Relative</code>, <code>transactionId</code> = TX2 and a <code>chargingSchedule</code> with a <code>chargingRateUnit</code> = A that contains one <code>chargingSchedulePeriod</code> with a <code>limit</code> of 47 A. 9. The transaction of CS01 finishes. It sends a <code>TransactionEventRequest(Ended)</code> to LC. 10. LC registers that transaction TX1 on CS01 has finished and forwards the message on the websocket for CS01 to CSMS. 11. LC now allows the maximum to TX2. It sends a <code>SetChargingProfile</code> message to CS02 that updates the existing <code>TxProfile</code> and sets the <code>limit</code> to 94 A. (Note, that the <code>TxProfile</code> for TX1 on CS01 has automatically ceased to exist upon termination of the transaction.)
5	Prerequisites	The LC has been configured with a fixed maximum current level. The <code>SmartChargingCtrlr</code> component of Local Controller has been Enabled, which will trigger the Local Controller to read and interpret <code>TransactionEventRequest</code> messages from connected Charging Stations.
6	Post conditions	
7	Error Handling	

No.	Type	Description
8	Remarks	<p>As described in Part 1, a Local Controller replicates all websockets from Charging Stations in the cluster towards CSMS, and forwards messages from Charging Station to CSMS on the appropriate websocket (and vice versa). This allows the Local Controller to read messages, such as a TransactionEventRequest message, from the Charging Station.</p> <p>The Local Controller for local smart charging can be implemented in different ways, for example: as a separate physical component or as part of a "master" Charging Station controlling a number of other Charging Stations. The Local Controller MAY or MAY NOT have any EVSEs of its own.</p> <p>The limits on Charging Stations in a Local Smart Charging group can either be pre-configured in the Local Controller in one way or another, or they can be set by the CSMS. The Local Controller contains the logic to distribute this capacity among the connected EVSEs by adjusting their limits as needed.</p>



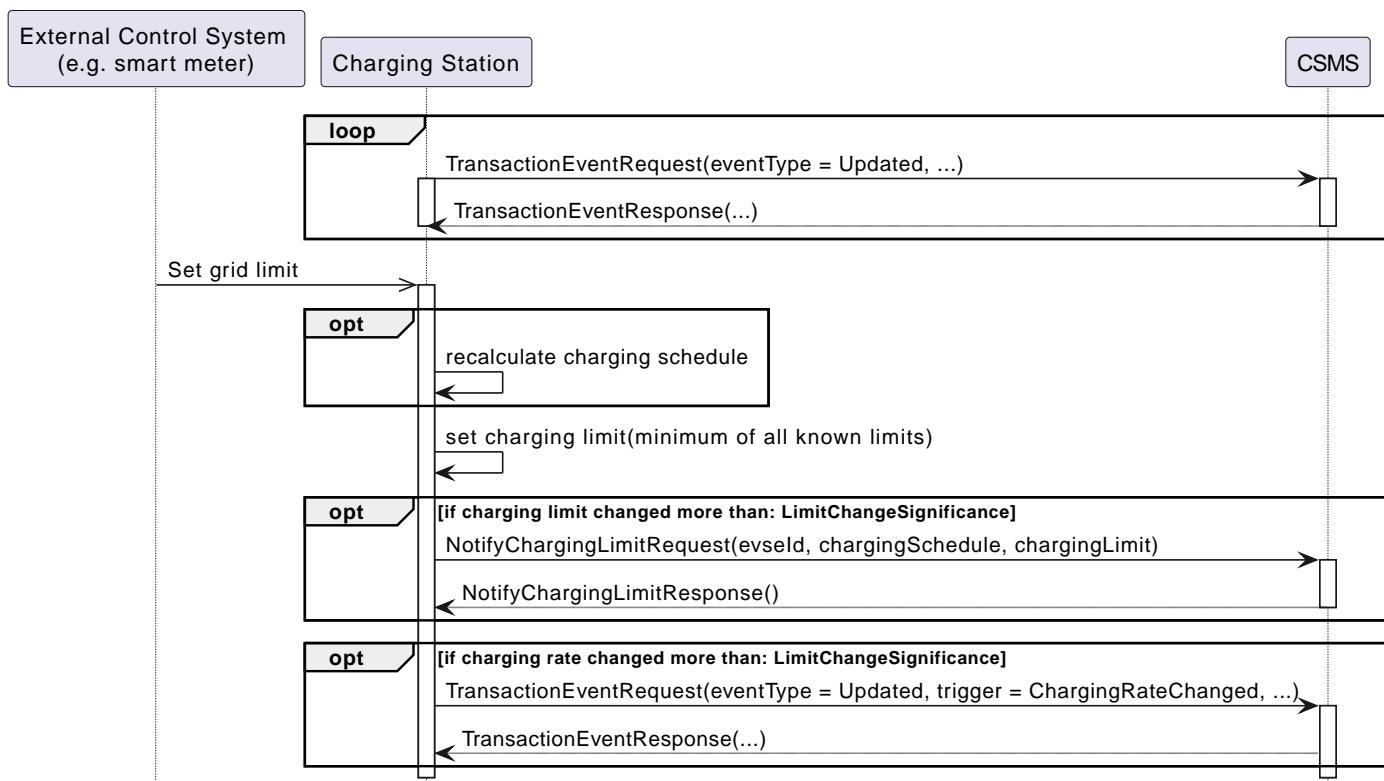
Local Controller performing local load-balancing

2.36. Page 274/275/276/277 - (2025-01) - K11/K12 - Updated use case descriptions and sequence diagrams

The use case descriptions and sequence diagrams have been updated to describe the more likely scenario of a smart meter or EMS as external actor.

K11 - Set / Update External Charging Limit With Ongoing Transaction

No.	Type	Description
1	Name	Set / Update External Charging Limit With Ongoing Transaction
2	ID	K11
3	Objective(s)	To inform the CSMS of a charging schedule or charging limit imposed by an External Control System on the Charging Station with ongoing transaction(s).
4	Description	An External Control System sends a charging limit/schedule to a Charging Station. This limit is sent to the CSMS. The External Control System can be a DSO, but also a smart meter or a home energy management system. The interface between External Control System and Charging Station is not specified. It can be any protocol that is supported by Charging Station for this purpose, even OCPP.
	Actors	External Control System, Charging Station, CSMS
	Scenario description	<ol style="list-style-type: none"> 1. External control system sends charging limit/schedule to Charging Station. 2. Optional: Charging Station calculates new charging schedule. 3. Charging Station adjusts the charging speed of the ongoing transaction(s). 4. If the charging limit changed by more than: LimitChangeSignificance, the Charging Station sends a NotifyChargingLimitRequest message to CSMS with optionally the set charging limit/schedule. 5. The CSMS responds with NotifyChargingLimitResponse to the Charging Station. 6. If the charging rate changes by more than: LimitChangeSignificance, the Charging Station sends a TransactionEventRequest message to inform the CSMS. 7. The CSMS responds with TransactionEventResponse to the Charging Station.
5	Prerequisites	Charging Station is not in error state. The external system can set/clear a charging limit/schedule on the Charging Station via a direct connection to the Charging Station.
...		

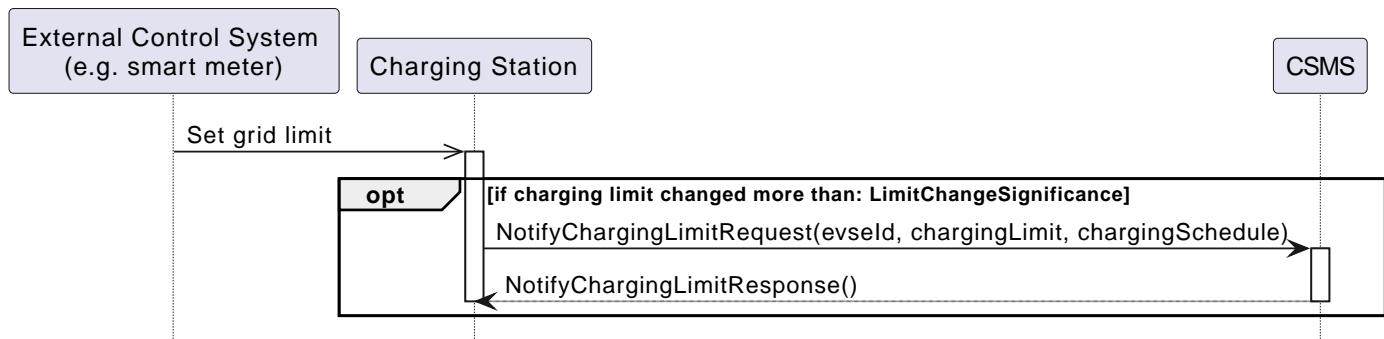


Sequence diagram of the use case "Setting / Updating External Charging Limit with Ongoing Transaction"

K12 - Set / Update External Charging Limit Without Ongoing Transaction

No.	Type	Description
1	Name	Set / Update External Charging Limit Without Ongoing Transaction

No.	Type	Description
2	ID	K12
3	Objective(s)	To inform the CSMS of a charging schedule or charging limit imposed by an external system on the Charging Station for new transactions or on the grid connection.
4	Description	To inform the CSMS of a charging schedule or charging limit imposed by an external system on the Charging Station for new transactions or on the grid connection. The External Control System can be a DSO, but also a smart meter or a home energy management system. The interface between External Control System and Charging Station is not specified. It can be any protocol that is supported by Charging Station for this purpose, even OCPP.
	Actors	External Control System, Charging Station, CSMS
	Scenario description	<ol style="list-style-type: none"> 1. External Control System sends a charging limit to Charging Station (not during a transaction). 2. Optional: Charging Station calculates new charging schedule. 3. Charging Station adjusts the charging speed. 4. If the charging limit changed by more than: LimitChangeSignificance, the Charging Station sends a NotifyChargingLimitRequest message to CSMS with optionally the set charging limit/schedule. 5. The CSMS responds with a NotifyChargingLimitResponse to the Charging Station.
5	Prerequisites	<p>Charging Station is not in error state. The external system can set/clear a charging limit/schedule on the Charging Station via a direct connection to the Charging Station.</p>
...		



Sequence diagram of the use case "Set / Update External Charging Limit Without Ongoing Transaction"

2.37. Page 278/279 - (2025-01) - K13 - Updated requirement preconditions

Moved precondition "transaction is ongoing" from K13.FR.01 to K13.FR.03.

Changed K13 requirements:

	ID	Precondition	Requirements
Old text	K13.FR.01	A transaction is ongoing AND External charging limit is released/removed	The Charging Station SHALL NOT limit charging anymore based on the previously received limit.
New text	K13.FR.01	External charging limit is released/removed	The Charging Station SHALL NOT limit charging anymore based on the previously received limit.
Old text	K13.FR.03	K13.FR.01 AND Charging rate changed by more than: LimitChangeSignificance	The Charging Station SHALL send a TransactionEventRequest message to the CSMS with trigger = ChargingRateChanged .

	ID	Precondition	Requirements
New text	K13.FR.03	K13.FR.01 AND A transaction is ongoing AND Charging rate changed by more than: LimitChangeSignificance	The Charging Station SHALL send a TransactionEventRequest message to the CSMS with trigger = ChargingRateChanged .

2.38. Page 279 - (2025-01) - K14 - Updated use case scenario description

Improved scenario step description 5 and 7.

No.	Type	Description
1	Name	Handle external charging limit with a local controller
2	ID	K14
3	Objective(s)	To adjust the charging limits according to the External Control System requirements.
4	Description	An external control system sends a charging limit to the Local Controller. The Local Controller notifies the CSMS, calculates the new charging schedules and sends a SetChargingProfileRequest messages to all Charging Stations for which the charging profile has changed.
	Actors	External control system, Local Controller, Charging Station, CSMS
	Scenario description	<ol style="list-style-type: none"> 1. External control system sends a charging limit/schedule to Local Controller. 2. Local Controller sends a NotifyChargingLimitRequest message to the CSMS. 3. Local Controller calculates new Charging Profiles for all connected Charging Stations. 4. Local Controller sends a SetChargingProfileRequest message to all Charging Stations for which the charging profile has changed. 5. External control releases a charging limit/schedule to Local Controller. 6. Local Controller sends a ClearedChargingLimitRequest message to the CSMS. 7. Local Controller clears Charging Profiles for all connected Charging Stations. 8. Local Controller sends a ClearChargingProfileRequest messages to all affected Charging Stations.
		...

2.39. Page 306 - (2025-01) - M. ISO 15118 Certificate Management

The functional block ISO 15118 Certificate Management is renamed to Certificate Management since this also contains certificate management of non-ISO 15118 related certificates.

2.40. Page 308 - (2025-01) - Update introduction sequence diagram ISO 15118

Not all ISO 15118 message sequences were complete, so the sequence diagram has been updated to show a more complete version of the message flow.

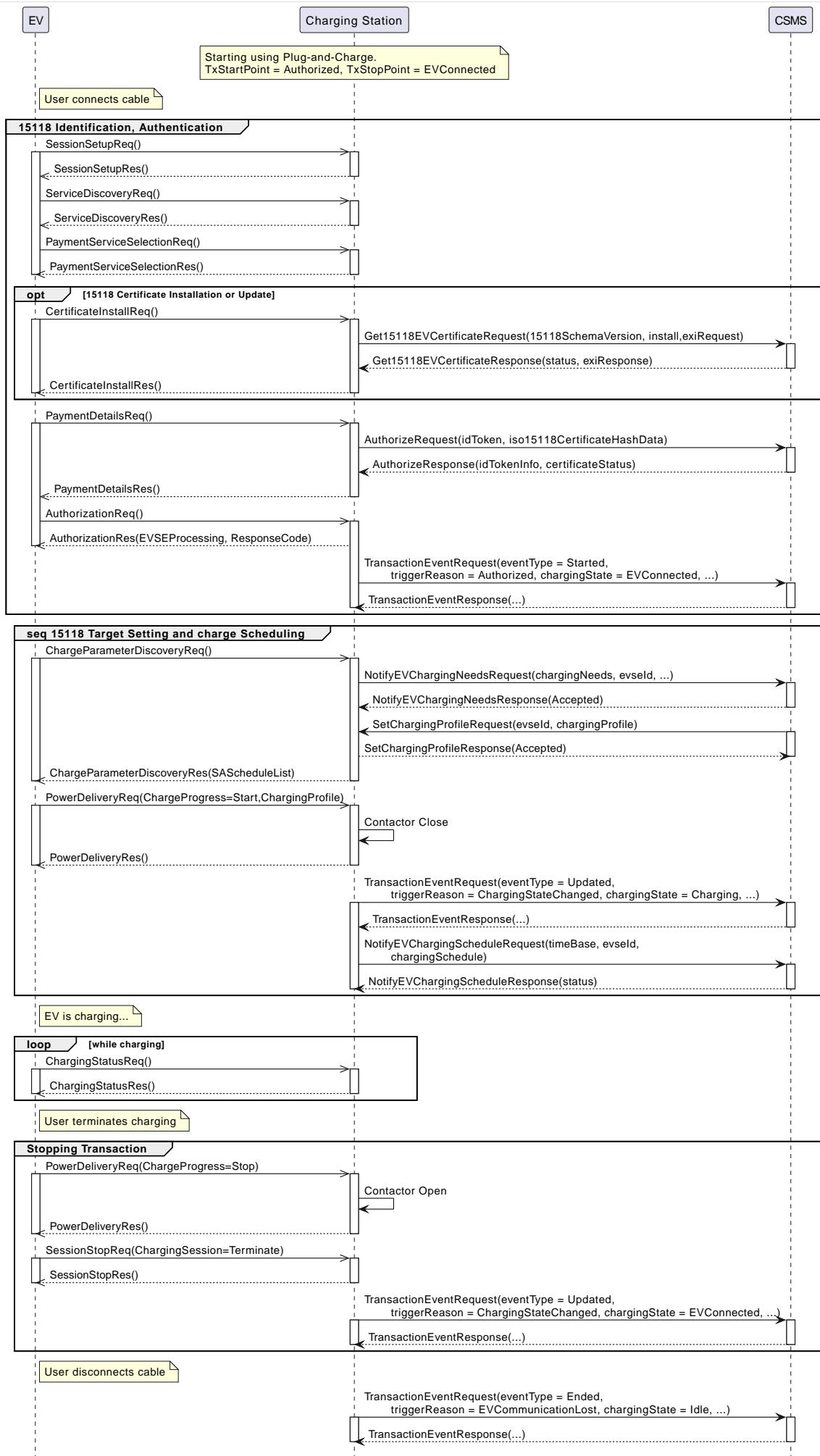


Figure 122. Sequence with Authorization and Scheduling with ISO 15118-2

NOTE

The time-out on the ChargeParameterDiscoveryReq is 2 seconds, but this can be prolonged up to 60 seconds to wait for charging profile (SAScheduleList) to be provided by the CSMS. See ISO 15118-2 [ISO15118-2].

2.41. Page 330 - (2025-01) - N03 Set Monitoring Base: Improved text of Remark

Old	8	Remark(s)	<p>Upon receipt of a SetMonitoringBaseRequest for HardWiredOnly or FactoryDefault the Charging Station will discard of any previously configured custom monitors and will activate the monitoring settings that are related to given MonitoringBase.</p> <p>For a MonitoringBase = All the Charging Station will activate all pre-configured monitors and leave previously configured custom monitors intact. This includes the custom monitors that were created when changing an existing pre-configured monitor.</p> <p>When the set of pre-configured monitors for All and FactoryDefault is the same, then the difference between the two is, that with FactoryDefault all custom monitors are deleted before the factory default pre-configured monitors are restored.</p>
New	8	Remark(s)	<p>Upon receipt of a SetMonitoringBaseRequest for:</p> <ul style="list-style-type: none"> • <i>monitoringBase = HardWiredOnly</i>: the Charging Station will deactivate all pre-configured monitors and remove any previously configured custom monitors. Only the HardWiredMonitor monitors remain. • <i>monitoringBase = FactoryDefault</i>: the Charging Station will (re)activate all PreconfiguredMonitor monitors and remove all custom monitors. • <i>monitoringBase = All</i>: the Charging Station will activate all pre-configured monitors and leave previously configured CustomMonitor monitors intact. This includes the custom monitors that were created when changing an existing pre-configured monitor.

2.42. Page 331 - (2025-01) - N03.FR.04: text improvement

Improved requirement text for N03.FR.04.

Old	N03.FR.04	N03.FR.01 AND When the Charging Station received a setMonitoringBaseRequest with <i>monitoringBase FactoryDefault</i>	Then the Charging Station SHALL delete all custom monitors (including overruled pre-configured monitors) and activate the default monitoring settings as recommended by the manufacturer.
New	N03.FR.04	N03.FR.01 AND When the Charging Station received a setMonitoringBaseRequest with <i>monitoringBase FactoryDefault</i>	Then the Charging Station SHALL delete all custom monitors (including overruled pre-configured monitors) and activate the pre-configured monitors of the Charging Station .

Page 434 - MonitoringBaseEnumType description update

The description has been updated to be better aligned with the use case.

Value	Description
All	Activate all pre-configured monitors while leaving custom monitors intact, including those that overrule a pre-configured monitor.
FactoryDefault	(Re)activate the default monitoring settings has recommended by the manufacturer. This is a subset of all pre-configured monitors of Charging Station and remove all custom monitors.
HardWiredOnly	Removes all custom monitors and disables all pre-configured monitors.

2.43. Page 350 - (2025-02) - 001 - Added missing requirements

Added missing requirements explicitly specifying behaviour of Charging Station it contains one or more displays.

New requirements

ID	Precondition	Requirement definition
<i>Multiple Display support</i>		
001.FR.20	When Charging Station has multiple displays AND Charging Station receives a [setdisplaymessagerequest] without a <i>display</i> element in its <i>MessageInfoType</i>	Charging Station SHOULD use the message for the main display(s)
001.FR.21	When receiving a <i>GetBaseReportRequest</i> AND Charging Station has one or more displays	Charging Station SHOULD include in the report a <i>Display</i> component for every display it contains.
001.FR.22	When Charging Station receives a [setdisplaymessagerequest] with <i>Display</i> element referencing an unknown <i>Display</i> in its <i>MessageInfoType</i>	Charging Station SHOULD respond with a [setdisplaymessageresponse] with <i>status</i> = <i>Rejected</i> .
001.FR.23	When Charging Station receives a [setdisplaymessagerequest] with <i>Display</i> element referencing a known <i>Display</i> in its <i>MessageInfoType</i>	Charging Station SHOULD use the message only for the specified display.

2.44. Page 446 - (2025-01) - ActiveNetworkProfile is incorrectly marked as optional

The referenced variable OCPPCommCtrlr.ActiveNetworkProfile is incorrectly marked as optional. As the description states this variable needs to be implemented when the Charging Station supports the NetworkConnectionProfile use cases B09/B10. These use cases are an integral part of the Core of OCPP. In addition, all other NetworkConnectionProfile related referenced variables are already required; OCPPCommCtrlr.NetworkConfigurationPriority and OCPPCommCtrlr.NetworkProfileConnectionAttempts. Therefore, OCPPCommCtrlr.ActiveNetworkProfile should also be marked as required.

ActiveNetworkProfile

Required	yes	
Component	componentName	OCPPCommCtrlr
Variable	variableName	ActiveNetworkProfile
	variableAttributes	mutability
	variableCharacteristics	readOnly
Description	This variable indicates the NetworkConnectionProfile configuration slot the Charging Station currently uses for its connection with the CSMS.	

2.45. Page 453 - (2025-01) - References to *monitorValue* changed to *value* [354]

The specification in N Diagnostics consistently refers to a variable *monitorValue*, but this variable is called *value* in JSON schemas.

All occurrences of *monitorValue* in N Diagnostics have been changed to *value*.

2.46. Page 454 - (2025-01) - N04.FR.06 Improved limit definition of thresholds [353]

The requirement has been defined more exact, because limits for upper and lower differ slightly.

	ID	Precondition	Requirement definition	Note
Old	N04.FR.06	When the Charging Station receives a SetVariableMonitoringRequest with monitor type UpperThreshold or LowerThreshold AND the <i>monitorValue</i> is lower or higher than the range of the given Variable	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: Rejected .	More information can be provided in the optional statusInfo element.
New	N04.FR.06	When the Charging Station receives a SetVariableMonitoringRequest with (monitor type = UpperThreshold AND monitorValue < minLimit OR monitorValue > maxLimit) OR (monitor type = LowerThreshold AND monitorValue < minLimit OR monitorValue > maxLimit)	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: Rejected .	minLimit and maxLimit refer to the [cmn_variablecharacteristicstype] for the [cmn_variabletype] . Be aware that setting a UpperThreshold to the maxLimit or setting a LowerThreshold to the minLimit will result in a monitor that will never trigger. More information on the reason of rejection can be provided in the optional statusInfo element.

2.47. Page 456 - (2025-02) - New configuration variable to allow TLS wildcard certificates

New configuration key

AllowCSMSTLSWildcards

Required	no		
Component	componentName	SecurityCtrlr	
Variable	variableName	AllowCSMSTLSWildcards	
	variableAttributes	mutability	ReadWrite
	variableCharacteristics	dataType	boolean
Description	This variable allows a Charging Station to support non-compliant OCPP behavior and connect to a CSMS that uses a wildcard TLS server certificate for the OCPP connection. If this variable is present it SHALL be ReadWrite. If this variable is not implemented or has value false, the OCPP-compliant behavior is that a Charging Station rejects a connection from a CSMS that presents a wildcard certificate. It is highly RECOMMENDED to not allow wildcard certificates.		

2.48. Page 458 - (2025-01) - Added optional variable to allow the Charging Station to report its supported idTokenTypeS

Currently there is no method for the Charging Station to report which idTokenTypeS it supports. This would be very useful for a CSMS to know and will improve the automated onboarding of Charging Stations.

SupportedIdTokenTypeS

Required	no
----------	----

Component	componentName	AuthCtrlr	
Variable	variableName	SupportedIdTokenType	
	variableAttributes	mutability	ReadOnly
	variableCharacteristics	dataType	MemberList
		valuesList	List of IdTokenEnumType.
Description	The subset of the list of supported IdTokenType as defined by IdTokenEnumtype, that is supported by the Charging Station.		

2.49. Page 462 - (2025-01) - Added note to EnergyTransfer description as TxStartPoint

A note has been added to the description of the EnergyTransfer TxStartPoint to warn for potential skews of the values of the energy meter readings associated with start of the transaction.

Value	Description
EnergyTransfer	<p>Energy is being transferred between EV and EVSE.</p> <p>Note: Since energy needs to start flowing first to cause the transaction to be started, there is a small time gap (order of milliseconds) between the start of energy transfer and start of transaction. Depending on the implementation this may potentially skew the value of the energy meter reading associated with start of the transaction. Use PowerPathClosed as TxStartPoint to avoid this situation.</p>

2.50. Page 476 - (2025-01) - Added Connector component to AvailabilityState referenced variable

It was already possible to report the AvailabilityState of the Connector component, however the definition was missing at this table.

AvailabilityState

Required	yes		
Components	componentName	ChargingStation	
		EVSE	
		Connector	
	evse	* (for EVSE and Connector)	
Variable	variableName	AvailabilityState	
	variableAttributes	mutability	ReadOnly
	variableCharacteristics	dataType	optionList
		valuesList	Available, Occupied, Reserved, Unavailable, Faulted
Description	<p>This variable reports current availability state for the ChargingStation, EVSE and Connector. When this variable reports the Connector AvailabilityState, it replicates the connectorStatus values as would be reported by the StatusNotification messages.</p> <p>An EVSE or Connector component is addressed on its own tier. So, EVSE #1 is addressed as component EVSE on tier evse.id = 1, and EVSE #1, Connector #1 is addressed as component Connector on tier evse.id = 1, evse.connectorId = 1.</p>		

3. Part 3

Currently no new errata for OCPP 2.0.1 part 3.

4. Part 4

4.1. Page 6 - (2025-01) - 3.1.1. The connection URL

The following clarifying text was added:

Old text	[...] percent-encoded [...]
New text	[...] percent-encoded / URL encoded [...]

4.2. Page 7 - (2025-01) - 3.1.2. OCPP version

The following text has been rewritten for clarity:

Old text	The OCPP version should not be part of the OCPP-J endpoint URL string if you want to select the OCPP version to use via the websocket protocol negotiation mechanism, as explained in Server Response .
New text	If the OCPP version is part of the OCPP-J endpoint URL it SHALL not determine the OCPP version to use, because the OCPP version is selected via the websocket protocol negotiation mechanism, as explained in section 3.3 [server-response] .

4.3. Page 8 - (2025-01) - 3.3. WebSocket Compression

The following text was duplicate. The duplication has been removed and the text has been clarified:

Old text	OCPP Requires the CSMS (and Local Controller) to support RFC 7692, WebSocket compression is seen as a relative simple way to reduce mobile data usage. For a Charging Station this is not a hard requirement, as this might be more complex to implement on an embedded platform, but as this is seen as efficient solution to reduce mobile data usage, it is RECOMMENDED to be implemented on a Charging Station that uses a mobile data connection. OCPP Requires the CSMS (and Local Controller) to support RFC 7692, WebSocket compression is seen as a relative simple way to reduce mobile data usage. For a Charging Station this is not a hard requirement, as this might be more complex to implement on an embedded platform, but as this is seen as efficient solution to reduce mobile data usage, it is RECOMMENDED to be implemented on a Charging Station that uses a mobile data connection.
New text	The CSMS (and Local Controller) SHALL support RFC 7692 , WebSocket compression, which is a relative simple way to reduce mobile data usage. For a Charging Station this is not a hard requirement, as this might be more complex to implement on an embedded platform. It is RECOMMENDED to be implemented on a Charging Station, because it is an efficient solution to reduce mobile data usage.

4.4. Page 10 - (2025-01) - 4.1.3. The message type

OLD: |CALL|2| Request message |CALLRESULT|3| Response message

New:

|CALL|2| Request message, i.e. messages ending in "Request" |CALLRESULT|3| Response message, i.e. message ending in "Response"

4.5. Page 10 - (2025-01) - 4.1.3. The message type

Refer to 'system', instead of 'server'.

Old text	When a server receives a message with a Message Type Number not in this list, it SHALL ignore the message payload. Each message type may have additional required fields.
New text	When a system receives a message with a Message Type Number not in this list, it SHALL ignore the message payload. Each message type may have additional required fields.

4.6. Page 10 - (2024-09) - 4.1.4. Message ID

Change the following text in paragraph 4.1.4:

Old text	The message ID serves to identify a request. A message ID for any CALL message MUST be different from all message IDs previously used by the same sender for any other CALL messages on any WebSocket connection using the same unique Charging Station identifier. This also applies to retries of messages.
New text	The message ID serves to identify a request. A message ID for any CALL message MUST be different from all message IDs previously used by the same sender for any other CALL messages on any WebSocket connection using the same unique Charging Station identifier. The message ID for a retried message (e.g. when no response was received within timeout) MAY be identical to the message ID of the original message.

4.7. Page 13 - (2025-01) - 4.2.3. CALLERROR

In the example, the ErrorDescription text has been updated:

Old text	"SetDisplayMessageRequest not implemented"
New text	"SetDisplayMessageRequest not supported "

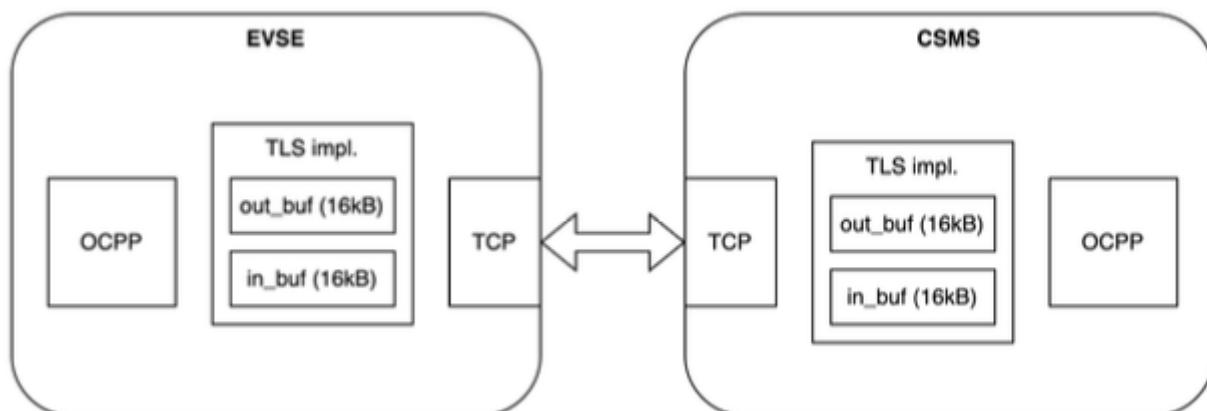
4.8. Page 15 - (2025-01) - 5. Connection

The following clarifying text has been added related to the TLS fragment length:

4.9. TLS fragment length

TLS involves sending "Records" between peers. Records can be of type "Handshake", "Alert", "ChangeCipherSpec", "Heartbeat" or "Application". OCPP messages are sent in Application records. The payload contains a "fragment" of the application data. The record layer fragments information blocks into TLSPlaintext records carrying data in chunks of 2^{14} bytes (16kB) or less.

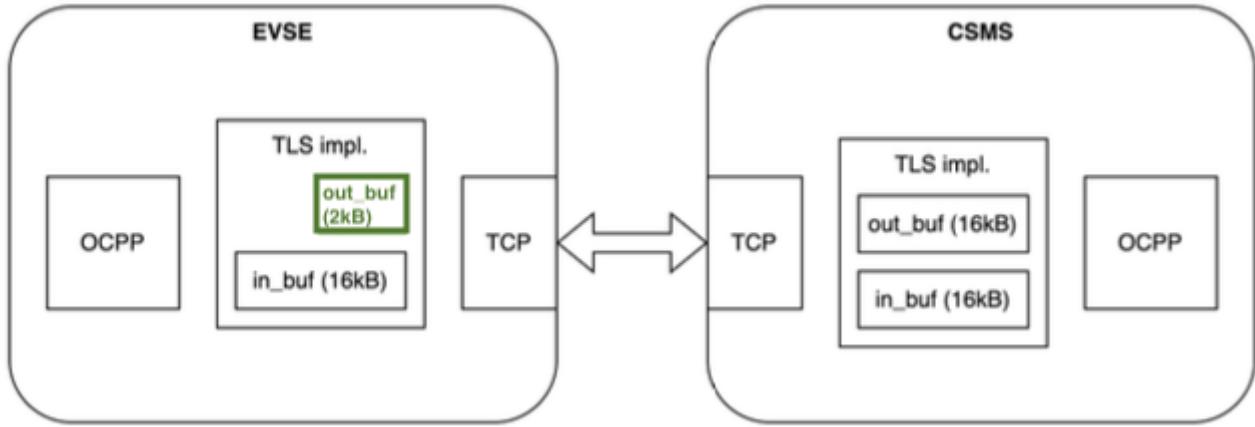
TLS peers need to maintain an input and an output buffer to store an entire fragment of 16 kB. For a low resource device it is a large cost to allocate 32 kB for the TLS connection.



Peers allocating standard 16 kB TLS buffers

A TLS extension is defined in TLS Extensions RFC6066 Section 4, that allows the client to ask for a different maximum fragment length than the default 16kB. A client can ask for a maximum fragment length of 0.5 kB, 1 kB, 2 kB or 4 kB. This TLS extension is, however, not widely supported and native managed cloud TLS termination services typically don't support this.

A resource-constrained Charging Station SHOULD try to negotiate a smaller TLS maximum fragment size, and if that is not accepted by the peer, then Charging Station MAY unilaterally decide to allocate less memory to its TLS output buffer. A TLS maximum fragment length of 2 kB is suggested based on data collection during certification tests, which shows that 99% of the messages fit in a 2 kB buffer.



Charging Station allocating a 2 kB TLS output buffer

4.10. Page 15 - (2025-01) - 5.3. WebSocket Ping in relation to OCPP Heartbeat

The following clarifying text was added:

New text	A Heartbeat message checks connectivity end-to-end, whereas a Websocket ping/pong only checks from point-to-point. This makes a difference in an extended network topology with a Local Controller between Charging Station and CSMS.
----------	---

4.11. Page 15 - (2025-01) - 5 Connection - Added section about TLS fragment length

As a result of the discussions at the OCPP lite taskgroup, a section has been added providing guidance on implementing TLS fragment negotiation.

4.12. Page 16 - (2025-04) - 5.3 Reconnecting - reset backoff wait timer

The RetryBackOffWaitMinimum timer is to be used the first time it tries to connect. A sentence has been added to below paragraph to make it explicit that it needs to be reset after successful connection.

The first reconnection attempts SHALL be after a back-off time of: `RetryBackOffWaitMinimum` seconds, plus a random value with a maximum of `RetryBackOffRandomRange` seconds. After every failed reconnection attempt the Charging Station SHALL double the previous back-off time, with a maximum of `RetryBackOffRepeatTimes`, adding a new random value with a maximum of `RetryBackOffRandomRange` seconds to every reconnection attempt. After `RetryBackOffRepeatTimes` reconnection attempts, the Charging Station SHALL keep reconnecting with the last back-off time, not increasing it any further. **After a successful connection the backoff wait timer SHALL be reset to `RetryBackOffWaitMinimum` seconds.**

4.13. Page 18 - (2025-02) - 6.3 Connection loss - Allow Local Controller to keep connection open

The sentence in this section was too strict about requiring a Local Controller to always close the connection with its charging stations when the connection with CSMS is lost. The sentence has been updated in order to allow for Local Controller implementations that are able to manage the local charging stations locally (for a limited time) when the connection with CSMS is down.

Old text	Whenever one or more WebSocket connections between CSMS and the Local Controller are lost, the Local Controller SHALL close all corresponding WebSockets to the Charging Stations that are connected to it.
----------	---

New text	Whenever one or more WebSocket connections between CSMS and the Local Controller are lost, the Local Controller SHALL close all corresponding WebSockets to the Charging Stations that are connected to it, unless the Local Controller is capable of responding to Charging Station requests, and forwards transaction-related requests to the CSMS once the connection is restored.
----------	---

5. Part 5

5.1. General - (2025-02) - Renamed OCTT to Test System

Updated (2025-04)

In the entire document, the term A few references to "OCTT" have been replaced by "Test System" (mostly "OCTT Id" has been replaced by "TC id").

5.2. Page 7 - (2025-02) - Changed definition of C-01 Support for offline authorization of transactions

The current definition does not take into account the supported local authorization methods in relation to the features to locally store idTokens.

Old	C-01	Support for offline authorization of transactions	Optional. Supporting this feature depends on whether at least one of the following is supported; - Certification Profile: Local Authorization List Management - C-02: Support for allowing offline authorization for unknown ids - C-49: Authorization Cache (AuthCacheEnabled)
New	C-01	Support for offline authorization of transactions	Conditional. Supporting this feature depends on whether at least one of the following feature combinations is supported; - Certification Profile: Local Authorization List Management AND at least one of the following local authorization options; C-30 or C-31 or C-32 or C-34. - C-02: Support for allowing offline authorization for unknown ids AND at least one of the following local authorization options; C-30 or C-31 or C-32 or C-33 or C-34. - C-49: Authorization Cache AND at least one of the following local authorization options; C-30 or C-31 or C-32 or C-34. - C-35: Local Authorization - NoAuthorization (Because there is no authorization, no local authorization mechanism is needed.)

5.3. Page 9 - (2024-11) - Optional feature list for charging station - Change name R-3

The specified name of feature R-3 is not correct. It should reflect the ability to disable reservations.

Old	R-3	Reservation support (ReservationEnabled)	Configuration Variable for H01
New	R-3	Support for disabling Reservations (ReservationEnabled)	Configuration Variable for H01

5.4. Page 9 - (2025-02) - Optional feature list for charging station - Added AQ-10 to make TC_N_48_CS conditional

Added AQ-10 to make TC_N_48_CS conditional.

Id	Additional questions for lab testing
AQ-10	Does your Charging Station support setting a Delta monitor on the WriteOnly component.variable SecurityCtrlr.BasicAuthPassword?

5.5. Page 9/11/27/48/52 - (2025-02) - Removed feature ISO-3, added additional question AQ-11

Page 9 - Optional feature list for charging station

	ISO 15118 support		
Removed	ISO-3	Combined charging station Certificate (for both OCPP and ISO 15118)	Optional

Page 11 - Optional feature list for CSMS

	ISO 15118 support		
Removed	ISO-3	Combined charging station certificate (for both OCPP and ISO 15118)	Optional

Page 27 - Test Cases Core

		Trigger message					
Old	TC_F_27	NotImplemented	C	M	For CS: can only be done when SignCombinedCertificate is notimplemented	NOT ISO-3	
New	TC_F_27	NotImplemented	C	M	For CS: can only be done when SignCombinedCertificate is notimplemented	NOT AQ-11	

Page 48 - Questions for Charging Stations

	Id	Additional questions for lab testing
Added	AQ-11	Does your Charging Station support a combined charging station Certificate (for both OCPP and ISO 15118)

Page 52 - Appendix C: Features vs. OCPP use cases

	Id	Feature				Related use cases
	ISO 15118 support					
Removed	ISO-3	Combined charging station certificate (for both OCPP and ISO 15118)				A02/A03

5.6. Page 13 - (2025-02) - Updated Id's of the additional questions of CSMSs in the appendix

The following Id's of additional questions for CSMSs have been updated:

Old	TC_B_30	Pending/Rejected - SecurityError	M	C	For CSMS: if CSMS can be configured to first respond to a BootNotificationRequest with status Pending or Rejected	C-44 or NOT AQ-6	BootNotification Pending or Does the CSMS reject unknown Charging Stations during websocket connection setup?
New	TC_B_30	Pending/Rejected - SecurityError	M	C	For CSMS: if CSMS can be configured to first respond to a BootNotificationRequest with status Pending or Rejected	C-44 or NOT AQ-16	BootNotification Pending or Does the CSMS reject unknown Charging Stations during websocket connection setup?
Old	TC_B_31	Pending/Rejected - TriggerMessage		C	For CSMS: if CSMS can be configured to first respond to a BootNotificationRequest with status Pending or Rejected	C-44 or NOT AQ-6	BootNotification Pending or Does the CSMS reject unknown Charging Stations during websocket connection setup?
New	TC_B_31	Pending/Rejected - TriggerMessage		C	For CSMS: if CSMS can be configured to first respond to a BootNotificationRequest with status Pending or Rejected	C-44 or NOT AQ-16	BootNotification Pending or Does the CSMS reject unknown Charging Stations during websocket connection setup?

5.7. Page 17/33 - (2025-02) - Test Cases Local Authorization List Management & Authorization Cache - Update conditions

Testcases marked as mandatory have conditions that are not applicable and should be removed. The CWG decided that online local authorization list and authorization cache testcase can also be tested with remote authorization. In addition, local auth method C-34 MacAddress has been added to the offline conditions as possible method to use.

Authorization Cache

	Store Authorization Data in the Authorization Cache					
TC_C_32	Persistent over reboot	C		If the Charging Station has an authorization cache, then it must support this use case	C-49	Authorization Cache

	Store Authorization Data in the Authorization Cache					
TC_C_33	Update on AuthorizeResponse	C		If the Charging Station has an authorization cache, then it must support this use case	C-49	Authorization Cache
TC_C_34	Update on TransactionResponse	C		If the Charging Station has an authorization cache, then it must support this use case	C-49	Authorization Cache
TC_C_36	AuthCacheCtrlr.LocalPreAuthorize = false	C		If the Charging Station has an authorization cache and AuthCacheEnabled is implemented	C-49	Authorization Cache
TC_C_46	AuthCacheLifeTime	C		If the Charging Station has an authorization cache and supports to set a lifetime for its entries.	C-49 and C-53	Authorization Cache & AuthCacheLifeTime
	Clear Authorization Data in Authorization Cache					
TC_C_37	Accepted	C	M	If the Charging Station has an authorization cache, then it must support this use case	C-49	Authorization Cache
TC_C_38	Rejected	C	M	If the Charging Station has an authorization cache and AuthCacheEnabled is implemented	C-49	Authorization Cache
	Authorization by GroupId					
TC_C_41	Success with Authorization Cache	C		For CS: - The Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34 - If the Charging Station has an authorization cache.	C-49 and (C-30 or C-31 or C-32 or C-34)	Authorization Cache
TC_C_44	Invalid status with Authorization Cache	C		For CS: - The Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34 - If the Charging Station has an authorization cache.	C-49 and (C-30 or C-31 or C-32 or C-34)	Authorization Cache
	Authorization through authorization cache					
TC_C_08	Accepted	C	M	If the Charging Station has an authorization cache	C-49	Authorization Cache
TC_C_09	Invalid & Not Accepted	C		If the Charging Station has an authorization cache	C-49	Authorization Cache
TC_C_12	Invalid & Accepted	C		If the Charging Station has an authorization cache	C-49	Authorization Cache
TC_C_10	Blocked	C		If the Charging Station has an authorization cache	C-49	Authorization Cache
TC_C_11	Expired	C		If the Charging Station has an authorization cache	C-49	Authorization Cache
TC_C_13	Accepted but cable not connected yet.	C		If the Charging Station has an authorization cache	C-49	Authorization Cache
TC_C_15	StopTxOnInvalidId = false, MaxEnergyOnInvalidId > 0	C		If the Charging Station has an authorization cache AND the Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34 If MaxEnergyOnInvalidId is implemented.	C-49 and C-03 and (C-30 or C-31 or C-32 or C-34)	Authorization Cache & MaxEnergyOnInvalidId

	Store Authorization Data in the Authorization Cache					
TC_C_16	StopTxOnInvalidId = true	C		If the Charging Station has an authorization cache AND the Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-49 and (C-30 or C-31 or C-32 or C-34)	Authorization Cache
TC_C_17	StopTxOnInvalidId = false	C		If the Charging Station has an authorization cache AND the Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-49 and (C-30 or C-31 or C-32 or C-34)	Authorization Cache
TC_C_18	StopTxOnInvalidId = true, MaxEnergyOnInvalidId > 0	C		If the Charging Station has an authorization cache AND the Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34 If MaxEnergyOnInvalidId is implemented.	C-49 and C-03 and (C-30 or C-31 or C-32 or C-34)	Authorization Cache & MaxEnergyOnInvalidId
TC_C_20	Invalid		M			
TC_C_57	AuthCacheDisablePostAuthorize	C		If the Charging Station supports the option for disabling remote authorization for cached invalid idTokens AND has an authorization cache	C-59 and C-49	
	Local start transaction - Authorization first					
					
TC_E_52	DisableRemoteAuthorization	C		If the Charging Station supports the option for disabling remote authorization and The Charging Station supports at least one of the following local start authorization options C-30, C-31, C-32, C-34 and Either Authorization Cache or Local Authorization List is supported.	C-58 and (C-30 or C-31 or C-32 or C-34) and (C-49 or Local Authorization List Management)	Local Authorization - using RFID ISO14443 / RFID ISO15693 / KeyCode / MacAddress & Authorization Cache & Local Authorization List.
					
	Offline Behaviour					

	Store Authorization Data in the Authorization Cache					
TC_E_45	Stop transaction during offline period - Same GroupId	C		For CS: the Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34 and Local Authorization List or Authorization Cache	(C-30 or C-31 or C-32 or C-34) AND (Local Authorization List Management or C-49)	Local Authorization - using RFID ISO14443 / RFID ISO15693 / KeyCode / MacAddress and Local Authorization List or Authorization Cache

Local Authorization List Management

					Related features	
TC Id	OCPP Compliance Testing Tool scenario	Conf. Test for Charging Station	Conf. test for CSMS	Condition / remark	Feature no.	Feature
Offline authorization through local authorization list						
TC_C_21	Accepted	C		The Charging Station must support at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	
TC_C_22	Invalid	C		The Charging Station must support at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	
TC_C_23	Blocked	C		The Charging Station must support at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	
TC_C_24	Expired	C		The Charging Station must support at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	
TC_C_25	Local Authorization List > Authorization Cache	C		The Charging Station must support at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	
Online authorization through local authorization list						
TC_C_27	Accepted	M		<Removed>	<Removed> >	
TC_C_28	Invalid & Not Accepted	M		<Removed>	<Removed> >	

						Related features
TC_C_31	Invalid & Accepted	M		<Removed>	<Removed>	
TC_C_29	Blocked	M		<Removed>	<Removed>	
TC_C_30	Expired	M		<Removed>	<Removed>	
TC_C_58	LocalAuthListDisablePostAuthorize	C		The Charging Station supports the option for disabling remote authorization for invalid idTokens stored at the Local Authorization List.	LA-3	
	Authorization by GroupId					
TC_C_40	Success with Local Authorization List	C	M	For CS: - The Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	
TC_C_43	Invalid status with Local Authorization List	C	M	For CS: - The Charging Station supports at least one of the following local start authorization options: C-30, C-31, C-32, C-34	C-30 or C-31 or C-32 or C-34	

5.8. Page 19 - (2024-09) - TC_E_04_CS Updated condition for test case to exclude it for MacAddress and ISO 15118 PnC

This test case cannot be performed with the local authorization option MacAddress or ISO 15118 PnC.

		Local start transaction - Authorization first					
Old	TC_E_04	Success	C	M	Applicable if one or more of the local start authorization options is implemented.	C-30 - C-35 or ISO 15118 support	Authorization options for local start
New	TC_E_04	Success	C	M	Applicable if one or more of the local start authorization options is implemented.	(C-30 or C-31 or C-32 or C-33 or C-35)	Authorization options for local start

5.9. Page 22 - (2024-09) - TC_E_17_CS Updated condition for test case to correctly specify the applicable TxStopPoint combinations

This testcase allows for a limited set of TxStopPoint combinations, otherwise it is not applicable.

		Local start transaction - Authorization first						
Old	TC_E_17	Deauthorized - EV side disconnect	C	M	- TxStopPoint can either be ReadOnly with a subset of the values or have a valueList of supported values, that contains a subset. This testcase is applicable if the value Authorized or PowerPathClosed is a supported value. - StopTxOnEVSideDisconnect needs to ReadWrite or ReadOnly with value true	(C-10.2 or C-10.3) and C-06.2 and AQ-9	Supported Transaction Stop points	
New	TC_E_17	Deauthorized - EV side disconnect	C	M	This testcase is applicable if the value Authorized is a supported value for TxStopPoint AND EVConnected, PowerPathClosed and EnergyTransfer must not be set as TxStopPoint AND StopTxOnEVSideDisconnect true must be a supported value.	C-10.2 and C-06.2 and AQ-9 and NOT (NOT C-52 AND (10.1 OR C-10.3 OR 10.4))	Supported Transaction Stop points	

5.10. Page 39 - (2025-02) - TC_N_48_CS Made conditional

This test case is only applicable if the Charging Station supports Delta monitoring on the `SecurityCtrlr.BasicAuthPassword` component variable.

Old	TC_N_48	Variable monitoring on write only	M					
New	TC_N_48	Variable monitoring on write only	C		CS: if the CS supports Delta monitoring on the <code>SecurityCtrlr.BasicAuthPassword</code>	AQ-10		

5.11. Page 40 - (2024-11) - TC_H_13_CS Updated invalid condition for test case

The condition should have been reversed.

Old	TC_H_13	Rejected	C		Depending on configuration variable <code>ReservationNonSpecificEVSE</code>	R-2	Support reservations of unspecified EVSE
New	TC_H_13	Rejected	C		Depending on the Charging Station not supporting the configuration variable <code>ReservationNonSpecificEVSE</code>	NOT R-2	Support reservations of unspecified EVSE

5.12. Page 42 - (2025-02) - Removed TC_O_15_CS from certification program

5.13. Test Cases Advanced User Interface

						Related features
Removed	TC_O_15	Language preference of the EV Driver	M			

5.14. Page 44 - (2025-02) - Removed TC_A_13_CS and TC_A_13_CSMS from certification program

Test Cases ISO 15118 Support

						Related features
Removed	TC_A_13	Success - Combined Certificate	C	C	If Combined Charging Station Certificate is supported.	ISO-3 Combined Charging Station Certificate

5.15. Page 48 - (2024-06) - Added additional questions to appendix

Note: This erratum is extended by erratum: [Page 48 - \(2025-02\) - Updated Id's of the additional questions for CSMSs in the appendix](#)

The following additional questions are added for CSMSs:

Id	Additional questions for lab testing
AQ-3	Does your CSMS support Absolute values for the following Charging Profiles:
AQ-3.1	<i>TxDefaultProfile</i>
AQ-3.2	<i>ChargingStationMaxProfile</i>
AQ-4	Does your CSMS support Recurring values for the following Charging Profiles:
AQ-4.1	<i>TxDefaultProfile</i>
AQ-4.2	<i>ChargingStationMaxProfile</i>

5.16. Page 48 - (2025-02) - Updated Id's of the additional questions for CSMSs in the appendix

Note: This erratum extends erratum: [Page 48 - \(2024-06\) - Added additional questions to appendix](#)

Note: This erratum is extended by erratum: [Page 48 - \(2025-04\) - Duplicate AQ-11 id](#)

The following Id's of additional questions for CSMSs have been updated:

Old Id	New Id
AQ-1	AQ-11
AQ-2	AQ-12
AQ-3.1	AQ-13.1
AQ-3.2	AQ-13.2
AQ-4	AQ-14
AQ-4.1	AQ-14.1
AQ-4.2	AQ-14.2
AQ-6	AQ-16

5.17. Page 48 - (2025-04) - Duplicate AQ-11 id

Note: This erratum extends erratum: [Page 48 - \(2025-02\) - Updated Id's of the additional questions for CSMSSs in the appendix](#)

At the 2025-02 errata sheet release, both for Charging Station and CSMS an additional question was added with id AQ-11, but these should be kept unique. The following Id's of additional questions for CSMSSs have been updated:

Old Id	New Id
AQ-11	AQ-17

6. Part 6

6.1. General

6.1.1. Page XX - (2024-11) - All testcases - Updated table structure of all testcases

The table structure of all testcases have been updated. This has been done for multiple reasons:

- It improves readability by providing more space for the main steps.
- It decreases the chance of testcase tables being broken, resulting in missing steps at the bottom of a testcase.
- It makes it easier for the Technical Editors to update testcases.

6.1.2. Page XX - (2025-02) - Renamed OCTT to Test System

In the entire document, the term "OCTT" has been replaced by "Test System".

6.2. Charging Station

6.2.1. Page 4 - (2025-02) - TC_A_01_CS - Updated old identifierString reference in description

Tool validations
* Step 1: [...] - BasicAuthPassword may only contain alpha-numeric characters and the special characters allowed by identifierString. <ins>- BasicAuthPassword may only contain alpha-numeric characters and the special characters allowed by passwordString.</ins>
Post scenario validations:
N/a

6.2.2. Page 7 - (2024-11) - TC_A_05_CS - Successfully reconnecting after every failed connection attempt

It is needed to reconnect after every (intended) failed connection, otherwise the retryBackoffTime may double itself several times, resulting in a very large number that may exceed the configured timeouts.

Table 6. Test Case Id: TC_A_05_CS

Test case name	TLS - server-side certificate - Invalid certificate
Test case Id	TC_A_05_CS
Use case Id(s)	A00
Requirement(s)	A00.FR.309,A00.FR.310,A00.FR.311,A00.FR.412,A00.FR.413,A00.FR.414
System under test	Charging Station
Description	The CSMS uses a server-side certificate to identify itself to the Charging Station, when using security profile 2 or 3.
Purpose	To verify whether the Charging Station is able to terminate the connection when the received server certificate is invalid.
Prerequisite(s)	<ul style="list-style-type: none">- The charging station supports security profile 2 and/or 3- The active NetworkConnectionProfile uses either security profile 2 OR 3. <ins><Removed></ins>

Test case name	TLS - server-side certificate - Invalid certificate	
Before (Preparations)	Configuration State: OCPPCommCtrlr.NetworkConfigurationPriority only contains <Value from ActiveNetworkProfile>	
Memory State:	N/a	
Reusable State(s):	N/a	
Main (Test scenario)	Charging Station	CSMS
	1. The OCTT aborts the connection with the Charging Station.	
	2. The Charging Station initiates a TLS handshake and sends a Client Hello to the OCTT.	3. The OCTT responds with a Server Hello With a <Configured valid server certificate>
	Note(s): - The OCTT will use this as an indication of the time it takes the Charging Station to reconnect.	
	4. The OCTT aborts the connection with the Charging Station.	
	5. The Charging Station initiates a TLS handshake and sends a Client Hello to the OCTT.	6. The OCTT responds with a Server Hello With a <Configured invalid server certificate>
	7. The Charging Station deems the server certificate invalid and terminates the connection.	
	Note : The OCTT will wait two times the measured reconnection time from step 3, before switching the server certificate back to the valid server certificate. The reason for this is that the OCTT is not always able to detect a failed connection attempt.	
	8. The Charging Station initiates a TLS handshake and sends a Client Hello to the OCTT.	9. The OCTT responds with a Server Hello With a <Configured valid server certificate>
	Note(s): - The OCTT will accept the connection to prevent doubling of the RetryBackOffWaitMinimum.	
	10 The Charging Station sends a SecurityEventNotificationRequest	11 The OCTT responds with a SecurityEventNotificationResponse
	Note(s) : - Steps 4 to 11 are repeated per configured invalid server certificate. - In case default certificates are being used, the OCTT will use three different invalid server certificates; "Not signed by installed Root certificate", "Expired", "CommonName that does not equal the FQDN of the server". - In case custom certificates are being used, the OCTT will loop through all certificates configured at the 'CSMS Keystore Invalid'.	
Tool validations	* Step 11: Message: SecurityEventNotificationRequest - type must be <i>InvalidCsmsCertificate</i>	
	Post scenario validations: N/a	

6.2.3. Page 7 - (2025-02) - TC_A_05_CS

Before (Preparations)

Configuration State:
OCPPCommCtrlr.NetworkProfileConnectionAttempts is 3 OCPPCommCtrlr.NetworkConfigurationPriority only contains <Value from ActiveNetworkProfile>
Memory State:
N/a

Main (Test scenario)	
Charging Station	CSMS
[...]	
5. The Charging Station initiates a TLS handshake and sends a Client Hello to the Test System.	6. The Test System responds with a Server Hello With a <Configured invalid server certificate> With a <Generated invalid server certificate>
[...]	
10 The Charging Station sends a SecurityEventNotificationRequest	11 The Test System responds with a SecurityEventNotificationResponse
Note(s):	
- Steps 4 to 11 are repeated per configured invalid server certificate. - In case default certificates are being used, the Test System will use three different invalid server certificates; "Not signed by installed Root certificate", "Expired", "CommonName that does not equal the FQDN of the server". - In case custom certificates are being used, the Test System will loop through all certificates configured at the 'CSMS Keystore Invalid'. The Test System will loop through steps 4 to 11 for a set of generated invalid certificates; "Expired", "Future validity date", "Not signed by installed CSMS Root certificate", "CommonName that does not equal the FQDN of the server", "CommonName containing a wildcard hostname matching the FQDN".	

6.2.4. Page 7 - (2025-04) - TC_A_05_CS - Updated before steps to take into account the AllowCSMSTLSWildcards variable

Test case name	TLS - server-side certificate - Invalid certificate
Test case Id	TC_A_05_CS
...	

Before (Preparations)	
Configuration State:	
OCPPCommCtrlr.NetworkProfileConnectionAttempts is 3 OCPPCommCtrlr.NetworkConfigurationPriority only contains <Value from ActiveNetworkProfile> SecurityCtrlr.AllowCSMSTLSWildcards is false (If implemented)	
Memory State:	
N/a	
Reusable State(s):	
N/a	

6.2.5. Page 20 - (2025-02) - TC_A_23_CS - CSMS returns a CertificateSigned message for each request

TC_A_23_CS: Update Charging Station Certificate by request of CSMS - CertificateSignedRequest Timeout

Test case name	Update Charging Station Certificate by request of CSMS - CertificateSignedRequest Timeout
Test case Id	TC_A_23_CS
...	...
Purpose	To verify if the Charging Station is able to send a new signCertificateRequest when it did not receive a certificateSignedRequest after the configured timeout. CSMS will after a delay send a CertificateSignedRequest for each SignCertificateRequest that it has accepted.
...	...

Before (Preparations)

Main (Test scenario)	
Charging Station	CSMS
...	...
12. The Charging Station responds with a CertificateSignedResponse	11. The Test System sends a CertificateSignedRequest With certificateChain <Certificate generated from the received CSR from step 3 and signed by the provided CSMS Root certificate> certificateType ChargingStationCertificate
14. The Charging Station responds with a CertificateSignedResponse	13. The Test System sends a CertificateSignedRequest With certificateChain <Certificate generated from the received CSR from step 6 and signed by the provided CSMS Root certificate> certificateType ChargingStationCertificate
16. The Charging Station responds with a CertificateSignedResponse	15. The Test System sends a CertificateSignedRequest With certificateChain <Certificate generated from the received CSR from step 9 and signed by the provided CSMS Root certificate> certificateType ChargingStationCertificate

Tool validations	
...	
* Step 12, 14, 16:	
Message: CertificateSignedResponse	
- status must be Accepted or Rejected	
Post scenario validations:	
<i>Note: It does not matter whether Charging Station accepts first or last or all certificates. At least one CertificateSignedResponse must have status Accepted</i>	

6.2.6. Page 22 - (2024-09) - TC_A_19_CS - Fixed references to ConfigurationSlot [O20-4762]

Test case name	Upgrade Charging Station Security Profile - Accepted
Test case Id	TC_A_19_CS

Test case name	Upgrade Charging Station Security Profile - Accepted	
Main (Test scenario)	Charging Station	CSMS
	2. The Charging Station responds with a SetNetworkProfileResponse	<p>1. The OCTT sends a SetNetworkProfileRequest with configurationSlot is <Configured configurationSlot> or <Configured configurationSlot2> depending on which one is already in use</p> <ul style="list-style-type: none"> - connectionData.messageTimeout <Configured messageTimeout> - connectionData.ocppCsmsUrl <Configured ocppCsmsUrl> - connectionData.ocppInterface <Configured ocppInterface> - connectionData.ocppVersion OCPP20 - connectionData.securityProfile <Configured securityProfile + 1>
	4. The Charging Station responds with a SetVariablesResponse	<p>3. The OCTT sends a SetVariablesRequest with variable.name is "NetworkConfigurationPriority" component.name is "OCPPCommCtrlr" attributeValue is "<Configured configurationSlot2>,<Configured configurationSlot>"</p>

Tool validations	<ul style="list-style-type: none"> * Step 2: Message SetNetworkProfileResponse - status Accepted * Step 4: Message SetVariablesResponse - setVariableResult[0].attributeStatus Accepted OR RebootRequired * Step 6: Message ResetResponse - status Accepted * Step 11: Message GetVariablesResponse - getVariableResult[0].attributeValue <Configured securityProfile + 1> * Step 13: Message GetVariablesResponse - getVariableResult[0].attributeValue Does not contain the configurationSlot with the previous (lower) security profile <p>Post scenario validations:</p> <ul style="list-style-type: none"> - N/a 	

6.2.7. Page 24 - (2025-02) - TC_A_20_CS - Testcase did not take into account that the used configuration slot could already be set

The testcase did not take into account that the NetworkConnectionProfile slot updated during the main steps, might already be set at the OCPPCommCtrlr.NetworkConfigurationPriority. This would cause setting the networkConnectionProfile to be rejected already.

Test case name	Upgrade Charging Station Security Profile - No valid CSMSRootCertificate installed
Test case Id	TC_A_20_CS
...	...

Test case name	Upgrade Charging Station Security Profile - No valid CSMSRootCertificate installed
Prerequisite(s)	<ul style="list-style-type: none"> - The Test System connectionData configuration for SUT Charging Station only allows for ip addresses the Test System is able to bind. - The Charging Station supports at least 2 security profiles, one of which is security profile 1. - The Charging Station does not have a valid CSMSRootCertificate installed. - The first Test System connectionData configuration slot must be configured for security profile 1. - The second Test System connectionData configuration slot must be configured for security profile 2 or 3. - The Charging Station is connected using security profile 1. - When starting this testcase the Test System will start another webSocket server for the second connectionData slot.

Before (Preparations)
Configuration State: OCPPCommCtrlr.NetworkConfigurationPriority is <ActiveNetworkProfile slot> (All others are removed)
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetNetworkProfileResponse	<p>1. The Test System sends a SetNetworkProfileRequest with - configurationSlot is <Configured configurationSlot2> or <Configured configurationSlot> (the one currently not used for the active connection)</p> <ul style="list-style-type: none"> - connectionData.messageTimeout <Configured messageTimeout2> - connectionData.ocppCsmsUrl <ocppCsmsUrl that is not currently active> - connectionData.ocppInterface <Configured ocppInterface2> - connectionData.ocppVersion OCPP20 - connectionData.securityProfile <Configured securityProfile2>
4. The Charging Station responds with a SetVariablesResponse	<p>3. The Test System sends a SetVariablesRequest with variable.name is "NetworkConfigurationPriority" component.name is "OCPPCommCtrlr" attributeValue is <configurationSlot set at step 1>,<previous configurationSlot></p>

6.2.8. Page 24 - (2025-04) - TC_A_20_CS - SetNetworkConnectionProfile is allowed to be rejected

Test case name	Upgrade Charging Station Security Profile - No valid CSMSRootCertificate installed
Test case Id	TC_A_20_CS
...	...

Tool validations
* Step 2: Message SetNetworkProfileResponse - status Accepted or Rejected
* Step 4: Message SetVariablesResponse - setVariableResult[0].attributeStatus Rejected
Post scenario validations: - N/a

6.2.9. Page 25 - (2025-02) - TC_A_21_CS

TC_A_21_CS: Upgrade Charging Station Security Profile - No valid ChargingStationCertificate installed

Test case name	Upgrade Charging Station Security Profile - No valid ChargingStationCertificate installed
...	...
Description	The CSMS is able to change the connectionData at the Charging Station. By doing this it is able to upgrade the connection to a security profile 3.
Purpose	To verify if the Charging Station is able to reject upgrading to a security profile 3 when it does not have a valid ChargingStationCertificate installed.
Prerequisite(s)	<ul style="list-style-type: none"> - The Test System connectionData configuration for SUT Charging Station only allows for ip addresses the Test System is able to bind. - The Charging Station support at least 2 security profiles. - The Charging Station does not have a valid ChargingStationCertificate installed. - The Charging Station has a valid CSMSRootCertificate installed. - The second Test System connectionData configuration slot must be configured for security profile 3. - When starting this testcase the Test System will start another webSocket server for the second connectionData slot.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetNetworkProfileResponse	<p>1. The Test System sends a SetNetworkProfileRequest with</p> <ul style="list-style-type: none"> - configurationSlot is <i><Configured configurationSlot></i> or <i><Configured configurationSlot></i> depending on which one is already in use - connectionData.messageTimeout <i><Configured messageTimeout2></i> - connectionData.ocppCsmsUrl <i><ocppCsmsUrl that is not currently active></i> - connectionData.ocppInterface <i><Configured ocppInterface2></i> - connectionData.ocppVersion <i>OCPP20</i> - connectionData.securityProfile <i><Configured securityProfile2></i> 3

6.2.10. Page 26 - (2025-02) - TC_A_22_CS

TC_A_22_CS: Upgrade Charging Station Security Profile - Downgrade security profile - Rejected

Test case name	Upgrade Charging Station Security Profile - Downgrade security profile - Rejected
[...]	
Description	The CSMS is able to change the connectionData at the Charging Station. It tries to downgrade the connection to a lower security profile 1.
Purpose	To verify if the Charging Station is able to reject downgrading to a lower security profile than the currently active security profile 1.
Prerequisite(s)	<ul style="list-style-type: none"> - The Test System connectionData configuration for SUT Charging Station only allows for ip addresses the Test System is able to bind. - The Charging Station supports security profile 2 and/or 3. - The second Test System connectionData configuration slot must be configured for a security profile lower than the first Test System connectionData configuration slot. - The Charging Station has a connection using security profile 2 or 3. - When starting this testcase the Test System will start another webSocket server for the second connectionData slot.

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. The Charging Station responds with a SetNetworkProfileResponse	<p>1. The Test System sends a SetNetworkProfileRequest with:</p> <ul style="list-style-type: none"> - configurationSlot is <Configured configurationSlot> or <Configured configurationSlot> depending on which one is already in use - configurationSlot is <Configured configurationSlot2> or <Configured configurationSlot2> depending on which one is already in use - connectionData.messageTimeout <Configured messageTimeout2> - connectionData.ocppCsmsUrl <ocppCsmsUrl that is not currently active> - connectionData.ocppInterface <Configured ocppInterface2> - connectionData.ocppVersion OCPP20 - connectionData.securityProfile <Configured securityProfile2> - connectionData.securityProfile 1

6.2.11. Page 51 - (2025-02) - TC_B_16_CS - Correctly validate result of reading WriteOnly component variables

Test case name	Get Custom Report - with component criteria
Test case Id	TC_B_16_CS
Use case Id(s)	B08
Requirement(s)	B08.FR.01, B08.FR.03, B08.FR.04, B08.FR.07, B08.FR.09, B089.FR.10, B08.FR.12, B08.FR.13, B08.FR.14 , B06.FR.09
[...]	

Tool validations
[...]
* Step 6:
Message: GetVariablesResponse with:
For component variables where NotifyReportRequest.reportData.variableAttribute.mutability from step 3 is not WriteOnly
- attributeStatus = Accepted
- attributeValue = true
For component variables where NotifyReportRequest.reportData.variableAttribute.mutability from step 3 is WriteOnly
- attributeStatus = Rejected
- attributeValue = <omitted>
[...]

6.2.12. Page 56 - (2024-09) - TC_B_20_CS - Added check on omitting evseld [4390]

Test case name	Reset Charging Station - Without ongoing transaction - OnIdle	
Test case Id	TC_B_20_CSMS	
...		
Main (Test scenario)	Charging Station	CSMS
	...	

Test case name	Reset Charging Station - Without ongoing transaction - OnIdle
Tool validations	<ul style="list-style-type: none"> * Step 1: Message ResetRequest - evselid must be omitted * Step 4: Message BootNotificationResponse - status Accepted
	Post scenario validations: - N/a

6.2.13. Page 57 - (2024-09) - TC_B_21_CS - Added check on omitting evselid [4390]

Test case name	Reset Charging Station - With Ongoing Transaction - OnIdle	
Test case Id	TC_B_21_CSMS	
...		
Main (Test scenario)	Charging Station	CSMS
...		
Tool validations	<ul style="list-style-type: none"> * Step 1: Message ResetRequest - type OnIdle - evselid must be omitted * Step 8: Message BootNotificationResponse - status Accepted 	
	Post scenario validations: - N/a	

6.2.14. Page 58 - (2024-09) - TC_B_22_CS - Added check on omitting evselid [4390]

Test case name	Reset Charging Station - With Ongoing Transaction - Immediate	
Test case Id	TC_B_22_CSMS	
...		
Main (Test scenario)	Charging Station	CSMS
...		
Tool validations	<ul style="list-style-type: none"> * Step 1: Message ResetRequest - type Immediate - evselid is omitted * Step 6: Message BootNotificationResponse - status Accepted 	
	Post scenario validations: - N/a	

6.2.15. Page 63 - (2025-02) - Changed reset to Immediate

The reset in step #1 has been changed to "Immediate", because having a reservation pending is not an "idle" situation. As a result a Charging Station would respond with status "Scheduled" instead of resetting.

Test case name	Reset Charging Station - Reserved persists reset	
Test case Id	TC_B_24_CS	
...		

Main (Test scenario)		
Charging Station	CSMS	
2. The Charging Station responds with a ResetResponse		1. The Test System sends a ResetRequest with type Immediate
...		

6.2.16. Page 85 - (2025-04) - TC_B_53_CS - Check if all required values are provided

The testcase currently does not explicitly describe that the value fields for standardized component.variables should not be omitted (with EVSE.Power as an exception. For this only the maxLimit is required). These are optional at the JSON schema, because they are allowed to omit in case of WriteOnly variables.

Test case name	Get Base Report - Test mandatory DM variables via FullInventory	
Test case Id	TC_B_53_CS	
...		
Main (Test scenario)	Charging Station	CSMS

Tool validations	<p>...</p> <p>Post scenario validations: The Test System checks that all implemented standardized components / variables are implemented correctly according to the OCPP specification:</p> <ul style="list-style-type: none"> - The components / variables that are required according to the OCPP specification are implemented. - For each component/variable, where variableCharacteristics.dataType is set to OptionList, SequenceList or MemberList, the variableCharacteristics.valuesList is not omitted or empty. - For each component/variable, where variableCharacteristics.dataType is OptionList, SequenceList or MemberList, the variableAttribute.value is allowed based on the values in the provided variableCharacteristics.valuesList. - For variables with mutability set to WriteOnly the variableAttribute.value is omitted in the NotifyReportRequest. - For variables with mutability NOT set to WriteOnly the variableAttribute.value is NOT omitted in the NotifyReportRequest. There is one exception to this rule and that is for EVSE.Power. This variable only has a maxLimit. 	

6.2.17. Page 89 - (2025-02) - Add setting of NetworkProfileConnectionAttempts

TC_B_57_CS: Network Reconnection - After connection loss

Test case name	Network Reconnection - After connection loss	
Test case Id	TC_B_57_CS	
...	...	

Before (Preparations)		
Configuration State:		
OCPPCommCtrlr.NetworkProfileConnectionAttempts	is 3	
OCPPCommCtrlr.RetryBackOffRepeatTimes	is 2	
OCPPCommCtrlr.RetryBackOffRandomRange	is 0	
OCPPCommCtrlr.RetryBackOffWaitMinimum	is <Configured RetryBackOffWaitMinimum>	

Before (Preparations)
Memory State: N/a
Reusable State(s): N/a

6.2.18. Page 117/141 - (2025-02) - Added remote support for Authorization Cache & Local Authorization List testcases

The CWG decided that online local authorization list and authorization cache testcase can also be tested with remote authorization.

...	...
Test case Id	TC_C_08_CS, TC_C_09_CS, TC_C_10_CS, TC_C_11_CS, TC_C_12_CS, TC_C_13_CS, TC_C_33_CS, TC_C_34_CS, TC_C_36_CS, TC_C_37_CS, TC_C_38_CS
...	...

...	...
Test case Id	TC_C_57_CS
...	...

...	...
Test case Id	TC_C_27_CS, TC_C_28_CS, TC_C_29_CS, TC_C_30_CS, TC_C_31_CS
...	...

...	...
Test case Id	TC_C_58_CS
...	...

...	...
Test case Id	TC_C_32_CS
...	...

Main (Test scenario)
Main steps have been updated to use the Reusable State Authorized based on the configured scenario Local or Remote .
...

6.2.19. Page 101 - (2025-04) - TC_C_14_CS - Fixing invalid component variable reference

Test case name	Authorization through authorization cache - GroupID equal to MasterPassGroupId.
...	...
Prerequisite(s)	<ul style="list-style-type: none"> - AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - The Charging station supports MasterPass feature. - The Charging Station supports authorization methods other than NoAuthorization

6.2.20. Page 102 - (2025-04) - TC_C_15_CS - Fixing invalid component variable reference

Test case name	Authorization through authorization cache - StopTxOnInvalidId = false, MaxEnergyOnInvalidId > 0
...	...
Prerequisite(s)	<ul style="list-style-type: none"> - AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - The Charging Station has MaxEnergyOnInvalidId implemented - At least one of the following must be supported; Local auth list, auth cache, StartTxUnknownIds. - The Charging Station supports authorization methods other than NoAuthorization

6.2.21. Page 104 - (2025-04) - TC_C_16_CS - Fixing invalid component variable reference

Test case name	Authorization through authorization cache - StopTxOnInvalidId = true
...	...
Prerequisite(s)	<ul style="list-style-type: none"> - AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - At least one of the following must be supported; Local auth list, auth cache, StartTxUnknownIds. - The Charging Station supports authorization methods other than NoAuthorization

6.2.22. Page 105 - (2025-04) - TC_C_17_CS - Fixing invalid component variable reference

Test case name	Authorization through authorization cache - StopTxOnInvalidId = false
...	...
Prerequisite(s)	<ul style="list-style-type: none"> - AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - At least one of the following must be supported; Local auth list, auth cache, StartTxUnknownIds. - The Charging Station supports authorization methods other than NoAuthorization

6.2.23. Page 108 - (2025-04) - TC_C_18_CS - Fixing invalid component variable reference

Test case name	Authorization through authorization cache - StopTxOnInvalidId = true, MaxEnergyOnInvalidId > 0
...	...
Prerequisite(s)	<ul style="list-style-type: none"> - AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - The Charging Station has MaxEnergyOnInvalidId implemented. - At least one of the following must be supported; Local auth list, auth cache, StartTxUnknownIds. - The Charging Station supports authorization methods other than NoAuthorization

6.2.24. Page 120 - (2025-04) - TC_C_34_CS - Making use of the Deauthorized reusable state

Test case name	Store Authorization Data in the Authorization Cache - Update on TransactionResponse
Test case Id	TC_C_34_CS
...	

Before (Preparations)
Configuration State: AuthCacheEnabled is true (If implemented) LocalPreAuthorize is true LocalAuthListEnabled is true StopTxOnInvalidId is true MaxEnergyOnInvalidId is 0
Memory State: <i>IdTokenCached</i> for <Configured valid IdToken fields>
Reusable State(s): N/a

Main (Test scenario)
Charging Station CSMS
1. Execute Reusable State <i>Authorized</i> (Cached idToken, idTokenInfo.status invalid)
2. Execute Reusable State Deauthorized
3. Execute Reusable State <i>EVDisconnected</i>
4. Execute Reusable State <i>ParkingBayUnoccupied</i>
5. Execute Reusable State <i>ParkingBayOccupied</i>
6. Execute Reusable State <i>Authorized</i> (idTokenInfo.status invalid)

6.2.25. Page 128 - (2025-04) - TC_C_41_CS - Fixing invalid component variable reference

Test case name	Authorization by GroupId - Success with Authorization Cache
...	...
Prerequisite(s)	- AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - The Charging Station supports authorization methods other than NoAuthorization

6.2.26. Page 134 - (2025-04) - TC_C_44_CS - Fixing invalid component variable reference

Test case name	Authorization by GroupId - Invalid status with Authorization Cache
...	...
Prerequisite(s)	- AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - The Charging Station supports authorization methods other than NoAuthorization

6.2.27. Page 137 - (2025-04) - TC_C_46_CS - Fixing invalid component variable reference

Test case name	Store Authorization Data in the Authorization Cache - AuthCacheLifeTime
...	...
Prerequisite(s)	- AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - Configuration variable AuthCacheLifeTime is implemented

6.2.28. Page 138 - (2024-09) - TC_C_47_CS - StoppedReason must be validated in Ended event [O20-4467]

Test case name	Stop Transaction with a Master Pass - With UI - All transactions	
Test case Id	TC_C_47_CS	
...		
Main (Test scenario)	Charging Station	CSMS

Tool validations	<ul style="list-style-type: none"> * Step 1: Message AuthorizeRequest - idToken.idToken <Configured masterpass_idtoken_idtoken> - idToken.type <Configured masterpass_idtoken_type> * Step 3: Message TransactionEventRequest - transactionInfo.stoppedReason MasterPass (in last TransactionEventRequest) - idToken omit or - idToken.idToken <Configured masterpass_idtoken_idtoken> and - idToken.type <Configured masterpass_idtoken_type> (once per stopped transaction) - eventType Ended (in last TransactionEventRequest) <p>Post scenario validations:</p> <ul style="list-style-type: none"> - N/a 	

6.2.29. Page 145 - (2025-04) - TC_C_25_CS - Fixing invalid component variable reference

Test case name	Offline authorization through local authorization list - Local Authorization List > Authorization Cache
...	...
Prerequisite(s)	<ul style="list-style-type: none"> - LocalAuthListCtrlr.LocalAuthListAvailable is implemented with value true - AuthCacheCtrlr.AuthCacheAvailable AuthCacheCtrlr.Available is implemented with value true - OfflineTxForUnknownIdEnabled is implemented. - The Charging Station supports authorization methods other than NoAuthorization

6.2.30. Page 146 - (2024-11) - TC_C_26_CS - Allow StatusNotification status = Occupied

The connector status change should be reported after the connection is restored. The Charging Station should be allowed to report StatusNotificationRequest status = Occupied (or NotifyEventRequest).

Test case name	Set Variable Monitoring - Periodic event	
Test case Id	TC_C_26_CS	
Use case Id(s)	C15 & C13	
Requirement(s)	C15.FR.02,C15.FR.06,C15.FR.08,C13.FR.04	
...		
Main (Test scenario)	Charging Station	CSMS

	1. The Charging Station notifies the CSMS about the current state of all connectors.	2. The OCTT responds accordingly.

	3. Execute Reusable State StopAuthorized	
	4. Execute Reusable State EVConnectedPostSession	
	5. Execute Reusable State EVDDisconnected	

Test case name	Set Variable Monitoring - Periodic event
Tool validations	<ul style="list-style-type: none"> * Step 1: <p>Message: StatusNotificationRequest</p> <ul style="list-style-type: none"> - connectorStatus must be <i>Occupied</i> <p>Message: NotifyEventRequest</p> <ul style="list-style-type: none"> - eventData[0].trigger must be <i>Delta</i> - eventData[0].actualValue must be <i>Occupied</i> - eventData[0].component.name must be <i>Connector</i> - eventData[0].variable.name must be <i>AvailabilityState</i> ...
	Post scenario validations: N/A

6.2.31. Page 147-153 - (2025-04) - TC_C_50_CS, TC_C_51_CS, TC_C_52_CS, TC_C_53_CS, TC_C_54_CS, TC_C_55_CS - Always re-install V2G certificates

Note: This erratum revises erratum: [Page 147 - \(2025-02\) - TC_C_50_CS](#) and the related errata below

It is not possible to detect whether all expected certificates are installed, when reusing the SubCAs. Therefore the Test System always re-installs the V2G certificates.

Before (Preparations)
Memory State: <i>CertificateInstalled</i> for certificateType <i>V2GRootCertificate</i> <i>CertificateInstalled</i> for certificateType <i>MORootCertificate</i>
<i>RenewV2GChargingStationCertificate</i> (If none are present, when checking with <i>GetInstalledCertificateIds.certificateType = V2GCertificateChain</i>)

6.2.32. Page 147 - (2025-02) - TC_C_50_CS

Note: This erratum and the related errata below are revised by erratum: [Page 147-153 - \(2025-04\) - TC_C_50_CS, TC_C_51_CS, TC_C_52_CS, TC_C_53_CS, TC_C_54_CS, TC_C_55_CS - Always re-install V2G certificates](#)

Before (Preparations)
Configuration State: TxCtrlr.TxStartPoint contains one or more of <i>PowerPathClosed, Authorized, EVConnected, ParkingBayOccupancy</i> AuthCtrlr.AuthEnabled is <i>true</i> (If implemented AND ReadWrite) AuthCtrlr.DisableRemoteAuthorization is <i>false</i> (If implemented) For the ISO15118Ctrlr of the EVSE used in the PnC transaction: ISO15118Ctrlr.CentralContractValidationAllowed is <i>false</i> ISO15118Ctrlr.ContractCertificateInstallationEnabled is <i>true</i> ISO15118Ctrlr.V2GCertificateInstallationEnabled is <i>true</i> ISO15118Ctrlr.PnCEnabled is <i>true</i> ISO15118Ctrlr.Seccid is configured <i>seccid</i> ISO15118Ctrlr.CountryName is <i>NL</i> ISO15118Ctrlr.OrganizationName is configured <i>vendorId</i>
Memory State: <i>CertificateInstalled</i> for certificateType <i>V2GRootCertificate</i> <i>CertificateInstalled</i> for certificateType <i>MORootCertificate</i> <i>RenewV2GChargingStationCertificate</i> (If none are present, when checking with <i>GetInstalledCertificateIds.certificateType = V2GCertificateChain</i>)
Reusable State(s): State is <i>EVConnectedPreSession</i>

Main (Test scenario)	
Charging Station	CSMS
[...]	
3. The Charging Station sends a TransactionEventRequest	4. The Test System responds with a TransactionEventResponse With idTokenInfo.status Accepted
<u>Note(s):</u> - This step needs to be executed when TxStartPoint contains Authorized OR the transaction already started. So in the case TxStartPoint contains ParkingBayOccupancy or (EVConnected , in the case this test case was initiated from state EVConnectedPreSession)	
5. Execute Reusable State EnergyTransferStarted	

6.2.33. Page 148 - (2025-02) - TC_C_51_CS

Before (Preparations)
Configuration State: TxCtrlr.TxStartPoint contains one or more of PowerPathClosed, Authorized, EVConnected, ParkingBayOccupancy AuthCtrlr.AuthEnabled is true (If implemented AND ReadWrite) AuthCtrlr.DisableRemoteAuthorization is false (If implemented) For the ISO15118Ctrlr of the EVSE used in the PnC transaction: ISO15118Ctrlr.CentralContractValidationAllowed is false ISO15118Ctrlr.PnCEnabled is true
Memory State: CertificateInstalled for certificateType V2GRootCertificate CertificateInstalled for certificateType MORootCertificate RenewV2GChargingStationCertificate (If none are present, when checking with GetInstalledCertificateIds.certificateType = V2GCertificateChain)
Reusable State(s): State is EVConnectedPreSession

6.2.34. Page 149 - (2025-02) - TC_C_52_CS

TC_C_52_CS: Authorization using Contract Certificates 15118 - Online - Central contract certificate validation - Accepted

Test case name	Authorization using Contract Certificates 15118 - Online - Central contract certificate validation - Accepted
Test case Id	TC_C_52_CS
Use case Id(s)	C07
Requirement(s)	C07.FR.01,C07.FR.02,C07.FR.06
System under test	Charging Station
Description	The Charging Station is able to authorize with contract certificates when it supports ISO 15118.
Purpose	To verify if the Charging Station is able to authorize, while not being able to locally validate the contract certificate and then send it to the CSMS.
Prerequisite(s)	- The V2G/MO Root certificate that is needed to validate the EV Contract certificate must NOT be installed at the Charging Station. - The Charging Station supports central contract validation.

Before (Preparations)
Configuration State:
TxCtrlr.TxStartPoint contains one or more of <i>PowerPathClosed, Authorized, EVConnected, ParkingBayOccupancy</i>
AuthCtrlr.AuthEnabled is <i>true</i> (If implemented AND ReadWrite)
AuthCtrlr.DisableRemoteAuthorization is <i>false</i> (If implemented)
For the ISO15118Ctrlr of the EVSE used for the PnC transaction:
ISO15118Ctrlr.CentralContractValidationAllowed is <i>true</i>
ISO15118Ctrlr.ContractCertificateInstallationEnabled is <i>true</i>
ISO15118Ctrlr.V2GCertificateInstallationEnabled is <i>true</i>
ISO15118Ctrlr.PnCEnabled is <i>true</i>
ISO15118Ctrlr.SeccId is configured <i>seccId</i>
ISO15118Ctrlr.CountryName is <i>NL</i>
ISO15118Ctrlr.OrganizationName is configured <i>vendorId</i>
.
Memory State:
N/a - CertificateInstalled for certificateType <i>V2GRootCertificate</i>
RenewV2GChargingStationCertificate (If none are present, when checking with GetInstalledCertificateIds.certificateType = <i>V2GCertificateChain</i>)
Reusable State(s):
State is <i>EVConnectedPreSession</i>

Main (Test scenario)	
Charging Station	CSMS
[...]	
3. The Charging Station sends a TransactionEventRequest	4. The Test System responds with a TransactionEventResponse With idTokenInfo.status Accepted
Note(s): This step needs to be executed when TxStartPoint contains Authorized OR the transaction already started. So in the case TxStartPoint contains ParkingBayOccupancy or (EVConnected, in the case this testcase was initiated from state EVConnectedPreSession.)	
5. Execute Reusable State <i>EnergyTransferStarted</i>	

6.2.35. Page 150 - (2025-02) - TC_C_53_CS

TC_C_53_CS: Authorization using Contract Certificates 15118 - Online - Central contract validation fails

Test case name	Authorization using Contract Certificates 15118 - Online - Central contract validation fails
Test case Id	TC_C_53_CS
Use case Id(s)	C07
Requirement(s)	N/a
System under test	Charging Station
Description	The Charging Station is able to authorize with contract certificates when it supports ISO 15118.
Purpose	To verify if the Charging Station is able to handle an invalid contract certificate.
Prerequisite(s)	- The V2G/MO Root certificate that is needed to validate the EV Contract certificate must NOT be installed at the Charging Station. - The Charging Station supports central contract validation.

Before (Preparations)
Configuration State:
TxCtrlr.TxStartPoint contains one or more of <i>PowerPathClosed, Authorized, EVConnected, ParkingBayOccupancy</i>
AuthCtrlr.AuthEnabled is <i>true</i> (If implemented AND ReadWrite)
AuthCtrlr.DisableRemoteAuthorization is <i>false</i> (If implemented)
For the ISO15118Ctrlr of the EVSE involved in the PnC transaction:
ISO15118Ctrlr.CentralContractValidationAllowed is <i>true</i>
ISO15118Ctrlr.PnCEnabled is <i>true</i>
Memory State:
N/a - CertificateInstalled for certificateType <i>V2GRootCertificate</i>
RenewV2GChargingStationCertificate (If none are present, when checking with GetInstalledCertificateIds.certificateType = <i>V2GCertificateChain</i>)
Reusable State(s):
State is <i>EVConnectedPreSession</i>

6.2.36. Page 151 - (2024-09) - TC_C_54_CS - removed reusable state IdTokenCached [O20-3510]

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is true			
Test case Id	TC_C_54_CS			
...				
Before (Preparations)	Configuration State: ...			
Memory State:				
<i>CertificateInstalled</i> for certificateType <i>V2GRootCertificate</i> <i>CertificateInstalled</i> for certificateType <i>MORootCertificate</i> <i>IdTokenCached</i> for <Configured valid IdToken fields> (If implemented) <i>IdTokenLocalAuthList</i> for <Configured valid IdToken fields> (If implemented)				
Reusable State(s):				
N/a				
Main (Test scenario)	Charging Station	CSMS		
...				
...				

6.2.37. Page 151 - (2025-02) - TC_C_54_CS

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is true	
[...]		
Description	The Charging Station is able to authorize with contract certificates when it supports ISO 15118- for an EMAID that exists in authorization cache or local authorization list, while offline.	
[...]		

Before (Preparations)
[...]
Memory State:
<i>CertificateInstalled</i> for certificateType <i>V2GRootCertificate</i> <i>CertificateInstalled</i> for certificateType <i>MORootCertificate</i> <i>RenewV2GChargingStationCertificate</i> (If none are present, when checking with GetInstalledCertificateIds.certificateType = <i>V2GCertificateChain</i>) <i>IdTokenCached</i> for <Configured valid IdToken fields> (If implemented) <i>IdTokenLocalAuthList</i> for <Configured valid IdToken fields> (If implemented)
Reusable State(s):
N/a

Main (Test scenario)	
Charging Station	CSMS
[...]	
5. The Charging Station sends a TransactionEventRequest <u>Note(s):</u> - This step needs to be executed when TxStartPoint contains EVConnected OR the transaction already started. So in the case TxStartPoint contains ParkingBayOccupancy (in the case TxStartPoint contains ParkingBayOccupancy_)	6. The Test System responds with a TransactionEventResponse
7. The Charging Station sends a TransactionEventRequest <u>Note(s):</u> - This step needs to be executed when TxStartPoint contains Authorized OR the transaction already started. So in the case TxStartPoint contains ParkingBayOccupancy or EVConnected. .	8. The Test System responds with a TransactionEventResponse With idTokenInfo.status Accepted
9. Execute Reusable State <i>EnergyTransferStarted</i>	

6.2.38. Page 153 - (2024-09) - TC_C_55_CS - removed reusable state IdTokenCached [O20-3510]

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is false			
Test case Id	TC_C_55_CS			
...				
Before (Preparations)	Configuration State: ...			
Memory State:				
<i>CertificateInstalled</i> for certificateType V2GRootCertificate <i>CertificateInstalled</i> for certificateType M0RootCertificate <i>IdTokenCached</i> for <Configured valid IdToken fields> (If implemented) <i>IdTokenLocalAuthList</i> for <Configured valid IdToken fields> (If implemented)				
Reusable State(s):	N/a			
Main (Test scenario)	Charging Station	CSMS		
...				
...				

6.2.39. Page 153 - (2025-02) - TC_C_55_CS

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is false	
[...]		
Description	The Charging Station is able to authorize with contract certificates when it supports ISO 15118. The Charging Station will not authorize with contract certificates when offline.	
[...]		

Before (Preparations)
Configuration State:
TxCtrlr.TxStartPoint contains one or more of <i>PowerPathClosed, Authorized, EVConnected, ParkingBayOccupancy</i>
AuthCtrlr.AuthEnabled is true (If implemented AND ReadWrite)
AuthCacheCtrlr.Enabled is true OR LocalAuthListCtrlr.Enabled is true
OfflineTxForUnknownIdEnabled is true (If implemented)
OfflineTxForUnknownIdEnabled is false (If implemented)
OfflineThreshold is <Configured RetryBackOffWaitMinimum_duration> + 60.0
RetryBackOffWaitMinimum is <Configured RetryBackOffWaitMinimum_duration>
RetryBackOffRandomRange is 0
Note:
<Configured RetryBackOffWaitMinimum_duration should be long enough to execute manual tasks>
For the ISO15118Ctrlr of the EVSE involved in the PnC transaction:
ISO15118Ctrlr.ContractValidationOffline is false
ISO15118Ctrlr.PnCEnabled is true
Memory State:
CertificateInstalled for certificateType <i>V2GRootCertificate</i>
CertificateInstalled for certificateType <i>M0RootCertificate</i>
RenewV2GChargingStationCertificate (If none are present, when checking with GetInstalledCertificateIds.certificateType = <i>V2GCertificateChain</i>)
IdTokenCached for <Configured valid IdToken fields> (If implemented)
IdTokenLocalAuthList for <Configured valid IdToken fields> (If implemented)
Reusable State(s):
N/a

Main (Test scenario)	
Charging Station	CSMS
[...]	
5. The Charging Station sends a TransactionEventRequest	6. The Test System responds with a TransactionEventResponse
<p>Note(s):</p> <ul style="list-style-type: none"> - This step needs to be executed when TxStartPoint contains <i>EVConnected OR the transaction already started - So in the case TxStartPoint contains ParkingBayOccupancy (in the case TxStartPoint contains ParkingBayOccupancy_)</i> 	
[...]	

6.2.40. Page 165/169 - (2025-02) -

TC_E_01_CS/TC_E_09_CS/TC_E_10_CS/TC_E_12_CS/TC_E_13_CS - Extended the testcases until the Charging Station start charging

This is needed to test the TransactionEventRequest with eventType Updated messages, while specific TxStartPoints are configured.

Test case name	Start transaction options - PowerPathClosed
Test case Id	TC_E_01_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>EVConnectedPreSession</i>	
2. Execute Reusable State <i>EnergyTransferStarted</i>	

Test case name	Start transaction options - EVConnected
Test case Id	TC_E_09_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>EVConnectedPreSession</i>	
2. Execute Reusable State <i>Authorized</i>	
3. Execute Reusable State <i>EnergyTransferStarted</i>	

Test case name	Start transaction options - Authorized - Local
Test case Id	TC_E_10_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>Authorized (Local)</i>	
2. Execute Reusable State <i>EVConnectedPreSession</i>	
3. Execute Reusable State <i>EnergyTransferStarted</i>	

Test case name	Start transaction options - ParkingBayOccupied
Test case Id	TC_E_12_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>ParkingBayOccupied</i>	
2. Execute Reusable State <i>Authorized</i>	
3. Execute Reusable State <i>EVConnectedPreSession</i>	
4. Execute Reusable State <i>EnergyTransferStarted</i>	

Test case name	Start transaction options - Authorized - Remote
Test case Id	TC_E_13_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>Authorized (Remote)</i>	
2. Execute Reusable State <i>EVConnectedPreSession</i>	
3. Execute Reusable State <i>EnergyTransferStarted</i>	

6.2.41. Page 174 - (2025-04) - TC_E_17_CS - Aligned configuration before steps with updated prerequisites

Note: This erratum extends erratum: [Page 174 - \(2024-09\) - TC_E_17_CS - Updated prerequisite for test case to correctly specify the applicable TxStopPoint combinations](#)

The prerequisite was updated, but the before configuration steps were not aligned with this change.

Test case name	Stop transaction options - Deauthorized - EV side disconnect
Test case Id	TC_E_17_CS
...	...

Before (Preparations)

Configuration State:

If the mutability of **TxStopPoint** is **ReadWrite** then **TxStopPoint** contains **PowerPathClosed AND/OR Authorized**
StopTxOnEVSideDisconnect is **true**
UnlockOnEVSideDisconnect is **false**
AuthCtrlr.AuthEnabled is **true** (If implemented AND **ReadWrite**)
AuthCtrlr.DisableRemoteAuthorization is **false** (If implemented)

6.2.42. Page 174 - (2024-09) - TC_E_17_CS - Updated prerequisite for test case to correctly specify the applicable TxStopPoint combinations

Note: This erratum is extended by erratum: [Page 174 - \(2025-04\) - TC_E_17_CS - Aligned configuration before steps with updated prerequisites](#)

This testcase allows for a limited set of TxStopPoint combinations, otherwise it is not applicable.

Test case name	Stop transaction options - Deauthorized - EV side disconnect
Test case Id	TC_E_17_CS
...	
Old: Prerequisite(s)	<ul style="list-style-type: none">- The Charging Station does NOT have the following configuration; The mutability of TxStopPoint is ReadOnly AND the value Authorized OR PowerPathClosed is NOT set OR (EnergyTransfer OR DataSigned OR EVConnected is set).- If the mutability of TxStopPoint is _ReadWrite, then the value Authorized OR PowerPathClosed must be supported.
New: Prerequisite(s)	This testcase is applicable if the value Authorized is a supported value for TxStopPoint AND EVConnected , PowerPathClosed and EnergyTransfer must not be set as TxStopPoint AND StopTxOnEVSideDisconnect true must be a supported value.

6.2.43. Page 176 - (2024-11) - TC_E_39_CS - Missing StatusNotificationRequest/NotifyEventRequest

Test case name	Stop transaction options - Deauthorized - timeout	
Test case Id	TC_E_39_CS	
...		
Main (Test scenario)	Charging Station	CSMS

	Manual Action: Connect the EV and EVSE on EV side.	
	Manual Action: Connect the EV and EVSE on EVSE side.	
	3. The Charging Station notifies the CSMS about the status change of the connector.	4. The OCTT responds accordingly.

Test case name	Stop transaction options - Deauthorized - timeout
Tool validations	<ul style="list-style-type: none"> * Step 1: Message: TransactionEventRequest - triggerReason must be <i>EVConnectTimeout</i> - eventType must be <i>Updated</i> if TxStartPoint is <i>ParkingBayOccupancy</i>, else <i>Ended</i> - transactionInfo.stoppedReason must be <i>Timeout</i> * Step 3: Message: StatusNotificationRequest - connectorStatus must be <i>Occupied</i> Message: NotifyEventRequest - eventData[0].trigger must be <i>Delta</i> - eventData[0].actualValue must be <i>Occupied</i> - eventData[0].component.name must be <i>Connector</i> - eventData[0].variable.name must be <i>AvailabilityState</i> * Step 5: Message: TransactionEventRequest - triggerReason can only be <i>CablePluggedIn</i> - transactionInfo.chargingState should not be <i>Charging</i> - eventType must be <i>Updated</i> if TxStartPoint is <i>ParkingBayOccupancy</i>, else <i>Ended</i> <p>Post scenario validations: N/a</p>

6.2.44. Page 182 - (2025-04) - TC_E_52_CS - Testcase is not able to determine the authorization is refused in case TxStartPoint is not Authorized

The testcase is not able to verify the authorization is refused when the TxStartPoint is not Authorized. Therefore the testcase is extended to also plugin the cable to determine whether the authorization was actually refused.

Test case name	Local start transaction - Authorization first - DisableRemoteAuthorization	
Test case Id	TC_E_52_CS	
...		

Main (Test scenario)	
Charging Station	CSMS
<u>Manual Action:</u> Present an idToken which is not configured in the Local Authorization List nor present in Authorization Cache.	
1. The Charging Station does NOT send a AuthorizeRequest	
2. Execute Reusable State <i>EVConnectedPreSession</i>	
3. The Charging Station does NOT start charging.	

6.2.45. Page 185 - (2024-09) - TC_E_35_CS - StoppedReason must be validated in Ended event [O20-4467]

Test case name	Stop transaction options - PowerPathClosed - Remote stop	
Test case Id	TC_E_35_CS	
...		
Main (Test scenario)	Charging Station	CSMS

Test case name	Stop transaction options - PowerPathClosed - Remote stop
Tool validations	<ul style="list-style-type: none"> * Step 2: Message: RequestStopTransactionResponse - status must be Accepted * Step 3: Message: TransactionEventRequest - triggerReason must be <i>RemoteStop</i> (for one of the TransactionEventRequests) - transactionInfo.stoppedReason must be <i>Remote</i> (for the last TransactionEventRequest) - eventType must be <i>Ended</i> (for the last TransactionEventRequest) <p>Post scenario validations: N/a</p>

6.2.46. Page 188 - (2025-02) - TC_E_22_CS - Stop transaction options - EnergyTransfer stopped - will end transaction

When TxStopPoint is EnergyTransfer this will end transaction when EV suspends energy.

TC_E_22_CS: Stop transaction options - EnergyTransfer stopped - SuspendedEV

Test case name	Stop transaction options - EnergyTransfer stopped - SuspendedEV
Test case Id	TC_E_22_CS
..	..

Tool validations
<ul style="list-style-type: none"> * Step 1: Message: TransactionEventRequest - triggerReason must be <i>ChargingStateChanged</i> - transactionInfo.chargingState must be <i>EVConnected</i> OR - transactionInfo.chargingState must be <i>SuspendedEV</i> AND - transactionInfo.stoppedReason must be <i>StoppedByEV</i> - eventType must be <i>Ended</i> (if chargingState is EVConnected) OR - eventType must be <i>Updated</i> (if chargingState is SuspendedEV)
Post scenario validations:
N/a

6.2.47. Page 199 - (2025-04) - TC_E_27_CS - Remove manual action between step 4/5

There seems to have been copy/paste errors between testcase TC_E_26_CS and TC_E_27_CS. The cable should not have been reconnected between these steps as we are trying to trigger the timeout.

Test case name	Disconnect cable on EV-side - Suspend transaction - Fixed cable connection timeout
Test case Id	TC_E_27_CS
...	

Main (Test scenario)	
Charging Station	CSMS
<u>Manual Action:</u> Disconnect the EV and EVSE on EV side (EVSE loses connection with EV).	
1. The Charging Station sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse

Main (Test scenario)	
3. The Charging Station notifies the CSMS about the current state of the connector.	4. The Test System responds accordingly.
<u>Note(s):</u> This step needs to be executed when the Charging Station has a permanently attached cable on the Charging Station side.	
<u>Manual Action:</u> Reconnect the EV and EVSE on EV side.	
<u>Note(s):</u> If the Charging Station has a permanently attached cable on the Charging Station side, then this step needs to be executed before the configured EVConnectionTimeout expires.	
5. The Charging Station sends a TransactionEventRequest	6. The Test System responds with a TransactionEventResponse
<u>Note(s):</u> Optionally the Charging Station can send a StatusNotificationRequest or NotifyEventRequest with status Available	

6.2.48. Page 204 - (2025-02) - TC_E_31_CS - Add steps for when running the testcase in Remote mode

As the Charging Station will go offline, the transaction must be stopped from EV side in case of remote authorization.

TC_E_31_CS: Check Transaction status - Transaction with id ended - with message in queue

Test case name	Check Transaction status - Transaction with id ended - with message in queue
Test case Id	TC_E_31_CS
...	...

Before (Preparations)
Configuration State:
...
UnlockOnEVSideDisconnect is true (If ReadWrite)
...

Main (Test scenario)	
Charging Station	CSMS
The OCTT closes the WebSocket connection AND does not accept a reconnect.	
<u>Manual Action:</u> Present the same idToken as used to start the transaction.	
<u>Notes(s):</u> Only if configured scenario is local	
Manual Action: Stop the energy transfer via the EV.	
Notes(s): Only if configured scenario is remote	
Manual Action: Disconnect the EV and EVSE.	
Manual Action: Drive EV out of parking bay. (Only needed if TxStopPoint is ParkingBayOccupancy)	
...	

Tool validations
...
* Step 3:
Message: TransactionEventRequest
The tool validations from the reusable states need to be used to verify whether all required TransactionEventRequests have been received.
From StopAuthorized through ParkingBayUnoccupied (in case of scenario Local).
And from EnergyTransferSuspended through ParkingBayUnoccupied (in case of scenario Remote).

Tool validations
Post scenario validations: N/a

6.2.49. Page 214 - (2024-06) TC_E_43_CS Move reusable state TransactionEventsInQueueEnded to Before [768]

State TransactionEventsInQueueEnded is moved to Before stage.

Test Case Id: TC_E_43_CS

Test case name	Offline Behaviour - Transaction during offline period	
Test case Id	TC_E_43_CS	
Use case Id(s)	E12	
Requirement(s)	E12.FR.01,E12.FR.02,E12.FR.06	
System under test	Charging Station	
Description	The Charging Station queues TransactionEvent messages to inform the CSMS that a transaction occurred while the Charging Station was Offline.	
Purpose	To verify if the Charging Station is able to queue TransactionEvent messages while it was offline.	
Prerequisite(s)	The Charging Station supports authorization methods other than NoAuthorization	
Before (Preparations)	Configuration State: N/a	
	Memory State: N/a	
	Reusable State(s): State is TransactionEventsInQueueEnded	
Main (Test scenario)	Charging Station 1. Execute Reusable State TransactionEventsInQueueEnded 1. The Charging Stations sends a TransactionEventRequest <u>Note(s):</u> - The Charging Station will empty its Transaction message queue. This will contain one or more TransactionEventRequest messages	CSMS 2. The OCTT responds with a TransactionEventResponse
Tool validations	* Step 1: All messages: TransactionEventRequest - offline must be <i>true</i> One of the messages: TransactionEventRequest - eventType <i>Started</i> One of the messages: TransactionEventRequest - eventType <i>Ended</i> Post scenario validations: N/a	

NOTE If the Charging Station supports ISO15118, this testcase needs to be executed using EIM.

6.2.50. Page 217 - (2025-04) - TC_E_46_CS - Testcase updated to use the specialized Authorized15118 reusable state

Testcase updated to use the specialized Authorized15118 reusable state for starting the transaction, instead of the standard Authorized reusable state.

Test case name	End of charging process 15118
Test case Id	TC_E_46_CS
...	

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s):
State is <i>EVConnectedPreSession</i>
State is <i>Authorized15118</i>
State is <i>EnergyTransferStarted</i>

6.2.51. Page 221 - (2024-06) TC_F_04_CS Made mandatory in part 5, but prerequisite in part 6 was not updated

Removed Prerequisite(s):

Old	The Charging Station supports TxCtrlr.TxStartPoint ParkingBayOccupancy OR Authorized.
New	N/a

And added to Preparations:

Before (Preparations)
Configuration State:
...
- TxCtrlr.TxStartPoint is <i>ParkingBayOccupancy</i> OR <i>Authorized</i> (If supported)

6.2.52. Page 221 - (2025-02) TC_F_04_CS Prerequisite only if supported

Update Preparation(s):

Updated Preparations to only set TxStartPoint if supported:

Before (Preparations)
Configuration State:
...
- TxCtrlr.TxStartPoint is <i>ParkingBayOccupancy</i> OR <i>Authorized</i> (If supported)

6.2.53. Page 295 - (2024-09) - TC_J_XX_CS Meter Values

Meter values cannot have *location* = "EV", unless it is for measurand "SoC".

For all test cases in J add the following Post scenario validation:

...	...
Tool validations	<p>Post scenario validations:</p> <p>Message: <i>MeterValuesRequest/TransactionEventRequest</i></p> <ul style="list-style-type: none"> - ... - None of the provided <i>sampledValue</i>s shall have <i>location</i> = <i>EV</i>, except when <i>measurand</i> = <i>SoC</i>.

6.2.54. Page 236 - (2025-02) - TC_F_19_CS - The testcase ends while the firmware update is still ongoing

Testcase TC_F_19_CS starts a firmware update and ends the testcase while it is still ongoing. This may cause problems for the next testcase, without proper cleanup. Therefore an invalid signature is given to prevent a full firmware update as this can take quite some time. These messages are then handled by the cleanup, but these described steps are not part of the scope of the testcase and therefore the SUT is unable to fail during these steps.

Test case name	Trigger message - FirmwareStatusNotification - Downloading
Test case Id	TC_F_19_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a UpdateFirmwareResponse	1. The Test System sends a UpdateFirmwareRequest firmware.location is <Configured firmware_location> firmware.retrieveDateTime is <Current dateTime - 2 hours> firmware.installDateTime is omitted firmware.signingCertificate is <Configured signingCertificate> firmware.signature is <Configured invalid firmware signature>
...	
Note: Step 9 through 14 are cleanup to prevent an ongoing firmware update after the testcase is already ended. The behavior part of these steps is part of TC_L_06_CS and therefore not part of the scope for this testcase.	
9. The Charging Station sends a FirmwareStatusNotificationRequest . With status <i>Downloaded</i>	10. The Test System responds with a FirmwareStatusNotificationResponse .
11. The Charging Station sends a FirmwareStatusNotificationRequest . With status <i>InvalidSignature</i>	12. The Test System responds with a FirmwareStatusNotificationResponse .
13. The Charging Station sends a SecurityEventNotificationRequest . With type <i>InvalidFirmwareSignature</i>	14. The Test System responds with a SecurityEventNotificationResponse .

6.2.55. Page 272 - (2025-02) - TC_H_08_CS Reserve an unspecified EVSE - Accepted

Wrong token was used for RequestStartTransaction.

TC_H_08_CS: Reserve an unspecified EVSE - Accepted

Test case name	Reserve an unspecified EVSE - Accepted
Test case Id	TC_H_08_CS
...	...

Before (Preparations)	
...	
Main (Test scenario)	
Charging Station	CSMS
...	
5. Execute Reusable State <i>Authorized</i>	
Note(s): - <><Configured valid_idtoken_idtoken> is used for the authorization.	
...	

Tool validations

* Step 2:

Message: **ReserveNowResponse**

- **status** must be *Accepted*

* Step 3: **(Optional)**

Message: **StatusNotificationRequest**

- **connectorStatus** must be *Reserved*

- **evseld** must be *<Configured evseld>*

- **connectordId** must be *<Configured connectordId>*

Message: **NotifyEventRequest**

- **trigger** must be *Delta*

- **actualValue** must be "Reserved"

- **component.name** must be "Connector"

- **evse.id** must be *<Configured evseld>*

- **eves.connectordId** must be *<Configured connectordId>*

- **variable.name** must be "AvailabilityState"

(Optional)

Message: **NotifyEventRequest**

- **eventData[0].trigger** must be *Delta*

- **eventData[0].actualValue** must be *Available*

- **eventData[0].component.name** must be *EVSE*

- **eventData[0].variable.name** must be *AvailabilityState*

Post scenario validations:

N/a

6.2.56. Page 282 - (2025-02) - TC_H_17_CS - made more explicit on what to validate

Tool validations

* Step 2:

Message: **CancelReservationResponse**

- **status** must be *Accepted*

* Step 3:

For each connector on the *<Configured evseld>* one of the following messages must be sent:

Message: **StatusNotificationRequest**

- **connectorStatus** must be *Available*

- **evseld** must be *<Configured evseld>*

- **connectordId** must be *<Configured connectordId>*

Message: **NotifyEventRequest**

- **trigger** must be *Delta*

- **actualValue** must be "Available"

- **component.name** must be "Connector"

- **evse.id** must be *<Configured evseld>*

- **eves.connectordId** must be *<Configured connectordId>*

- **variable.name** must be "AvailabilityState"

Post scenario validations:

N/a

6.2.57. Page 297 - (2025-02) - TC_J_02_CS Clock-aligned Meter Values - reporting multiple phases

An erratum was added on adding support for validating reporting measurands on multiple phases: [TC_J_02_CS Clock-aligned Meter Values \(2024-09\)](#). However the validation is too strict, so this erratum supersedes it. Reporting measurands per phase must be done, when the energy meter supports it and it is applicable for the specified measurand.

Test case name	Clock-aligned Meter Values - Transaction ongoing	
Test case Id	TC_J_02_CS	
Use case Id(s)	J01	
...		
Before (Preparations)	Configuration State: AlignedDataInterval is <Configured clock-aligned Meter Values interval> AlignedDataSendDuringIdle is <i>false</i> (If implemented) RegisterValuesWithoutPhases is <i>false</i> (If implemented) Memory State: N/a	
	Reusable State(s): State is EnergyTransferStarted	
Main (Test scenario)	Charging Station	CSMS
...		
Tool validations	<p>Note: The following steps do not need to be sent in a specific order.</p> <ul style="list-style-type: none"> * Step 1: Message: MeterValuesRequest <ul style="list-style-type: none"> - meterValue[0].sampledValue[0].context must be Sample.Clock - meterValue[0].sampledValue must contain <An element per configured measurand at the AlignedDataMeasurands> <p>Notes :</p> <ul style="list-style-type: none"> - The measurand field may be omitted when the measurand is "Energy.Active.Import.Register" - It is possible that measurands are reported on multiple locations or phases, based on the capabilities of the energy meter . * Step 3: Message: TransactionEventRequest <ul style="list-style-type: none"> - triggerReason must be MeterValueClock - metervalue[0].sampledValue[0].context must be Sample.Clock - metervalue[0].sampledValue must contain <An element per configured measurand at the AlignedDataMeasurands> <p>Notes :</p> <ul style="list-style-type: none"> - The measurand field may be omitted when the measurand is "Energy.Active.Import.Register" - It is possible that measurands are reported on multiple locations or phases, based on the capabilities of the energy meter . <p>Post scenario validations:</p>	

6.2.58. Page 297 - (2024-09) - TC_J_02_CS Clock-aligned Meter Values

Meter values must be reported for all phases.

Test case name	Clock-aligned Meter Values - Transaction ongoing	
Test case Id	TC_J_02_CS	
Use case Id(s)	J01	
...		

Test case name	Clock-aligned Meter Values - Transaction ongoing	
Before (Preparations)	<p>Configuration State: AlignedDataInterval is <Configured clock-aligned Meter Values interval> AlignedDataSendDuringIdle is <i>false</i> (If implemented) RegisterValuesWithoutPhases is <i>false</i> (If implemented)</p>	
	<p>Memory State: N/a</p>	
	<p>Reusable State(s): State is <i>EnergyTransferStarted</i></p>	
Main (Test scenario)	Charging Station	CSMS
	...	
Tool validations	<p><i>Note: The following steps do not need to be sent in a specific order.</i></p> <ul style="list-style-type: none"> * Step 1: Message: MeterValuesRequest - meterValue[0].sampledValue[0].context must be <i>Sample.Clock</i> - meterValue[0].sampledValue must contain <An element per configured measurand at the AlignedDataMeasurands <i>for the number of phases reported in SupplyPhases</i>. The measurand field may be omitted when the measurand is "Energy.Active.Import.Register"> * Step 3: Message: TransactionEventRequest - triggerReason must be <i>MeterValueClock</i> - metervalue[0].sampledValue[0].context must be <i>Sample.Clock</i> - metervalue[0].sampledValue must contain <An element per configured measurand at the AlignedDataMeasurands <i>for the number of phases reported in SupplyPhases</i>. The measurand field may be omitted when the measurand is "Energy.Active.Import.Register"> 	
	<p>Post scenario validations: ...</p>	

6.2.59. Page 306 - (2024-06) - TC_J_10_CS - Remove reference to non-existing requirements [4697]

Test case name	Sampled Meter Values - EventType Ended
Test case Id	TC_J_10_CS
Use case Id(s)	J02 & (E06,E07,E08,E09,E10,E12)
Requirement(s)	J02.FR.01,J02.FR.02,J02.FR.03,J02.FR.04,J02.FR.10, E06.FR.11, E06.FR.17 , E07.FR.08, E07.FR.13 ,E08.FR.09,E09.FR.05,E10.FR.04,E12.FR.07
...	

6.2.60. Page 318 - (2025-04) TC_K_09_CS: Removed validFrom/To from test

Test case name	Clear Charging Profile - Clearing a TxDefaultProfile - With ongoing transaction
Test case Id	TC_K_09_CS
...	...

Before (Preparations)
Configuration State: SmartChargingCtrlr.LimitChangeSignificance is 1.0

Before (Preparations)
Memory State:
<i>SetChargingProfile</i> with
ChargingProfile 1:
chargingProfilePurpose is <i>TxDefaultProfile</i>
chargingProfileKind should be <i>Absolute</i>
stackLevel should be 0
evseld <Configured evseld>
validFrom <current dateTime - <Configured max time deviation> seconds>
validTo <current dateTime + <Configured max time deviation> + 401 seconds>
startSchedule <current dateTime>
numberPhases <Configured numberPhases>
ChargingSchedule:
duration 400 + <Configured max time deviation>
...

6.2.61. Page 343 - (2025-04) TC_K_23_CS: Removed validFrom/To from test

Test case name	Set Charging Profile - StartSchedule
Test case Id	TC_K_23_CS
...	...

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetChargingProfileResponse	<p>1. The Test System sends a SetChargingProfileRequest with chargingProfile.chargingProfilePurpose is <i>TxDefaultProfile</i> chargingProfile.chargingProfileKind is <i>Absolute</i></p> <p>evseld <configured evseld> chargingProfile.validFrom <current dateTime - <Configured max time deviation> + 50 seconds> chargingProfile.validTo <current dateTime + <Configured max time deviation> + 400 seconds> chargingProfile.chargingSchedule[0].startSchedule <current dateTime> + 60 seconds> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].numberPhases <Configured numberPhases> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit 6 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier></p>
...	...

6.2.62. Page 337 - (2025-04) TC_K_28_CS: Removed validFrom/To from test

Test case name	Set Charging Profile - TxDefaultProfile with transaction ongoing
Test case Id	TC_K_28_CS
...	...

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. The Charging Station responds with a SetChargingProfileResponse	<p>1. The Test System sends a SetChargingProfileRequest with chargingProfile.chargingProfilePurpose is <i>TxDefaultProfile</i> chargingProfile.chargingProfileKind is <i>Absolute</i> chargingProfile.chargingSchedule[0].duration is <i><400 + <Configured max time deviation> seconds></i> evseld <i><Configured evseld></i> chargingProfile.validFrom <i><current dateTime - <Configured max time deviation> seconds></i> chargingProfile.validTo <i><current dateTime + <Configured max time deviation> + 401 seconds></i> chargingProfile.chargingSchedule[0].startSchedule <i><current dateTime - <Configured max time deviation> seconds></i> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].numberPhases <i><Configured numberPhases></i> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod <i>0</i> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit <i>6 * <limit multiplier></i> ...</p>

6.2.63. Page 345 - (2025-02) TC_K_XX_CS: Use realistic values for composite schedules

In order to use realistic limits for charging profiles, the limit of a charging profile will be determined by *<Configured chargingRateUnit>* and *<Configured numberPhases>*.

Added new section to section K Smart Charging

Determine Charging Profile Limit Multiplier

Not all chargers support setting limits in A or W. This can be configured with the configuration variable *<Configured chargingRateUnit>*. To calculate the limit to be used, the following rules must be followed:

If *<Configured chargingRateUnit>* is A, then *<limit multiplier>* is 1
 If *<Configured chargingRateUnit>* is W and *<Configured numberPhases>* is 1, then *<limit multiplier>* is 230
 If *<Configured chargingRateUnit>* is W and *<Configured numberPhases>* is 2, then *<limit multiplier>* is 460
 If *<Configured chargingRateUnit>* is W and *<Configured numberPhases>* is 3, then *<limit multiplier>* is 690

Example 1

Given a test case is configured with:

<Configured chargingRateUnit> W
<Configured numberPhases> 2

When the test case specifies:

chargingProfile.chargingSchedule.chargingSchedulePeriod.limit *6 * <limit multiplier>*

Then it should set

chargingProfile.chargingSchedule.chargingSchedulePeriod.limit *2760*

Example 2

Given a test case is configured with:

<Configured chargingRateUnit> A
<Configured numberPhases> 3

When the test case specifies:

chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6 * <limit multiplier>
--

Then it should set

chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6

All test cases using limits are updated

Test case name	[...]
Test case Id	TC_K_01_CS / TC_K_03_CS / TC_K_04_CS / TC_K_09_CS / TC_K_10_CS / TC_K_13_CS / TC_K_21_CS / TC_K_22_CS / TC_K_23_CS / TC_K_28_CS / TC_K_60_CS / TC_K_37_CS / TC_K_38_CS / TC_K_40_CS / TC_K_41_CS / TC_K_56_CS / TC_K_58_CS
[...]	

Main (Test scenario)	
Charging Station	CSMS
[...]	<p>[...]</p> <p>Replace all *.limit assignments with:</p> <p>*.limit ? * <limit multiplier></p> <p>Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier></p>

Tool validations	
* Step 2:	<p>Replace all *.limit validations with:</p> <p>*.limit <?> * <limit multiplier></p> <p>Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier></p>
Post scenario validations:	[...]

6.2.64. Page 345 - (2024-06) TC_K_35_CS Get Charging Profile - Evseld > 0 + chargingProfilePurpose [773]

Change initial charging state from "N/A" to:

	Charging State: State is EnergyTransferStarted
--	---

6.2.65. Page 352 - (2025-04) - TC_K_39_CS - Validation of scheduleStart

Test case name	Get Composite Schedule - No ChargingProfile installed on Charging Station
Test case Id	TC_K_39_CS
...	...

Tool validations
<p>* Step 2: (Message: GetCompositeScheduleResponse) status Accepted evsId 0 scheduleStart <The time the GetCompositeScheduleRequest was transmitted +/- <Configured max time deviation>> duration is 300 chargingRateUnit <Configured chargingRateUnit> startPeriod 0</p> <p>Post scenario validations: ...</p>

6.2.66. Page 353 - (2025-04) - TC_K_40_CS: startSchedule improvements

Now using same startSchedule time for both profiles.

Test case name	Get Composite Schedule - Stacking ChargingProfiles
Test case Id	TC_K_40_CS
Use case Id(s)	K08
Requirement(s)	K08.FR.02,K08.FR.06
System under test	Charging Station
Description	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. 2 ChargingProfiles with same startSchedule and different stackLevels are submitted.

Before (Preparations)
Configuration State: N/a

Before (Preparations)

Memory State:

```
set <startScheduleTime> = <current dateTime> - <Configured max time deviation>
```

SetChargingProfile with

ChargingProfile 1:

chargingProfilePurpose is *TxDefaultProfile*

chargingProfileKind should be *Absolute*

stackLevel should be 0

evseld <Configured evseld>

validFrom <current dateTime - <Configured max time deviation> seconds>

validTo <current dateTime + <Configured max time deviation> + 401 seconds>

startSchedule <startScheduleTime>

numberPhases <Configured numberPhases>

ChargingSchedule:

duration 400 + <Configured max time deviation>

chargingRateUnit <Configured chargingRateUnit>

startPeriod 0, **limit** 6 * <limit multiplier>

startPeriod 100, **limit** 8 * <limit multiplier>

startPeriod 200, **limit** 10 * <limit multiplier>

Note: Check [\[csKSmartChargingChargingProfileLimits\]](#) for <limit multiplier>

ChargingProfile 2:

chargingProfilePurpose is *TxDefaultProfile*

chargingProfileKind should be *Absolute*

stackLevel should be 1

evseld <Configured evseld>

validFrom <current dateTime - <Configured max time deviation> seconds>

validTo <current dateTime + <Configured max time deviation> + 401 seconds>

startSchedule <startScheduleTime>

numberPhases <Configured numberPhases>

ChargingSchedule:

duration 150 + <Configured max time deviation>

chargingRateUnit <Configured chargingRateUnit>

startPeriod 0, **limit** 7 * <limit multiplier>

Note: Check [\[csKSmartChargingChargingProfileLimits\]](#) for <limit multiplier>

startPeriod 100, **limit** 9 * <limit multiplier>

Note: Check [\[csKSmartChargingChargingProfileLimits\]](#) for <limit multiplier>

Reusable State(s):

N/a

Main (Test scenario)

Charging Station	CSMS
2. The Charging Station responds with a GetCompositeScheduleResponse	1. The Test System sends a GetCompositeScheduleRequest with evseld <Configured evseld> duration is 350 chargingRateUnit <Configured chargingRateUnit>

Tool validations

* Step 2:
(Message: **GetCompositeScheduleResponse**)
status Accepted
evseld <Configured evseld>
ChargingSchedule:
duration 350
chargingRateUnit <Configured chargingRateUnit>
scheduleStart <The time the GetCompositeScheduleRequest was transmitted> plus/minus <Configured max time deviation>
Note: The period of time between the scheduleStart from the SetChargingProfileRequest with ChargingProfile_2 and the scheduleStart from the GetCompositeScheduleResponse is called x.
Note: The period of time between the scheduleStart from the SetChargingProfileRequest with ChargingProfile_1 and the scheduleStart from the GetCompositeScheduleResponse is called y.
startPeriod 0, limit 7 * <limit multiplier> (stackLevel 1)
startPeriod (100 - x), limit 9 * <limit multiplier> (stackLevel 1)
startPeriod (150 - x +<Configured max time deviation>), limit 8 * <limit multiplier> (stackLevel 0)
startPeriod (200 - x), limit 10 * <limit multiplier> (stackLevel 0)
Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>

Post scenario validations:

N/a

6.2.67. Page 355 - (2025-04) - TC_K_41_CS: startSchedule improvement

Test case name	Get Composite Schedule - Combining chargingProfilePurposes
Test case Id	TC_K_41_CS
...	...

Before (Preparations)

Configuration State:

N/a

Memory State:

set <startScheduleTime> = <current dateTime> - <Configured max time deviation> seconds

Note: Set MaxProfile for the next 24 hours:

SetChargingProfile with

ChargingProfile 1:

chargingProfilePurpose is ChargingStationMaxProfile

chargingProfileKind should be Absolute

stackLevel should be 0

evseld 0

startSchedule <startScheduleTime>

numberPhases <Configured numberPhases>

ChargingSchedule:

duration 86400

chargingRateUnit <Configured chargingRateUnit>

startPeriod 0, limit 10 * <limit multiplier>

Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>

Before (Preparations)	
<p>Note: Set a default profile for 300 seconds</p> <p>ChargingProfile 2:</p> <p>chargingProfilePurpose is <i>TxDefaultProfile</i></p> <p>chargingProfileKind should be <i>Absolute</i></p> <p>stackLevel should be 0</p> <p>evseld <Configured evseld></p> <p>validFrom <current dateTime -> <Configured max time deviation> seconds></p> <p>validTo <current dateTime +> <Configured max time deviation> + 401 seconds></p> <p>startSchedule <startScheduleTime></p> <p>numberPhases <Configured numberPhases></p> <p>ChargingSchedule:</p> <p>duration 300</p> <p>chargingRateUnit <Configured chargingRateUnit></p> <p>startPeriod 0,60,120,180,260, limit 6,10,8,15,8 * <limit multiplier></p> <p>Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier></p>	<p>Note: set a <i>TxProfile</i> for 260 seconds:</p> <p>ChargingProfile 3:</p> <p>chargingProfilePurpose is <i>TxProfile</i></p> <p>chargingProfileKind should be <i>Absolute</i></p> <p>stackLevel should be 0</p> <p>evseld <Configured evseld></p> <p>validFrom <current dateTime -> <Configured max time deviation> seconds></p> <p>validTo <current dateTime +> <Configured max time deviation> + 401 seconds></p> <p>startSchedule <startScheduleTime></p> <p>numberPhases <Configured numberPhases></p> <p>ChargingSchedule:</p> <p>duration 260</p> <p>chargingRateUnit <Configured chargingRateUnit></p> <p>startPeriod 0,50,140,200,240, limit 8,11,16,6,12 * <limit multiplier></p> <p>Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier></p>
Reusable State(s):	
N/a	
Main (Test scenario)	
Charging Station	CSMS
...	...
Tool validations	
<p>* Step 2:</p> <p>(Message: GetCompositeScheduleResponse)</p> <p>status Accepted</p> <p>evseld <Configured evseld></p> <p>ChargingSchedule:</p> <p>duration 400</p> <p>chargingRateUnit <Configured chargingRateUnit></p> <p>scheduleStart <The time the GetCompositeScheduleRequest was transmitted +/- <Configured max time deviation>></p> <p>Note: The period of time between the scheduleStart from the SetChargingProfileRequest with <i>ChargingProfile 2</i> and the scheduleStart from the GetCompositeScheduleResponse is called x.</p> <p>Note: The period of time between the scheduleStart from the SetChargingProfileRequest with <i>ChargingProfile 2</i> and the scheduleStart from the GetCompositeScheduleResponse is called y.</p> <p>startPeriod 0, limit 8 * <limit multiplier> (<i>TxProfile</i>)</p> <p>startPeriod (50 - x), limit 10 * <limit multiplier> (<i>ChargingStationMaxProfile</i>)</p> <p>startPeriod (200 - x), limit 6 * <limit multiplier> (<i>TxProfile</i>)</p> <p>startPeriod (240 - x), limit 10 * <limit multiplier> (<i>ChargingStationMaxProfile</i>)</p> <p>startPeriod (260 - x), limit 8 * <limit multiplier> (<i>TxDefaultProfile</i>)</p> <p>startPeriod (300 - x), limit 10 * <limit multiplier> (<i>ChargingStationMaxProfile</i>)</p> <p>Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier></p>	
Post scenario validations:	
N/a	

6.2.68. Page 354/355 - (2025-02) - TC_K_40_CS & TC_K_41_CS - Updated composite schedule validation

The startPeriod validation did not take the (to the duration added) <configured max time deviation> into account.

Test case name	Get Composite Schedule - Stacking ChargingProfiles
Test case Id	TC_K_40_CS
...	...

Tool validations

* Step 2:
(Message: **GetCompositeScheduleResponse**)
status Accepted
evsId <Configured evsId>
ChargingSchedule:
duration 350
chargingRateUnit <Configured chargingRateUnit>
scheduleStart <The time the GetCompositeScheduleRequest was transmitted \+/- <Configured max time deviation>>
Note: If <Configured chargingRateUnit> is W, then the **limit** field will be multiplied by 1000.
Note: The period of time between sending the second SetChargingProfileRequest and the **scheduleStart** from the GetCompositeScheduleResponse is called **x**.
Note: The period of time between the **scheduleStart** from the SetChargingProfileRequest with **ChargingProfile 3** and the **scheduleStart** from the GetCompositeScheduleResponse is called **x**.
Note: The period of time between the **scheduleStart** from the SetChargingProfileRequest with **ChargingProfile 2** and the **scheduleStart** from the GetCompositeScheduleResponse is called **y**.
startPeriod 0, **limit** 7
startPeriod (100 - x), **limit** 9
startPeriod (150 - x)
startPeriod (200 - x), **limit** 10
startPeriod 0, **limit** 8
startPeriod (50 - x), **limit** 10
startPeriod (200 - x), **limit** 6
startPeriod (240 - x), **limit** 10
startPeriod (260 - x + <Configured max time deviation>), **limit** 8 (**TxDefaultProfile**)
startPeriod (300 - y + <Configured max time deviation>), **limit** 10 (**ChargingStationMaxProfile**)

Post scenario validations:

N/a

Test case name	Get Composite Schedule - Combining chargingProfilePurposes
Test case Id	TC_K_41_CS
...	...

Before (Preparations)

Configuration State:

N/a

Memory State:

SetChargingProfile with

ChargingProfile 1:

chargingProfilePurpose is **ChargingStationMaxProfile**

chargingProfileKind should be **Absolute**

stackLevel should be 0

evsId 0

startSchedule <current dateTime - <Configured max time deviation> seconds>

numberPhases <Configured numberPhases>

ChargingSchedule:

duration <86400 <Configured max time deviation> seconds>

chargingRateUnit <Configured chargingRateUnit>

Note: If <Configured chargingRateUnit> is W, then the **limit** field will be multiplied by 1000.

startPeriod 0, **limit** 10

Before (Preparations)	
ChargingProfile 2: chargingProfilePurpose is <i>TxDefaultProfile</i> chargingProfileKind should be <i>Absolute</i> stackLevel should be 0 evseld <Configured evseld> validFrom <current dateTime - <Configured max time deviation> seconds> validTo <current dateTime + <Configured max time deviation> + 401 seconds> startSchedule <current dateTime - <Configured max time deviation> seconds> numberPhases <Configured numberPhases> ChargingSchedule: duration <300 <Configured max time deviation> seconds> chargingRateUnit <Configured chargingRateUnit> Note: If <Configured chargingRateUnit> is W, then the limit field will be multiplied by 1000. startPeriod 0,60,120,180,260, limit 6,10,8,15,8	ChargingProfile 3: chargingProfilePurpose is <i>TxProfile</i> chargingProfileKind should be <i>Absolute</i> stackLevel should be 0 evseld <Configured evseld> validFrom <current dateTime - <Configured max time deviation> seconds> validTo <current dateTime + <Configured max time deviation> + 401 seconds> startSchedule <current dateTime - <Configured max time deviation> seconds> numberPhases <Configured numberPhases> ChargingSchedule: duration <260 <Configured max time deviation> seconds> chargingRateUnit <Configured chargingRateUnit> Note: If <Configured chargingRateUnit> is W, then the limit field will be multiplied by 1000. startPeriod 0,50,140,200,240, limit 8,11,16,6,12
Reusable State(s):	
N/a	

Tool validations	
* Step 2: (Message: GetCompositeScheduleResponse) status Accepted evseld <Configured evseld> ChargingSchedule: duration 400 chargingRateUnit <Configured chargingRateUnit> Note: If <Configured chargingRateUnit> is W, then the limit field will be multiplied by 1000. Note: The period of time between sending the second SetChargingProfileRequest and the scheduleStart from the GetCompositeScheduleResponse is called x: startPeriod 0, limit 8 startPeriod (50 - x), limit 10 startPeriod (200 - x), limit 6 startPeriod (240 - x), limit 10 startPeriod (260 - x + <Configured max time deviation>), limit 8 (<i>TxDefaultProfile</i>) startPeriod (300 - x + <Configured max time deviation>), limit 10 (<i>ChargingStationMaxProfile</i>)	
Post scenario validations:	
N/a	

6.2.69. Page 355 - (2025-04) - TC_K_41_CS: Added missing EnergyTransferStarted reusable state

A transaction needs to be started to be able to set a TxProfile. The tool already started a transaction for this reason, but the testcase description lacks this information.

Test case name	Get Composite Schedule - Combining chargingProfilePurposes
Test case Id	TC_K_41_CS
...	

Before (Preparations)	
Configuration State:	
N/a	

Before (Preparations)
Memory State:
...
Reusable State(s): State is EnergyTransferStarted

6.2.70. Page 359 - (2025-04) - TC_K_53_CS: Added missing validations

Before (Preparations)
Configuration State:
N/a
Memory State:
N/a
Reusable State(s):
State is Authorized (local)
State is EVConnectedPreSession

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State Authorized15118	
2. The Charging Station sends a NotifyEVChargingNeedsRequest	3. The Test System responds with a NotifyEVChargingNeedsResponse with status Accepted
5. The Charging Station responds with a SetChargingProfileResponse	4. The Test System sends a SetChargingProfileRequest with chargingProfile.chargingProfilePurpose TxProfile chargingProfile.transactionId <transactionId> chargingProfile.chargingSchedule[0].id <Id generated by Test System> chargingProfile.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule[0].chargingSchedulePeriod.startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod.limit 6 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>
6. The Charging Station sends a NotifyEVChargingScheduleRequest	7. The Test System responds with a NotifyEVChargingScheduleResponse with status Accepted
4. Execute Reusable State EnergyTransferStarted	
8. The Charging Station sends a TransactionEventRequest	9. The Test System responds with a TransactionEventResponse

Tool validations
* Step 1: Message: TransactionEventRequest of <i>Authorized15118</i> step 3 - eventType must be <i>Updated</i> or <i>Started</i> - triggerReason must be <i>Authorized</i>
* Step 2: Message: NotifyEVChargingNeedsRequest IF chargingNeeds.acChargingParameters is <omitted> THEN chargingNeeds.dcChargingParameters must be <not omitted> END IF IF chargingNeeds.dcChargingParameters is <omitted> THEN chargingNeeds.acChargingParameters must be <not omitted> END IF
* Step 5: Message: SetChargingProfileResponse - status must be <i>Accepted</i>
* Step 6: Message: NotifyEVChargingScheduleRequest - chargingSchedule must be within bounds of chargingSchedule of step 4
* Step 8: Message: TransactionEventRequest - triggerReason must be <i>ChargingStateChanged</i> - transactionInfo.chargingState must be <i>Charging</i>
Post scenario validations: N/a

6.2.71. Page 360 - (2025-02) - TC_K_54_CS: EVConnected must be before Authorization

TC_K_54_CS: Charging with load leveling based on High Level Communication - No SASchedule (rejected)

Test case name	Charging with load leveling based on High Level Communication - No SASchedule (rejected)
Test case Id	TC_K_54_CS
...	...

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): State is <i>EVConnectedPreSession</i> State is <i>Authorized</i> (local)

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>Authorized15118</i>	
2. The Charging Station sends a NotifyEVChargingNeedsRequest .	3. The Test System responds with a NotifyEVChargingNeedsResponse . With status Rejected
[...]	

6.2.72. Page 362 - (2025-02) - TC_K_56_CS: EVConnected must be before Authorization

TC_K_56_CS: Charging with load leveling based on High Level Communication - Offline

Test case name	Charging with load leveling based on High Level Communication - Offline
Test case Id	TC_K_56_CS
...	...

Before (Preparations)
Configuration State: RetryBackOffWaitMinimum is <Configured RetryBackOffWaitMinimum>
Memory State: ...
Reusable State(s): State is <i>EVConnectedPreSession</i> State is <i>Authorized</i> (local)

6.2.73. Page 364 - (2025-02) - TC_K_57_CS: EVConnected must be before Authorization

TC_K_57_CS: Renegotiating a Charging Schedule - Initiated by EV

Test case name	Renegotiating a Charging Schedule - Initiated by EV
Test case Id	TC_K_57_CS
...	...

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): State is <i>EVConnectedPreSession</i> State is <i>Authorized</i> (local) State is <i>RenegotiateChargingLimits</i>

6.2.74. Page 362 - (2024-09) - TC_K_56_CS Removed expecting triggerReason=ChargingRateChanged [776]

A trigger reason ChargingStateChange must only be sent, when an external actor (not CSMS) changes the charging rate. Therefore, removed the check that triggerReason=ChargingStateChanged is sent. Also added a check that the EV charging schedule fits within the given charging profile.

Test case name	Charging with load leveling based on High Level Communication - Offline	
Test case Id	TC_K_56_CS	
...		
Main (Test scenario)	Charging Station	CSMS 6. The OCTT responds with a TransactionEventResponse.
	...	

Test case name	Charging with load leveling based on High Level Communication - Offline
Tool validations	<ul style="list-style-type: none"> * Step 3: (Message: NotifyEVChargingScheduleRequest) evseld <Configured evseld> chargingSchedule.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingSchedule.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod = 0 If <Configured chargingRateUnit> is W: chargingSchedule.chargingSchedule[0].chargingSchedulePeriod[0].limit <= 8000 Else: chargingSchedule.chargingSchedule[0].chargingSchedulePeriod[0].limit <= 8 * Step 5: <p>Message: TransactionEventRequest</p> <ul style="list-style-type: none"> - triggerReason must be <i>ChargingStateChanged</i> - transactionInfo.chargingState must be <i>Charging</i> - offline true <p>Post scenario validations:</p> <p>N/a</p>

6.2.75. Page 366 - (2025-02) - TC_K_58_CS

Before (Preparations)
Configuration State:
N/a
Memory State:
N/a
Reusable State(s):
<ul style="list-style-type: none"> State is <i>Authorized</i> (local) State is <i>EVConnectedPreSession</i> State is <i>RenegotiateChargingLimits</i> State is <i>EVConnectedPreSession</i> State is <i>Authorized15118</i> State is <i>EnergyTransferStarted</i>

6.2.76. Page 384 – TC_L_10_CS - Allow Download/InstallationFailed upon AcceptedCanceled

When a new firmware update is issued, the ongoing firmware update is canceled, but Charging Station may still send a FirmwareStatusNotification(DownloadFailed/InstallationFailed)

Test case name	Secure Firmware Update - AcceptedCanceled
Test case Id	TC_L_10_CS
...	...

Before (Preparations)
...

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a UpdateFirmwareResponse With status Accepted	<p>1. The Test System sends a UpdateFirmwareRequest with requestId = <#1></p> <p>firmware.installDateTime <Current DateTime - 2 hours></p> <p>firmware.location <Configured firmware_location></p> <p>firmware.retrieveDateTime <Current DateTime - 2 hours></p> <p>firmware.signingCertificate <Configured signingCertificate></p> <p>firmware.signature <Configured signature></p>

Main (Test scenario)	
3. The Charging Station sends a FirmwareStatusNotificationRequest With requestId <#1> and status Downloading	4. The Test System responds with a FirmwareStatusNotificationResponse
6. The Charging Station responds with a UpdateFirmwareResponse With requestId <#1> and status AcceptedCanceled	5. The Test System sends a UpdateFirmwareRequest with requestId = <#2> firmware.installDateTime <Current DateTime - 2 hours> firmware.location <Configured firmware_location> firmware.retrieveDateTime <Current DateTime - 2 hours> firmware.signingCertificate <Configured signingCertificate> firmware.signature <Configured signature>
7. The Charging Station sends a FirmwareStatusNotificationRequest With requestId <#2> and status Downloading	8. The Test System responds with a FirmwareStatusNotificationResponse
...	...

Tool validations
* Step 2: Message UpdateFirmwareResponse - status Accepted
* Step 3: Message FirmwareStatusNotificationRequest - status Downloading - requestId = <#1>
* Step 6: Message UpdateFirmwareResponse - status AcceptedCanceled A FirmwareStatusNotificationRequest <i>DownloadFailed</i> or <i>InstallationFailed</i> may be sent for requestId <#1> before or after step 6. (The requestId at the FirmwareStatusNotificationRequest messages must refer to the id <#2> from the second UpdateFirmwareRequest from this point on)
* Step 7: Message FirmwareStatusNotificationRequest - status Downloading ...
Post scenario validations: N/a

6.2.77. Page 387 - TC_L_06_CS - SecurityEventNotification and FirmwareStatusNotification can be sent in any order

The SecurityEventNotification with type *InvalidFirmwareSignature* and the FirmwareStatusNotification with status *InvalidSignature* can be sent in any order. These events are triggered simultaneously, so which message is transmitted first may vary.

Test case name	Secure Firmware Update - InvalidSignature
Test case Id	TC_L_06_CS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a UpdateFirmwareResponse	1. The Test System sends a UpdateFirmwareRequest with firmware.installDateTime <Current DateTime - 2 hours> firmware.location <Configured firmware_location> firmware.retrieveDateTime <Current DateTime - 2 hours> firmware.signingCertificate <Configured signingCertificate> firmware.signature <Configured invalid firmware signature>

Main (Test scenario)	
3. The Charging Station sends a FirmwareStatusNotificationRequest . With status Downloading	4. The Test System responds with a FirmwareStatusNotificationResponse .
5. The Charging Station sends a FirmwareStatusNotificationRequest . With status Downloaded	6. The Test System responds with a FirmwareStatusNotificationResponse .
Note: Step 7 through 10 can be sent in a different order.	
7. The Charging Station sends a FirmwareStatusNotificationRequest . With status InvalidSignature	8. The Test System responds with a FirmwareStatusNotificationResponse .
9. The Charging Station sends a SecurityEventNotificationRequest . With type InvalidFirmwareSignature	10. The Test System responds with a SecurityEventNotificationResponse .

6.2.78. Page 422 - (2025-02) - TC_M_15_CS - V2GCertificateChain is not installed before being retrieved

Test case name	Retrieve certificates from Charging Station - V2GCertificateChain
Test case Id	TC_M_15_CS
...	

Before (Preparations)
Configuration State: N/a
Memory State: RenewV2GChargingStationCertificate
Reusable State(s): N/a

6.2.79. Page 436 - (2025-02) - TC_N_01_CS - Made used component variable configurable

Before (Preparations)
Configuration State: The following monitors (on arbitrary variables) must be present as 'hard-wired' or 'preconfigured' or must have been configured by CSMS: <ul style="list-style-type: none"> - LowerThreshold - UpperThreshold - Delta - Periodic - PeriodicClockAligned The following monitors must have been configured by CSMS for Component Variable <Configured threshold monitor component variable>: <ul style="list-style-type: none"> - LowerThreshold using value <Configured threshold monitor component variable LowerThreshold trigger value> - UpperThreshold using value <Configured threshold monitor component variable UpperThreshold trigger value> - Periodic using value <Configured Clock Aligned MeterValues Interval> The following monitors must have been configured by CSMS for Component Variable <Configured numeric delta component variable>: <ul style="list-style-type: none"> - Delta using value <Configured numeric delta component variable Delta numeric trigger value> - PeriodicClockAligned using value <Configured Clock Aligned MeterValues Interval>
Memory State: N/a

Before (Preparations)
Reusable State(s): N/a

6.2.80. Page 455 - (2025-02) - TC_N_12_CS - Updating test case for using configuration variables

Test case name	Set Variable Monitoring - Value out of range - Delta monitor
[...]	
Prerequisite(s)	<p>Charging Station has implemented device model monitoring and MonitoringCtrlr::Enabled = true.</p> <p>This test case assumes the following component exists and can be monitored:</p> <ul style="list-style-type: none"> - Component "EVSE", evse "1", variable "AvailabilityState", monitor type <i>Delta</i> <p><i>Note: Variable_AvailabilityState is mandatory for an EVSE and it is likely (but not guaranteed), that it can be monitored.</i></p> <p>This test case assumes a numeric component variable exists which can be monitored.</p>

Main (Test scenario)	
Charging Station	<p>2. Charging Station responds with: SetVariableMonitoringResponse</p> <p><i>Install monitors</i></p> <p>1. Test System sends SetVariableMonitoringRequest with:</p> <ul style="list-style-type: none"> - setMonitoringData[0].value = -1 - setMonitoringData[0].type = <i>Delta</i> - setMonitoringData[0].severity = <Configured severity> - setMonitoringData[0].component.name = "EVSE" - setMonitoringData[0].component.evse.id = <Configured evselid> - setMonitoringData[0].variable.name = "AvailabilityState" - setMonitoringData[0].component = <Configured numeric delta component> - setMonitoringData[0].variable = <Configured numeric delta component variable>

Tool validations
* Step 2: Message: SetVariableMonitoringResponse with (in arbitrary order): <pre>setMonitoringResult = { - id is absent - status = Rejected - type = Delta - severity = <Configured severity> - component.name = "EVSE" - component.evse.id = <Configured evselid> - variable.name = "AvailabilityState" - component = <Configured numeric delta component variable> - variable = <Configured numeric delta component variable> - statusInfo is absent or statusInfo.reasonCode = "ValueOutOfRange" or statusInfo.reasonCode = "ValuePositiveOnly" }</pre>
Post scenario validations: N/A

6.2.81. Page 456 - (2025-02) - TC_N_13_CS - Updating test case for using more specific configuration variables

TC_N_13_CS: Set Variable Monitoring - Value out of range - Threshold monitor

Test case name	Set Variable Monitoring - Value out of range - Threshold monitor
Test case Id	TC_N_13_CS
Use case Id(s)	N04
Requirement(s)	N04.FR.13 N04.FR.06
System under test	Charging Station
Description	CSMS tries to set a threshold monitor with a value that is out of range.
Purpose	To test that Charging Station checks that value is within range of variable.
Prerequisite(s)	<p>Charging Station has implemented device model monitoring and MonitoringCtrlr::Enabled = true.</p> <p>This test case assumes the <Configured threshold monitor component variable> component.variable exists and can be monitored and has variableCharacteristics.maxLimit < <Configured threshold monitor value></p> <p>This test case assumes the <Configured threshold monitor component variable with maxLimit> component.variable exists and can be monitored and has variableCharacteristics.maxLimit < <Configured threshold monitor component variable with maxLimit exceeding maxLimit value></p> <p>+ Note: Variable _Power(maxLimit) is mandatory for an EVSE, but the actual value not, but it is likely (but not guaranteed), that it can be monitored._</p>

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	<p>CSMS</p> <p><i>Install monitors</i></p> <p>2. Charging Station responds with: SetVariableMonitoringResponse</p> <p>1. Test System sends SetVariableMonitoringRequest with:</p> <ul style="list-style-type: none"> - setMonitoringData[0].value = <Configured threshold monitor value> - setMonitoringData[0].value = <Configured threshold monitor component variable with maxLimit exceeding maxLimit value> - setMonitoringData[0].type = UpperThreshold - setMonitoringData[0].severity = <Configured severity> - setMonitoringData[0].component.name = <Configured threshold monitor component variable> - setMonitoringData[0].component.evse.id = <Configured evselid> - setMonitoringData[0].variable.name = <Configured threshold monitor component variable> - setMonitoringData[0].component = <Configured threshold monitor component variable with maxLimit> - setMonitoringData[0].variable = <Configured threshold monitor component variable with maxLimit>

Tool validations

* Step 2:

Message: **SetVariableMonitoringResponse** with (in arbitrary order):

```
setMonitoringResult = {
    - id is absent
    - status = Rejected
    - type = UpperThreshold
    - severity = <Configured severity>
    - component.name = <Configured threshold monitor component variable>
    - component.evse.id = <Configured evseId>
    - variable.name = <Configured threshold monitor component variable>
    - component = <Configured threshold monitor component variable with maxLimit>
    - variable = <Configured threshold monitor component variable with maxLimit>
    - statusInfo is absent or statusInfo.reasonCode = "ValueOutOfRange"
}
```

Post scenario validations:

N/A

6.2.82. Page 463 - (2025-02) TC_N_20_CS - Updating test case for using more specific configuration variables

TC_N_20_CS: Alert Event - Threshold value exceeded

Test case name	Alert Event - Threshold value exceeded
Test case Id	TC_N_20_CS
Use case Id(s)	N07
Requirement(s)	N07.FR.06, N07.FR.07, N07.FR.16, N07.FR.17
System under test	Charging Station
Description	A monitored variable exceeds a threshold monitor and causes a NotifyEventRequest message to be sent.
Purpose	To test that Charging Station supports threshold monitors
Prerequisite(s)	Charging Station has implemented device model monitoring and MonitoringCtrlr::Enabled = true.

Before (Preparations)
Configuration State: N/a
Memory State: This test requires the Monitoring Base to be set to <i>All</i> . - SetMonitoringBaseRequest with monitoringBase = <i>All</i> . Furthermore this test requires the existence of a <i>LowerThreshold</i> and <i>UpperThreshold</i> monitor on a (numerical) variable. Since it is not mandated which variables are required to be monitored, this test used the variable "Power" of component "EVSE". -setMonitoringData[0].value = <Configured threshold monitor value> - setMonitoringData[0].value = <Configured threshold monitor component variable UpperThreshold trigger value> - setMonitoringData[0].type = UpperThreshold -setMonitoringData[0].severity = <Configured severity> -setMonitoringData[0].component.name = <Configured threshold monitor component variable> -setMonitoringData[0].component.evse.id = <Configured evselid> -setMonitoringData[0].variable.name = <Configured threshold monitor component variable> - setMonitoringData[0].severity = 5 - setMonitoringData[0].component = <Configured threshold monitor component variable> - setMonitoringData[0].variable = <Configured threshold monitor component variable>
Set MonitoringLevel to 8
Notes: - If componentVariable is set to "Power" or "Current", the value is set to the configured maxLimit 100.0 - Take a threshold that can easily be exceeded.
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State EnergyTransferStarted or manually trigger the monitor. Notes: If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.	
2. Charging Station sends a NotifyEventRequest with: - Power exceeding upper threshold	3. Test System responds with a NotifyEventResponse

Main (Test scenario)	
<p>5. Charging Station responds with a SetVariableMonitoringResponse with:</p> <ul style="list-style-type: none"> - status Accepted 	<p>4. Test System sends a SetVariableMonitoringRequest with:</p> <ul style="list-style-type: none"> - type LowerThreshold - component.name <Configured threshold monitor component variable> - component.evse.id <configured evselid> - variable.name <Configured threshold monitor component variable> - value <Configured threshold monitor2 value> - component <Configured threshold monitor component variable> - variable <Configured threshold monitor component variable> - value <Configured threshold monitor component variable> <p>LowerThreshold trigger value></p> <p>Notes:</p> <p>If componentVariable is set to "Power" or "Current", the value is set to the configured maxLimit 10.0</p> <ul style="list-style-type: none"> - Take a threshold that won't be exceeded.
<p>6. Execute Reusable State StopAuthorized or manually trigger the second monitor.</p> <p>Notes: If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.</p>	
<p>7. Charging Station sends:</p> <p>NotifyEventRequest for 2 events:</p> <ul style="list-style-type: none"> - Returning below upper threshold (cleared) - Dropping below lower threshold <p>8. Test System responds:</p> <p>NotifyEventResponse</p> <p>Notes: Steps 2, 3, 7, and 8 may be repeated if the data is sent using two requests instead of one. Depending on the configuration the Charging Station may also send other notifications during step 4 and 9.</p>	

Tool validations
<p>* Step 2: Message: NotifyEventRequest with:</p> <ul style="list-style-type: none"> - generatedAt = <time of generation at Charging Station> - seqNo = 0 <p>and an eventData element with:</p> <ul style="list-style-type: none"> - eventId = <id1> - timestamp = <time of event at Charging Station> - trigger = Alerting - actualValue = <current power> (must be > <Configured threshold monitor value>) - cleared is absent or cleared = false - transactionId = <transaction id> (delivery of power is always in transaction) - variableMonitoringId = <monitor id1> - component.name = <Configured threshold monitor component variable> - component.evse.id = <Configured evselid> - variable.name = <Configured threshold monitor component variable> - component = <Configured threshold monitor component variable> - variable.name = <Configured threshold monitor component variable> <p>Other eventData elements can be ignored.</p>

Tool validations

* Step 7: Message: **NotifyEventRequest** with:

- **generatedAt** = <time of generation at Charging Station>
- **seqNo** = 0

and an **eventData** element with:

- **eventId** = <id2>
- **timestamp** = <time of event at Charging Station>
- **trigger** = Alerting
- **actualValue** = <current power> (must be =< <Configured threshold monitor value>)
- **cleared** is true
- **transactionId** = <transaction id> (delivery of power is always in transaction)
- **variableMonitoringId** = <monitor id1>
- **eventNotificationType** = CustomMonitor
- **component.name** = <Configured threshold monitor component variable>
- **component.evse.id** = <Configured evseld>
- **variable.name** = <Configured threshold monitor component variable>
- **component** = <Configured threshold monitor component variable>
- **variable.name** = <Configured threshold monitor component variable>

and an **eventData** element with:

- **eventId** = <id3>
- **timestamp** = <time of event at Charging Station>
- **trigger** = Alerting
- **actualValue** = <current power> (must be < <Configured threshold monitor2 value>)
- **cleared** is absent or **cleared** is false
- **transactionId** = <transaction id> (delivery of power is always in transaction)
- **variableMonitoringId** = <monitor id2>
- **eventNotificationType** = CustomMonitor
- **component.name** = <Configured threshold monitor component variable>
- **component.evse.id** = <Configured evseld>
- **variable.name** = <Configured threshold monitor component variable>
- **component** = <Configured threshold monitor component variable>
- **variable.name** = <Configured threshold monitor component variable>

Other **eventData** elements can be ignored. This can also be sent in two **NotifyEventRequests**, instead of one.

Post scenario validations:

N/A

6.2.83. Page 468 - (2024-06) TC_N_23_CS Offline Notification - OfflineMonitoringEventQueuingSeverity set higher than severityLevel of the monitor [772]

Test case name	Offline Notification - OfflineMonitoringEventQueuingSeverity set higher than severityLevel of the monitor																											
Test case Id	TC_N_23_CS																											
Use case Id(s)	N07																											
Requirement(s)	N07.FR.04																											
System under test	Charging Station																											
Description	Charging Station does not queue event notifications when offline.																											
Purpose	To test that Charging Station does not queue event notifications with a severity higher than OfflineMonitoringEventQueuingSeverity.																											
Prerequisite(s)	Charging Station is online at start of test for configuration. CS has implemented device model monitoring and MonitoringCtrlr::Enabled = true.																											
Before (Preparations)	<p>Configuration State: SetConfiguration with: - component.name = "MonitoringCtrlr" - variable.name = "OfflineQueueingSeverity" - attributeValue = <Configured Severity></p> <p>Memory State: Charging Station has custom or predefined monitors on variable AvailabilityState of Configured EVSE and Configured ConnectorId with severity = <Configured severity> + 1</p> <p>Reusable State(s): N/a</p>																											
Main (Test scenario)	<table border="1"> <tr> <td>Charging Station</td> <td>CSMS</td> </tr> <tr> <td><u>Manual Action:</u> Connect the EV and EVSE.</td> <td></td> </tr> <tr> <td>1. The Charging Station notifies the CSMS about the status change of the connector.</td> <td>2. The OCTT responds accordingly.</td> </tr> <tr> <td colspan="2"><u>Note(s):</u> Step 3, 4, 5, 6, 7, and 8 need to be executed when TxStartPoint contains EVConnected OR ParkingBayOccupancy</td> </tr> <tr> <td>3. The Charging Station sends a TransactionEventRequest</td> <td>4. The OCTT responds with a TransactionEventResponse</td> </tr> <tr> <td colspan="2"><u>Manual Action:</u> Take Charging Station offline.</td> </tr> <tr> <td colspan="2"><u>Manual Action:</u> Disconnect the EV and EVSE.</td> </tr> <tr> <td colspan="2"><u>Manual Action:</u> Connect the EV and EVSE.</td> </tr> <tr> <td colspan="2"><u>Note(s):</u> The tool will now wait for <Configured Transaction Duration> seconds</td> </tr> <tr> <td colspan="2"><u>Manual Action:</u> Bring Charging Station back online.</td> </tr> <tr> <td>5. The Charging Station sends a TransactionEventRequest</td> <td>6. The OCTT responds with a TransactionEventResponse</td> </tr> <tr> <td>7. The Charging Station sends a TransactionEventRequest</td> <td>8. The OCTT responds with a TransactionEventResponse</td> </tr> <tr> <td colspan="2"><u>Note(s):</u> The CS shall not send a NotifyEventRequest for AvailabilityState of EVSE and Connector. A StatusNotification may still be received.</td> </tr> </table>		Charging Station	CSMS	<u>Manual Action:</u> Connect the EV and EVSE.		1. The Charging Station notifies the CSMS about the status change of the connector.	2. The OCTT responds accordingly.	<u>Note(s):</u> Step 3, 4, 5, 6, 7, and 8 need to be executed when TxStartPoint contains EVConnected OR ParkingBayOccupancy		3. The Charging Station sends a TransactionEventRequest	4. The OCTT responds with a TransactionEventResponse	<u>Manual Action:</u> Take Charging Station offline.		<u>Manual Action:</u> Disconnect the EV and EVSE.		<u>Manual Action:</u> Connect the EV and EVSE.		<u>Note(s):</u> The tool will now wait for <Configured Transaction Duration> seconds		<u>Manual Action:</u> Bring Charging Station back online.		5. The Charging Station sends a TransactionEventRequest	6. The OCTT responds with a TransactionEventResponse	7. The Charging Station sends a TransactionEventRequest	8. The OCTT responds with a TransactionEventResponse	<u>Note(s):</u> The CS shall not send a NotifyEventRequest for AvailabilityState of EVSE and Connector. A StatusNotification may still be received.	
Charging Station	CSMS																											
<u>Manual Action:</u> Connect the EV and EVSE.																												
1. The Charging Station notifies the CSMS about the status change of the connector.	2. The OCTT responds accordingly.																											
<u>Note(s):</u> Step 3, 4, 5, 6, 7, and 8 need to be executed when TxStartPoint contains EVConnected OR ParkingBayOccupancy																												
3. The Charging Station sends a TransactionEventRequest	4. The OCTT responds with a TransactionEventResponse																											
<u>Manual Action:</u> Take Charging Station offline.																												
<u>Manual Action:</u> Disconnect the EV and EVSE.																												
<u>Manual Action:</u> Connect the EV and EVSE.																												
<u>Note(s):</u> The tool will now wait for <Configured Transaction Duration> seconds																												
<u>Manual Action:</u> Bring Charging Station back online.																												
5. The Charging Station sends a TransactionEventRequest	6. The OCTT responds with a TransactionEventResponse																											
7. The Charging Station sends a TransactionEventRequest	8. The OCTT responds with a TransactionEventResponse																											
<u>Note(s):</u> The CS shall not send a NotifyEventRequest for AvailabilityState of EVSE and Connector. A StatusNotification may still be received.																												

Test case name	Offline Notification - OfflineMonitoringEventQueuingSeverity set higher than severityLevel of the monitor
Tool validations	<ul style="list-style-type: none"> * Step 1: <i>(Optional:)</i> <p>Message: StatusNotificationRequest</p> <ul style="list-style-type: none"> - evseld <configured evseld> - connectorId <configured connectorId> - connectorStatus must be <i>Occupied</i> <p><i>(Required, but can be combined into one NotifyEventRequest:)</i></p> <p>Message: NotifyEventRequest</p> <ul style="list-style-type: none"> - eventData[0].trigger must be <i>Delta</i> - eventData[0].actualValue must be <i>Occupied</i> - eventData[0].component.name must be <i>Connector</i> - eventData[0].component.evse.id must be <i>Configured EVSE</i> - eventData[0].component.evse.connectorId must be <i>Configured ConnectorId</i> - eventData[0].variable.name must be <i>AvailabilityState</i> <p>Message: NotifyEventRequest</p> <ul style="list-style-type: none"> - eventData[0].trigger must be <i>Delta</i> - eventData[0].actualValue must be <i>Occupied</i> - eventData[0].component.name must be <i>EVSE</i> - eventData[0].component.evse.id must be <i>Configured EVSE</i> - eventData[0].variable.name must be <i>AvailabilityState</i> <ul style="list-style-type: none"> * Step 3: <p>Message: TransactionEventRequest</p> <ul style="list-style-type: none"> - triggerReason must be <i>CablePluggedIn</i> - transactionInfo.chargingState must be <i>EVConnected</i> <ul style="list-style-type: none"> * Step 5: <p>Message: TransactionEventRequest</p> <ul style="list-style-type: none"> - triggerReason must be <i>EVCommunicationLost</i> - transactionInfo.chargingState must be <i>Idle</i> <ul style="list-style-type: none"> * Step 7: <p>Message: TransactionEventRequest</p> <ul style="list-style-type: none"> - triggerReason must be <i>CablePluggedIn</i> - transactionInfo.chargingState must be <i>EVConnected</i>
	Post scenario validations: N/A

6.2.84. Page 470 - (2024-09) - TC_N_24_CS - Referring to incorrect use case and requirements [O20-4793]

Test case name	Set Variable Monitoring - Periodic event	
Test case Id	TC_N_24_CS	
Use case Id(s)	N04, N08	
Requirement(s)	N04.FR.01, N04.FR.08, N08.FR.05 and N08.FR.06	
...		
Main (Test scenario)	Charging Station	CSMS

Tool validations	<ul style="list-style-type: none"> ... Post scenario validations: N/A	

6.2.85. Page 492 - (2025-02) - TC_N_39_CS - Test case now searches suitable variables to do test with

TC_N_39_CS: Set Variable Monitoring - Component/Variable combination does NOT correspond

Test case name	Set Variable Monitoring - Component/Variable combination does NOT correspond
Test case Id	TC_N_39_CS
...	
Prerequisite(s)	<p>Charging Station supports Monitoring</p> <p>This test case assumes the Charging Station has a non-numeric Component Variable <Configured non-numeric delta component variable> and numeric Component Variable <Configured threshold monitor component variable>.</p>

Before (Preparations)
...
Memory State:
Variable monitor is already set with component.name = EVSE, variable.name = AvailabilityState, type = Delta N/a

Main (Test scenario)	
Charging Station	<p>CSMS</p> <p>1. The Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].type Delta setMonitoringData[0].value 1 setMonitoringData[0].component = <Configured non-numeric delta component variable> setMonitoringData[0].variable = <Configured non-numeric delta component variable></p>
2. The Charging Station responds with a SetVariableMonitoringResponse	<p>1. The Test System sends a SetVariableMonitoringRequest with setMonitoringData.type UpperThreshold setMonitoringData.variable.name Power setMonitoringData.component.name ChargingStation</p> <p>3. The Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].id <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> setMonitoringData[0].type Delta setMonitoringData[0].value 1 setMonitoringData[0].component = <Configured numeric delta component variable> setMonitoringData[0].variable = <Configured numeric delta component variable></p>

Main (Test scenario)	
4. The Charging Station responds with a GetMonitoringReportResponse	3. The Test System sends a GetMonitoringReportRequest with - requestId <Generated requestId>
6. The Charging Station responds with a GetMonitoringReportResponse	5. The Test System sends a GetMonitoringReportRequest with - requestId <Generated requestId>
5. The Charging Station sends a NotifyMonitoringReportRequest	6. The Test System responds with a NotifyMonitoringReportResponse
7. The Charging Station sends a NotifyMonitoringReportRequest	8. The Test System responds with a NotifyMonitoringReportResponse .
<u>Note(s):</u> - If tbc is True at Step 3 then step 3 and 4 will be repeated - If tbc is True at Step 7 then step 7 and 8 will be repeated	

Tool validations

* Step 2:

Message **SetVariableMonitoringResponse**
- **setMonitoringResult[0].status** Rejected
- **setMonitoringResult[0].type** UpperThreshold
- **setMonitoringResult[0].severity** <Configured severity>
- **setMonitoringResult[0].component.name** ChargingStation
- **setMonitoringResult[0].variable.name** Power

* Step 4:

Message **GetMonitoringReportResponse**
- **status** Accepted

* Step 5:

Message **NotifyMonitoringReportRequest**
- **monitor.component** EVSE
- **monitor.variable** AvailabilityState

* Step 2:

Message **SetVariableMonitoringResponse**
- **setMonitoringResult[0].id** <not omitted>
- **setMonitoringResult[0].status** Accepted
- **setMonitoringResult[0].type** Delta
- **setMonitoringResult[0].component** = <Configured non-numeric delta component variable>
- **setMonitoringResult[0].variable** = <Configured non-numeric delta component variable>

* Step 4:

Message **SetVariableMonitoringResponse**
- **setMonitoringResult[0].status** Rejected
- **setMonitoringResult[0].type** Delta
- **setMonitoringResult[0].component** = <Configured numeric delta component variable>
- **setMonitoringResult[0].variable** = <Configured numeric delta component variable>

* Step 6:

Message **GetMonitoringReportResponse**
- **status** Accepted

* Step 7:

Message **NotifyMonitoringReportRequest**

Must contain a monitor with

- **monitor[0].component** = <Configured non-numeric delta component variable>
- **monitor[0].variable** = <Configured non-numeric delta component variable>
- **monitor[0].variableMonitoring[0].id** = <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2>
- **monitor[0].variableMonitoring[0].value** = 1
- **monitor[0].variableMonitoring[0].type** = Delta

Post scenario validations:

- All report parts have been received

6.2.86. Page 472 - (2024-12) - TC_N_26_CS - Made test case more explicit and more time before ending

Note: This erratum was released in tag 2025-01, not the mentioned 2024-12.

TC_N_26_CS: Retrieve Log Information - Diagnostics Log - Upload failed

Main (Test scenario)

Charging Station

CSMS

Main (Test scenario)	
2. The Charging Station responds with a GetLogResponse	<p>1. The Test System sends a GetLogRequest with</p> <ul style="list-style-type: none"> - logType <i>DiagnosticsLog</i> - retries 3 - retryInterval <Configured retryInterval> - log.remoteLocation <Configured log location with non-existing path>
...	
Note(s):	
<ul style="list-style-type: none"> - Steps 3 & 4 are optional after the first attempt. - The Charging Station will perform step (3,) 5, four times with <Configured retryInterval> seconds in between. - Step 3-4, 5-6 and 3-6 may repeat multiple times depending on Charging Station's implementation. - The Test System waits at least (3 * <Configured retryInterval>), before ending the testcase. 	

Tool validations
...
* Step 3:
Must be sent exactly 1 or 4 times
Message LogStatusNotificationRequest
- status <i>Uploading</i>
- requestId Same Id as the <i>GetLogRequest</i>
* Step 5:
Must be sent exactly 1 or 4 times
Message LogStatusNotificationRequest
...
Post scenario validations:
- N/a

6.2.87. Page 470 - (2024-12) - TC_N_24_CS - Test case now searches suitable variable to do test with

Note: This erratum has been superseded by erratum: [Page 470 - \(2025-02\) - TC_N_24_CS - Updating test case for using more specific configuration variables](#)

TC_N_24_CS: Set Variable Monitoring - Periodic event

Test case name	Set Variable Monitoring - Periodic event
Test case Id	TC_N_24_CS
...	
Prerequisite(s)	<p>Charging Station has implemented device model monitoring and MonitoringCtrlr::Enabled = true.</p> <p>This test case assumes the device model exposes at least one component.variable which can be monitored.</p>

Before (Preparations)
...
Reusable State(s):
N/a

State is `[csCommunicatedBaseReport]`

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
<p>Search <code>[csCommunicatedBaseReport].baseReportData</code> to get a <code>baseReportData.reportData</code> WHERE <code>reportData.variableCharacteristics.supportsMonitoring</code> is <code>true</code> AND <code>reportData.component.instance</code> is <omitted> AND <code>reportData.instance.instance</code> is <omitted> AS <componentVariable></p>	
Set the monitor to generate a periodic event notification	
2. Charging Station responds with <code>SetVariableMonitoringResponse</code>	<p>1. Test System sends <code>SetVariableMonitoringRequest</code> with:</p> <ul style="list-style-type: none"> - <code>setMonitoringData[0].value = <Configured Clock Aligned MeterValues Interval></code> - <code>setMonitoringData[0].value = 2</code> - <code>setMonitoringData[0].type = Periodic</code> - <code>setMonitoringData[0].component.name = "EVSE"</code> - <code>setMonitoringData[0].component.evse.id = <Configured evselid></code> - <code>setMonitoringData[0].variable.name = "AvailabilityState"</code> - <code>setMonitoringData[0].component.name = <componentVariable.component.name></code> - <code>setMonitoringData[0].component.evse.id = <componentVariable.component.evse.id></code> - <code>setMonitoringData[0].variable.name = <componentVariable.variable.name></code>
3. Charging Station generates <code>NotifyEventRequest</code> for <code>EVSE #1::AvailabilityState</code> every <Configured Clock Aligned MeterValues Interval> seconds. 3. Charging Station sends a <code>NotifyEventRequest</code>	<p>4. Test System responds with a <code>NotifyEventResponse</code></p>
<p><u>Note(s): Step 3 and 4 will repeat every 2 seconds</u></p>	

Tool validations
* Step 2: Message: <code>SetVariableMonitoringResponse</code> with: <code>setMonitoringResult[0].status = Accepted</code> <code>setMonitoringResult[0].component.name = "EVSE"</code> <code>setMonitoringResult[0].component.evse.id = <Configured evselid></code> <code>setMonitoringResult[0].variable.name = "AvailabilityState"</code> <code>setMonitoringResult[0].component.name = <componentVariable.component.name></code> <code>setMonitoringResult[0].component.evse.id = <componentVariable.component.evse.id></code> <code>setMonitoringResult[0].variable.name = <componentVariable.variable.name></code> <code>setMonitoringResult[0].attributeStatusInfo is absent or attributeStatusInfo.reasonCode = "NoError"</code>
* Step 3: Message: a <code>NotifyEventRequest</code> message every <Configured Clock Aligned MeterValues Interval> seconds with: Message: <code>NotifyEventRequest</code> every 2 seconds with: with an <code>eventData</code> element with: <ul style="list-style-type: none"> - <code>trigger = Periodic</code> - <code>component.name = "EVSE"</code> - <code>component.evse.id = <Configured evselid></code> - <code>variable.name = "AvailabilityState"</code> - <code>component.name = <componentVariable.component.name></code> - <code>component.evse.id = <componentVariable.component.evse.id></code> - <code>variable.name = <componentVariable.variable.name></code>
Post scenario validations: N/A

6.2.88. Page 470 - (2025-02) - TC_N_24_CS - Updating test case for using more specific configuration variables

Before (Preparations)
Configuration State: N/a
Memory State: N/a Set MonitoringLevel to 8
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
<i>Set the monitor to generate a periodic event notification</i>	
2. Charging Station responds with SetVariableMonitoringResponse	<p>1. Test System sends SetVariableMonitoringRequest with:</p> <ul style="list-style-type: none"> - setMonitoringData[0].value = <Configured Clock Aligned MeterValues Interval> - setMonitoringData[0].type = Periodic - setMonitoringData[0].component.name = "EVSE" - setMonitoringData[0].component.evse.id = <Configured evseId> - setMonitoringData[0].variable.name = "AvailabilityState" - setMonitoringData[0].severity = 5 - setMonitoringData[0].component = <Configured threshold monitor component variable> - setMonitoringData[0].variable = <Configured threshold monitor component variable>
3. Charging Station generates NotifyEventRequest for EVSE #1::AvailabilityState every <Configured Clock Aligned MeterValues Interval> seconds.	4. Test System responds with a NotifyEventResponse
3. Charging Station sends a NotifyEventRequest	
Note(s): Step 3 and 4 will repeat every <Configured Clock Aligned MeterValues Interval> seconds	

Tool validations
* Step 2: Message: SetVariableMonitoringResponse with: <pre>setMonitoringResult[0].status = Accepted setMonitoringResult[0].component.name = "EVSE" setMonitoringResult[0].component.evse.id = <Configured evseId> setMonitoringResult[0].variable.name = "AvailabilityState" setMonitoringResult[0].type = Periodic setMonitoringResult[0].severity = 5 setMonitoringResult[0].component = <Configured threshold monitor component variable> setMonitoringResult[0].variable = <Configured threshold monitor component variable> setMonitoringResult[0].attributeStatusInfo is absent or attributeStatusInfo.reasonCode = "NoError"</pre>

Tool validations
* Step 3: Message: a NotifyEventRequest message every <Configured Clock Aligned MeterValues Interval> seconds with: Message: NotifyEventRequest every <Configured Clock Aligned MeterValues Interval> seconds with: with an eventData element with: - trigger = <i>Periodic</i> - component.name = "EVSE" - component.evse.id = 1 - variable.name = "AvailabilityState" - component = <Configured threshold monitor component variable> - variable = <Configured threshold monitor component variable>
Post scenario validations: N/A

6.2.89. Page 482 - (2025-02) - TC_N_63_CS - Clear Customer Information - add manual action to stop session

Note: This erratum is extended by erratum: [Page 482 - \(2025-04\) - TC_N_63_CS - Added missing reusable state EnergyTransferStarted at before steps](#)

Test case name	Clear Customer Information - Clear and report - customerCertificate
Test case Id	TC_N_63_CS
...	...

Before (Preparations)
Configuration State: - AuthCtrlr.Enabled is <i>true</i> - AuthCtrlr.DisableRemoteAuthorization is <i>false</i> - ISO15118Ctrlr.CentralContractValidationAllowed is <i>false</i> - ISO15118Ctrlr.V2GCertificateInstallationEnabled is <i>true</i> - ISO15118Ctrlr.ContractCertificateInstallationEnabled is <i>true</i> - ISO15118Ctrlr.PnCEnabled is <i>true</i> - ISO15118Ctrlr.SeecId is <i>NL</i> - ISO15118Ctrlr.CountryName is <i>secoid</i> - ISO15118Ctrlr.OrganizationName is <i>OCA</i>
Memory State: RenewV2GChargingStationCertificate (If none are present, when checking with GetInstalledCertificateIds.certificateType = V2GCertificateChain)
Reusable State: Execute Reusable State EVConnectedPreSession Execute Reusable State Authorized15118 (PnC)

Main (Test scenario)	
Charging Station	CSMS
Manual action: EV ends the charging session.	
Note: The Charging Station receives a SessionStopReq(Terminate) message from the EV to finish the transaction.	
2. The Charging Station responds with a CustomerInformationResponse	1. The Test System sends a CustomerInformationRequest with - report <i>true</i> AND - clear <i>true</i> AND - customerCertificate <i>certificate hash data of contract certificate</i>

Main (Test scenario)	
3. The Charging Station sends a NotifyCustomerInformationRequest	4. The Test System responds with a NotifyCustomerInformationResponse
Note(s): - If tbc is True at Step 3 then step 3 and 4 will be repeated	
6. The Charging Station responds with a CustomerInformationResponse	5. The Test System sends a CustomerInformationRequest with - report true AND - clear false AND - customerCertificate certificate hash data of contract certificate
7. The Charging Station sends a NotifyCustomerInformationRequest	8. The Test System responds with a NotifyCustomerInformationResponse
Note(s): - If tbc is True at Step 7 then step 7 and 8 will be repeated	

Tool validations
...
Post scenario validations: - All report parts have been received

6.2.90. Page 487 - (2025-04) - TC_N_36_CS - **LogStatusNotification(AcceptedCanceled) allowed before GetLogResponse**

It is allowed that the LogStatusNotification(AcceptedCanceled) for the canceled GetLog is sent before the new GetLogRequest is responded to by a GetLogResponse.

Test case name	Retrieve Log Information - Second Request
Test case Id	TC_N_36_CS
...	...

Before (Preparations)
...

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a GetLogResponse	1. The Test System sends a GetLogRequest with logType <Configured logType> and requestId <#1>
Note(s): - Charging Station is uploading log file	
3. The Charging Station sends a LogStatusNotificationRequest	4. The Test System responds with a LogStatusNotificationResponse .
Note(s): - Charging Station cancels uploading the first log file	
6. The Charging Station responds with a GetLogResponse	5. The Test System sends a GetLogRequest with logType <Configured logType> and requestId <#2>
Step 7 is allowed to occur before step 6 7. The Charging Station sends a LogStatusNotificationRequest	8. The Test System responds with a LogStatusNotificationResponse .
Note(s): - Charging Station is uploading log file	

Main (Test scenario)	
9. The Charging Station sends a LogStatusNotificationRequest	10. The Test System responds with a LogStatusNotificationResponse .
Note(s): - Log file is uploaded	
11. The Charging Station sends a LogStatusNotificationRequest	12. The Test System responds with a LogStatusNotificationResponse .

Tool validations
* Step 2: Message GetLogResponse - status Accepted
* Step 3: Message LogStatusNotificationRequest - status Uploading - requestId <#1>
Step 7 is allowed to occur before step 6
* Step 6: Message GetLogResponse - status AcceptedCanceled
* Step 7: Message LogStatusNotificationRequest - status AcceptedCanceled - requestId <#1>
* Step 9: Message LogStatusNotificationRequest - status Uploading - requestId <#2>
* Step 11: Message LogStatusNotificationRequest - status Uploaded - requestId <#2>
Post scenario validations: - N/a

6.2.91. Page 482 - (2025-04) - TC_N_63_CS - Added missing reusable state EnergyTransferStarted at before steps

Note: This erratum extends erratum: [Page 482 - \(2025-02\) - TC_N_63_CS - Clear Customer Information - add manual action to stop session](#)

Test case name	Clear Customer Information - Clear and report - customerCertificate
Test case Id	TC_N_63_CS
...	

Before (Preparations)
Configuration State: ...
Memory State: RenewV2GChargingStationCertificate (If none are present, when checking with GetInstalledCertificateIds.certificateType = V2GCertificateChain)

Before (Preparations)
Reusable State:
Execute Reusable State EVConnectedPreSession
Execute Reusable State Authorized15118 (PnC)
Execute Reusable State EnergyTransferStarted

6.2.92. Page 493 - (2024-09) - TC_N_41_CS - Set Variable Monitoring - Return to FactoryDefault

Moved preconfigured monitor to Prerequisite.

Test case name	Set Variable Monitoring - Return to FactoryDefault	
Test case Id	TC_N_41_CS	
...		
Prerequisite(s)	Charging Station supports Monitoring and a preconfigured monitor exists with <i>id <Preconfigured monitor id></i> for component <i>EVSE</i> and variable <i>AvailabilityState</i> and type = <i>Delta</i> and severity = <i><Preconfigured severity></i>	
...		
Before (Preparations)	Configuration state: N/a Memory state: a preconfigured monitor exists with <i>id <Preconfigured monitor id></i> for component <i>EVSE</i> and variable <i>AvailabilityState</i> and type = <i>Delta</i> and severity = <i><Preconfigured severity></i> Charging State: N/a	
Main (Test scenario)	Charging Station	CSMS
	...	

6.2.93. Page 482 - (2024-09) - TC_N_63_CS - Clear Customer Information - Clear and report - customerCertificate

Test case design top stop transaction was not correct for an ISO 15118 session.

Test case name	Clear Customer Information - Clear and report - customerCertificate	
Test case Id	TC_N_63_CS	
...		
Before (Preparations)	Configuration State: N/a Memory State: N/a Charging State: Execute Reusable State EVConnectedPreSession Execute Reusable State Authorized15118 Execute Reusable State ParkingBayUnoccupied	
...		

Test case name	Clear Customer Information - Clear and report - customerCertificate	
Main (Test scenario)	Charging Station	CSMS
	Note : The Charging Station receives a SessionStopReq(Terminate) message from the EV to finish the transaction.	
	2. The Charging Station responds with a CustomerInformationResponse	1. The OCTT sends a CustomerInformationRequest with - report true AND - clear true AND - customerCertificate customer information used in the transaction
	3. The Charging Station sends a NotifyCustomerInformationRequest	4. The OCTT responds with a NotifyCustomerInformationResponse
	Note(s): - If tbc is True at Step 3 then step 3 and 4 will be repeated	
	6. The Charging Station responds with a CustomerInformationResponse	5. The OCTT sends a CustomerInformationRequest with - report true AND - clear false AND - customerCertificate customer information used in the transaction
	7. The Charging Station sends a NotifyCustomerInformationRequest	8. The OCTT responds with a NotifyCustomerInformationResponse
	Note(s): - If tbc is True at Step 7 then step 7 and 8 will be repeated	
Tool validations	...	

6.2.94. Page 482 - (2025-02) - TC_N_63_CS - Added missing configuration state and authorize explicit using Plug and Charge (PnC)

Before (Preparations)
Configuration State: N/a
- AuthCtrlr.Enabled is true - AuthCtrlr.DisableRemoteAuthorization is false - ISO15118Ctrlr.CentralContractValidationAllowed is false - ISO15118Ctrlr.V2GCertificateInstallationEnabled is true - ISO15118Ctrlr.ContractCertificateInstallationEnabled is true - ISO15118Ctrlr.PnCEnabled is true - ISO15118Ctrlr.SeccId is NL - ISO15118Ctrlr.CountryName is seccId - ISO15118Ctrlr.OrganizationName is OCA
Memory State: N/a
Reusable State: Execute Reusable State EVConnectedPreSession Execute Reusable State Authorized15118 (PnC)

6.2.95. Page 493 - (2024-09) - TC_N_41_CS - Set Variable Monitoring - Return to FactoryDefault

Moved preconfigured monitor to Prerequisite.

Test case name	Set Variable Monitoring - Return to FactoryDefault	
Test case Id	TC_N_41_CS	
...		
Prerequisite(s)	Charging Station supports Monitoring and a preconfigured monitor exists with <i>id <Preconfigured monitor id></i> for component <i>EVSE</i> and variable <i>AvailabilityState</i> and type = <i>Delta</i> and severity = <i><Preconfigured severity></i>	
Before (Preparations)	Configuration state: N/a Memory state: a preconfigured monitor exists with <i>id <Preconfigured monitor id></i> for component <i>EVSE</i> and variable <i>AvailabilityState</i> and type = <i>Delta</i> and severity = <i><Preconfigured severity></i> Charging State: N/a	
Main (Test scenario)	Charging Station	CSMS
	...	

6.2.96. Page 493 - (2025-02) - TC_N_41_CS - Made less dependent on test case configuration variables, enables predefines monitors

TC_N_41_CS: Set Variable Monitoring - Return to FactoryDefault

Test case name	Set Variable Monitoring - Return to FactoryDefault	
Test case Id	TC_N_41_CS	
Use case Id(s)	N03	
Requirement(s)	N03.FR.04, N04.FR.15	
System under test	Charging Station	
Description	This test case describes how the CSMS requests the Charging Station to overrule a preconfigured monitor by a custom monitor. When monitoringBase is set to FactoryDefault the preconfigured monitor must return.	
Purpose	To verify if the Charging station is able to correctly restore monitors to FactoryDefault.	
Prerequisite(s)	Charging Station supports Monitoring and a preconfigured monitor exists with <i>id <Preconfigured monitor id></i> for component <i>EVSE</i> and variable <i>AvailabilityState</i> and type = <i>Delta</i> and severity = <i><Preconfigured severity></i> - Charging Station supports Monitoring - A preconfigured monitor exists with <i>id <Configured preconfigured monitor id></i> .	

Before (Preparations)
Configuration State: N/a
Memory State: N/a MonitoringBase has been set to <i>FactoryDefault</i>
Reusable State: N/a
Main (Test scenario)

Charging Station	CSMS
------------------	------

Main (Test scenario)	
2. The Charging Station responds with GetMonitoringReportResponse	1. The Test System sends GetMonitoringReportRequest with requestId = <Generated requestId>
3. The Charging Station sends NotifyMonitoringReportRequest	4. The Test System responds with NotifyMonitoringReportResponse
Note(s) : - If NotifyMonitoringReportRequest.tbc is True in Step 3 then step 3 and 4 will be repeated	
Search NotifyMonitoringReportRequest.monitoringReportData of step 3 to get a monitoringReportData.monitor WHERE monitor.variableMonitoring.id is <Configured preconfigured monitor id> AS <preconfiguredMonitor>	
6. The Charging Station responds with a SetVariableMonitoringResponse	5. The Test System sends a SetVariableMonitoringRequest with setMonitoringData.id <Preconfigured monitor id> AND setMonitoringData.type Delta setMonitoringData.severity <Preconfigured severity> + 1 setMonitoringData.id <preconfiguredMonitor.variableMonitoring.id> setMonitoringData.transaction <preconfiguredMonitor.variableMonitoring.transaction> setMonitoringData.value <preconfiguredMonitor.variableMonitoring.value> setMonitoringData.type <preconfiguredMonitor.variableMonitoring.type> setMonitoringData.severity IF <preconfiguredMonitor.variableMonitoring.severity> < 9 THEN <preconfiguredMonitor.variableMonitoring.severity> + 1 ELSE 5 END IF setMonitoringData.component <preconfiguredMonitor.component> setMonitoringData.variable <preconfiguredMonitor.variable>
8. The Charging Station responds with a GetMonitoringReportResponse	7. The Test System sends a GetMonitoringReportRequest with - requestId <Generated requestId> - id <Preconfigured monitor id> - componentVariable.component.name EVSE - componentVariable.component.evse.id evselid - componentVariable.variable.name AvailabilityState - monitoringCriteria DeltaMonitoring - id <preconfiguredMonitor.variableMonitoring.id> - componentVariable.component <preconfiguredMonitor.component> - componentVariable.variable <preconfiguredMonitor.variable>
9. The Charging Station sends a NotifyMonitoringReportRequest	10. The Test System responds with a NotifyMonitoringReportResponse .
12. The Charging Station responds with a SetMonitoringBaseResponse with - status Accepted	11. The Test System sends a SetMonitoringBaseRequest with - monitoringBase FactoryDefault

Main (Test scenario)	
14. The Charging Station responds with a GetMonitoringReportResponse	<p>13. The Test System sends a GetMonitoringReportRequest with</p> <ul style="list-style-type: none"> - requestId <Generated requestId> - id <Preconfigured monitor id> - componentVariable.component.name _EVSE - componentVariable.component.evse.id evselid - componentVariable.variable.name AvailabilityState - monitoringCriteria DeltaMonitoring - id <preconfiguredMonitor.variableMonitoring.id> - componentVariable.component <preconfiguredMonitor.component> - componentVariable.variable <preconfiguredMonitor.variable>
15. The Charging Station sends a NotifyMonitoringReportRequest	16. The Test System responds with a NotifyMonitoringReportResponse .

Tool validations

* Step 2:

Message GetMonitoringReportResponse

- status Accepted

* Step 6:

Message SetVariableMonitoringResponse

- setMonitoringResult[0].status Accepted

- setMonitoringResult[0].type Delta

- setMonitoringResult[0].component.name EVSE

- setMonitoringResult[0].variable.name AvailabilityState

- setMonitoringResult[0].component <preconfiguredMonitor.component>

- setMonitoringResult[0].variable <preconfiguredMonitor.variable>

* Step 8:

Message GetMonitoringReportResponse

- status Accepted

* Step 9:

Message NotifyMonitoringReportRequest

- monitor.component.name EVSE

- monitor.variable.name AvailabilityState

- monitor.variableMonitoring.id <Preconfigured id>

- monitor.variableMonitoring.severity <Preconfigured severity> + 1

Should contain monitor:

- monitor.component <preconfiguredMonitor.component>

- monitor.variable <preconfiguredMonitor.variable>

- monitor.variableMonitoring.id <preconfiguredMonitor.variableMonitoring.id>

- monitor.variableMonitoring.severity IF <preconfiguredMonitor.variableMonitoring.severity> < 9 THEN

<preconfiguredMonitor.variableMonitoring.severity> + 1

ELSE

5

END IF

* Step 15:

Message NotifyMonitoringReportRequest

- monitor.component.name EVSE

- monitor.variable.name AvailabilityState

- monitor.variableMonitoring.severity <Preconfigured severity> + 1

- monitor.variableMonitoring.id <Preconfigured id>

- monitor.variableMonitoring.severity <Preconfigured severity> + 1

Should contain monitor:

- monitor.component <preconfiguredMonitor.component>

- monitor.variable <preconfiguredMonitor.variable>

- monitor.variableMonitoring.id <preconfiguredMonitor.variableMonitoring.id>

- monitor.variableMonitoring.severity <preconfiguredMonitor.variableMonitoring.severity>

Post scenario validations:

- All report parts have been received

6.2.97. Page 495 - (2024-11) - TC_N_43_CS - Remove incorrect tool validation StatusInfo

Test case name	Set Variable Monitoring - First SetMonitoringData and third SetMonitoringData are valid, but the second contains an out of range value
Test case Id	TC_N_43_CS
...	...

Tool validations
* Step 2: Message: SetVariableMonitoringResponse with (in arbitrary order): setMonitoringResult[1] = { - status = Accepted - type = UpperThreshold - statusInfo is absent or statusInfo.reasonCode = "NoError" } setMonitoringResult[2] = { - status = Rejected - type = Delta - statusInfo is absent or statusInfo.reasonCode = "NoError" (Removed) } setMonitoringResult[3] = { - status = Accepted - type = LowerThreshold - statusInfo is absent or statusInfo.reasonCode = "NoError" }
Post scenario validations: - N/a

6.2.98. Page 495 - (2025-02) - TC_N_43_CS - Updating test case for using more specific configuration variables

TC_N_43_CS: Set Variable Monitoring - First SetMonitoringData and third SetMonitoringData are valid, but the second contains an out of range value

Test case name	Set Variable Monitoring - First SetMonitoringData and third SetMonitoringData are valid, but the second contains an out of range value
Test case Id	TC_N_43_CS
Use case Id(s)	N04
Requirement(s)	N/a
System under test	Charging Station
Description	This test case describes how the CSMS requests the Charging Station to set monitoring triggers on Variables. Multiple triggers can be set for upper or lower thresholds, delta changes or periodic reporting.
Purpose	To verify if the Charging station is able to correctly respond when one of requested variable monitor data is out of range replace as described at the OCPP specification.
Prerequisite(s)	Charging Station supports Monitoring This test case assumes the Charging Station has a numeric Component Variable <Configured numeric delta component variable> and numeric Component Variable <Configured threshold monitor component variable>.

Before (Preparations)
Configuration State: N/a

Before (Preparations)	
Memory State:	N/a
Reusable State:	N/a
Main (Test scenario)	
Charging Station	<p>CSMS</p> <p>2. The Charging Station responds with a SetVariableMonitoringResponse</p> <p>1. The Test System sends a SetVariableMonitoringRequest with</p> <ul style="list-style-type: none"> - setMonitoringData.component.name = <Configured threshold monitor component variable> - setMonitoringData.variable.name = <Configured threshold monitor component variable> - setMonitoringData[0].value = <Configured threshold monitor value> - setMonitoringData[0].component = <Configured threshold monitor component variable> - setMonitoringData[0].variable = <Configured threshold monitor component variable> - setMonitoringData[0].value = <Configured threshold monitor component variable UpperThreshold trigger value> - setMonitoringData[0].type = <i>UpperThreshold</i> <ul style="list-style-type: none"> - setMonitoringData[1].component = <Configured numeric delta component variable> - setMonitoringData[1].variable = <Configured numeric delta component variable> - setMonitoringData[1].value = -1.0 - setMonitoringData[1].type = <i>Delta</i> <ul style="list-style-type: none"> - setMonitoringData[2].value = <Configured threshold monitor2 value> - setMonitoringData[2].component = <Configured threshold monitor component variable> - setMonitoringData[2].variable = <Configured threshold monitor component variable> - setMonitoringData[2].value = <Configured threshold monitor component variable LowerThreshold trigger value> - setMonitoringData[2].type = <i>LowerThreshold</i>

Tool validations

* Step 2:

Message: **SetVariableMonitoringResponse** with (in arbitrary order):

```
setMonitoringResult[1] = {  
    - status = Accepted  
    - type = UpperThreshold  
    - statusInfo is absent or statusInfo.reasonCode = "NoError"  
}  
  
setMonitoringResult[2] = {  
    - status = Rejected  
    - type = Delta  
}  
  
setMonitoringResult[3] = {  
    - status = Accepted  
    - type = LowerThreshold  
    - statusInfo is absent or statusInfo.reasonCode = "NoError"  
}
```

Post scenario validations:

- N/a

6.2.99. Page 497 - (2025-02) - TC_N_45_CS - Updating test case for using more specific configuration variables

TC_N_45_CS: Alert Event - Delta value exceeded

Test case name	Alert Event - Delta value exceeded
Test case Id	TC_N_45_CS
Use case Id(s)	N07
Requirement(s)	N07.FR.06, N07.FR.07, N07.FR.18, N07.FR.19
System under test	Charging Station
Description	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.
Purpose	To verify if the Charging station is correctly communicating when a delta value has exceeded as described at the OCPP specification.
Prerequisite(s)	n/a

Before (Preparations)
Configuration State: N/a
Memory State: Variable monitor is configured with: - <code>setMonitoringData.component.name = <Configured threshold monitor component variable></code> - <code>setMonitoringData.component.evse.id = <Configured EVSEId></code> - <code>setMonitoringData.value = <Configured threshold monitor value></code> - <code>setMonitoringData.type = Delta</code> - <code>setMonitoringData.variable.name = <Configured delta monitor component variable></code> - <code>setMonitoringData.component = <Configured numeric delta component variable></code> - <code>setMonitoringData.variable = <Configured numeric delta component variable Delta numeric trigger value></code> - <code>setMonitoringData.value = <Configured numeric delta component variable></code> - <code>setMonitoringData.type = Delta</code> - <code>setMonitoringData.severity = 5</code>
Set MonitoringLevel to 8
<u>Notes:</u> If componentVariable is set to "Power" or "Current", the value is set to 100.0

Main (Test scenario)	
Charging Station	CSMS
<u>Manual Action:</u> If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.	
1. Execute Reusable State EnergyTransferStarted or manually trigger the monitor.	
2. The Charging Station sends a NotifyEventRequest	3. The Test System responds with a NotifyEventResponse .
<u>Note(s):</u> - If tbc is True at Step 2 then step 1 and 3 will be repeated	

Tool validations

* Step 2:

Message **NotifyEventRequest**

- **eventData[0].trigger** *Delta*
- **eventData[0].component.name** <Configured threshold monitor component variable>
- **eventData[0].variable.name** <Configured threshold monitor component variable>
- **eventData[0].variableMonitoringId** <Configured variableMonitoringId>
- **eventData[0].component** <Configured numeric delta component variable>
- **eventData[0].variable** <Configured numeric delta component variable>
- **eventData[0].variableMonitoringId** <SetVariableMonitoringResponse.setMonitoringResult.id in preparation phase>

Post scenario validations:

- N/a

6.2.100. Page 501 - (2025-02) - TC_N_51_CS - Updating test case for using more specific configuration variables

TC_N_51_CS: Set Variable Monitoring - Replace Variable Monitor **Modifying a VariableMonitor and trigger**

Test case name	Set Variable Monitoring – Replace Variable Monitor Set Variable Monitoring - Modifying a VariableMonitor and trigger
Test case Id	TC_N_51_CS
Use case Id(s)	N07
Requirement(s)	N07.FR.11
System under test	Charging Station
Description	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.
Purpose	To verify if the Charging station is able to correctly check if the current value exceeds the new threshold as described at the OCPP specification.
Prerequisite(s)	Charging Station supports Monitoring

Before (Preparations)
Configuration State: N/a
Memory State: Variable monitor is already set with: <code>setMonitoringData.component.name <Configured threshold monitor component variable> AND setMonitoringData.component.evse.id <Configured EVSEId> AND setMonitoringData.value <Configured threshold monitor value> AND setMonitoringData.type UpperThreshold AND setMonitoringData.variable.name <Configured threshold monitor component variable> setMonitoringData.component <Configured threshold monitor component variable> setMonitoringData.variable <Configured threshold monitor component variable> setMonitoringData.value <Configured threshold monitor component variable UpperThreshold non-trigger value> setMonitoringData.type UpperThreshold setMonitoringData.severity 5</code>
Set MonitoringLevel to 8
<u>Notes:</u> If componentVariable is set to "Power" or "Current", the value is set to the configured maxLimit -1

Reusable State: N/a	
Main (Test scenario)	
Charging Station	CSMS
<u>Notes:</u> If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.	
1. Execute Reusable State EnergyTransferStarted or manually trigger the monitor.	

Main (Test scenario)	
3. The Charging Station responds with a SetVariableMonitoringResponse	<p>2. The Test System sends a SetVariableMonitoringRequest with setMonitoringData.component.name <Configured threshold monitor component variable> AND setMonitoringData.component.evse.id <Configured EVSEId> AND setMonitoringData.id <Configured variableMonitoringId> AND setMonitoringData.value <Configured threshold monitor value2> AND setMonitoringData.type <i>UpperThreshold</i> setMonitoringData.variable.name <Configured threshold monitor component variable> setMonitoringData.component <Configured threshold monitor component variable> setMonitoringData.variable <Configured threshold monitor component variable> setMonitoringData.id <SetVariableMonitoringResponse.setMonitoringResult.id in preparation phase> setMonitoringData.value <Configured threshold monitor component variable UpperThreshold trigger value> setMonitoringData.type <i>UpperThreshold</i></p> <p><u>Notes:</u> If componentVariable is set to "Power" or "Current", the value is set to 0.0</p>
4. The Charging station sends a NotifyEventRequest	5. The Test System responds with a NotifyEventResponse .

Tool validations
<ul style="list-style-type: none"> * Step 3: <p>Message SetVariableMonitoringResponse</p> <ul style="list-style-type: none"> - setMonitoringResult[0].status Accepted - setMonitoringResult[0].type <i>UpperThreshold</i> - setMonitoringResult[0].severity <Configured severity> - setMonitoringResult[0].component.name <Configured threshold monitor component variable> - setMonitoringResult[0].variable.name <Configured threshold monitor component variable> - setMonitoringResult[0].severity 5 - setMonitoringResult[0].component <Configured threshold monitor component variable> - setMonitoringResult[0].variable <Configured threshold monitor component variable>
<ul style="list-style-type: none"> * Step 4: <p>Message NotifyEventRequest</p> <ul style="list-style-type: none"> - eventData[0].trigger Alerting - eventData[0].actualValue > <Configured threshold monitor value> - eventData[0].actualValue > <Configured threshold monitor component variable UpperThreshold non-trigger value>
<p>Post scenario validations:</p> <ul style="list-style-type: none"> - All report parts have been received

6.2.101. Page 503 - (2025-02) - TC_N_52_CS - Updating test case for using more specific configuration variables

TC_N_52_CS: Set Variable Monitoring - Removing a VariableMonitor

Test case name	Set Variable Monitoring - Removing a VariableMonitor
Test case Id	TC_N_52_CS
Use case Id(s)	N07
Requirement(s)	N07.FR.12
System under test	Charging Station
Description	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.
Purpose	To verify if the Charging station is able to correctly communicate when a threshold has been exceeded and the applicable monitor is removed as described at the OCPP specification.
Prerequisite(s)	Charging Station supports Monitoring

Before (Preparations)
Configuration State: N/a
Memory State: Variable monitor is already set with: <pre>setMonitoringData.component.name <Configured threshold monitor component variable> AND setMonitoringData.component.evse.id <Configured EVSEId> AND setMonitoringData.value <Configured threshold monitor value> AND setMonitoringData.type UpperThreshold AND setMonitoringData.variable.name <Configured threshold monitor component variable> setMonitoringData.component <Configured threshold monitor component variable> setMonitoringData.variable <Configured threshold monitor component variable> setMonitoringData.value <Configured threshold monitor component variable UpperThreshold trigger value> setMonitoringData.type UpperThreshold setMonitoringData.severity = 5</pre>
Set MonitoringLevel to 8
<u>Notes:</u> If componentVariable is set to "Power" or "Current", the value is set to 0.0
Reusable State: Execute Reusable State EnergyTransferStarted or manually trigger the monitor. <u>Notes:</u> If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a ClearVariableMonitoringResponse	1. The Test System sends a ClearVariableMonitoringRequest with id <Configured variableMonitoringId> id <SetVariableMonitoringResponse.setMonitoringResult.id in preparation phase>
4. The Charging Station responds with a GetMonitoringReportResponse	3. The Test System sends a GetMonitoringReportRequest with componentVariable.component <Configured threshold monitor component variable> componentVariable.variable <Configured threshold monitor component variable> monitoringCriteria ThresholdMonitoring

Main (Test scenario)
5. Execute Reusable State StopAuthorized or manually trigger the monitor. <u>Notes:</u> If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.
6. The Charging Station should not send a request for the cleared monitor

Tool validations
* Step 2: Message ClearVariableMonitoringResponse - clearMonitoringResult[0].status Accepted AND - clearMonitoringResult[0].id <Configured variableMonitoringId> - clearMonitoringResult[0].id <SetVariableMonitoringResponse.setMonitoringResult.id in preparation phase>
* Step 4: Message GetMonitoringReportResponse - getMonitoringResult[0].status EmptyResultSet
* Step 6: - No NotifyEventRequest with variableMonitoringId <Configured variableMonitoringId> is send - No NotifyEventRequest with variableMonitoringId <SetVariableMonitoringResponse.setMonitoringResult.id in preparation phase> is send
Post scenario validations: - N/a

6.2.102. Page 504 - (2025-02) - TC_N_53_CS - Updating test case for using more specific configuration variables

TC_N_53_CS: Alert Event - Persistant over reboot

Test case name	Alert Event - Persistant over reboot
Test case Id	TC_N_53_CS
Use case Id(s)	N07
Requirement(s)	N07.FR.13
System under test	Charging Station
Description	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.
Purpose	To verify if the Charging station is able to save the variableMonitor data persistent across reboot as described at the OCPP specification.
Prerequisite(s)	n/a

Before (Preparations)
Configuration State: N/a
Memory State: Variable monitor is already set with: <code>setMonitoringData.component.name <Configured threshold monitor component variable> AND</code> <code>setMonitoringData.component.evse.id <Configured EVSEId> AND</code> <code>setMonitoringData.value <Configured threshold monitor value> AND</code> <code>setMonitoringData.type UpperThreshold AND</code> <code>setMonitoringData.variable.name <Configured threshold monitor component variable></code> <code>setMonitoringData.component <Configured threshold monitor component variable></code> <code>setMonitoringData.variable <Configured threshold monitor component variable></code> <code>setMonitoringData.value <Configured threshold monitor component variable UpperThreshold trigger value></code> <code>setMonitoringData.type UpperThreshold</code>
Reusable State: Execute Reusable State Booted

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a GetMonitoringReportResponse	1. The Test System sends a GetMonitoringReportRequest with monitoringCriteria ThresholdMonitoring
3. The Charging Station sends a NotifyMonitoringReportRequest	4. The Test System responds with a NotifyMonitoringReportResponse .
Note(s): - If tbc is True at Step 3 then step 3 and 4 will be repeated	

Tool validations
* Step 3: Message NotifyMonitoringReportRequest - requestId <The Id of the request> AND - monitor.variableMonitoring.id <Received monitorId from set monitor> - monitor.variableMonitoring.id <SetVariableMonitoringResponse.setMonitoringResult.id in preparation phase> - monitor.variableMonitoring.type UpperThreshold
Post scenario validations: - All reports have been received

6.2.103. Page 505 - (2025-02) - TC_N_56_CS - Made test case configurable using configuration variables

Before (Preparations)

Configuration State:

N/a

Memory State:

Variable monitor is configured with:

component.evse.id <Configured EVSEId>
component.name EVSE
severity <Configured severity>
type Delta
value 1.0
variable.name AvailabilityState

Variable monitor is configured with:

- **setMonitoringData[0].value** = 1
- **setMonitoringData[0].type** = Delta
- **setMonitoringData[0].severity** = 5
- **setMonitoringData[0].component** = <Configured non-numeric delta component variable>
- **setMonitoringData[0].variable** = <Configured non-numeric delta component variable>

Set MonitoringLevel to 8

Notes :

- Take a non-numeric component variable which can easily modified to trigger the alert.

Reusable State:

N/a

Tool validations

* Step 1:

Message NotifyEventRequest

- **eventData[0].trigger** Delta
- **eventData[0].component.name** EVSE
- **eventData[0].variable.name** AvailabilityState
- **eventData[0].component** <Configured non-numeric delta component variable>
- **eventData[0].variable** <Configured non-numeric delta component variable>
- **eventData[0].variableMonitoringId** monitoringId of monitor set in Memory State

Post scenario validations:

- N/a

6.2.104. Page 520 - (2025-02) - TC_O_15_CS - Test case removed

This test case has been removed.

Test case name	Set Display Message - Language preference of the EV Driver
Test case Id	TC_O_15_CS
...	...

6.2.105. Page 530 - (2025-02) - TC_O_28_CS - Transaction id should be specified for DisplayMessage

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetDisplayMessageResponse	1. The Test System sends a SetDisplayMessageRequest with message.id <Generated displayMessageId> message.transactionId <Generated transactionId> message.priority <Configured Priority> message.endDateTime <Current dateTIme + 60 seconds>
[...]	

6.2.106. Page 533 - (2025-02) - TC_O_32_CS - Made notes about display behaviour more explicit

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetDisplayMessageResponse	1. The Test System sends a SetDisplayMessageRequest with message.id <Generated displayMessageId> message.transactionId <Received transactionId> AND message.priority AlwaysFront
Note(s): - Display message <Generated displayMessageId> is shown	
4. The Charging Station responds with a SetDisplayMessageResponse	3. The Test System sends a SetDisplayMessageRequest with message.id <Configured displayMessage2Id> message.transactionId <Received transactionId> AND message.priority AlwaysFront
Note(s): - Display message <Generated displayMessage1Id> is not displayed anymore - Display message <Generated displayMessage2Id> is shown	
6. The Charging Station responds with a GetDisplayMessagesResponse	5. The Test System sends a GetDisplayMessagesRequest with id <Generated displayMessageId>
8. The Charging Station responds with a GetDisplayMessagesResponse	7. The Test System sends a GetDisplayMessagesRequest with id <Generated displayMessageId2> requestId <Generated requestId>
9. The Charging Station sends a NotifyDisplayMessagesRequest	10. The Test System responds with a NotifyDisplayMessagesResponse .
11. Execute Reusable State <i>StopAuthorized</i>	
12. Execute Reusable State <i>EVConnectedPostSession</i>	
13. Execute Reusable State <i>EVDDisconnected</i>	
14. Execute Reusable State <i>ParkingBayUnoccupied</i>	

Main (Test scenario)	
<u>Note(s):</u> - The display message is not displayed anymore - Display message <Generated displayMessage2Id> is not displayed anymore	
16. The Charging Station responds with a GetDisplayMessagesResponse	15. The Test System sends a GetDisplayMessagesRequest with id <Generated displayMessageId2>

6.2.107. Page 544 - (2025-02) - TC_O_39_CS - Wait for StatusNotificationRequest or NotifyEventRequest

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetDisplayMessageResponse	1. The Test System sends a SetDisplayMessageRequest with message.id <Generated displayMessageId> message.priority <Configured Priority> message.state <Configured State> message.message Faulted
<u>Note(s):</u> The display message should NOT be displayed.	
<u>Manual Action:</u> Set the Charging Station to state Faulted.	
3. The Charging Station notifies the CSMS about the status change of the Charging Station.	4. The Test System responds accordingly.
<u>Note(s):</u> The display message should be displayed now. - Step 3/4 are used to detect if the Charging Station status has changed. - The display message should be displayed now.	
<u>Manual Action:</u> Set the Charging Station back to state Available.	
5. The Charging Station notifies the CSMS about the status change of the Charging Station.	6. The Test System responds accordingly.
<u>Note(s):</u> The display message should NOT be displayed anymore. - Step 5/6 are used to detect if the Charging Station status has changed. - The display message should NOT be displayed anymore.	
8. The Charging Station responds with a GetDisplayMessagesResponse	7. The Test System sends a GetDisplayMessagesRequest with id <Generated displayMessageId> requestId <Generated requestId>
9. The Charging Station sends a NotifyDisplayMessagesRequest	10. The Test System responds with a NotifyDisplayMessagesResponse .
<u>Note(s):</u> If tbc is True at Step 9 then step 9 and 10 will be repeated	

Tool validations

* Step 2:

Message **SetDisplayMessageResponse**

- **status Accepted**

* Step 3:

At least one of the following messages must be sent:

Message: **StatusNotificationRequest**

- **connectorStatus Faulted or Unavailable**

Message: **NotifyEventRequest**

- **eventData[0].trigger must be Delta**

- **eventData[0].actualValue must be Faulted or Unavailable**

- **eventData[0].component.name must be Connector**

- **eventData[0].variable.name must be AvailabilityState**

- **evse.id <not omitted>**

- **connector.id <not omitted>**

Message: **NotifyEventRequest**

- **eventData[0].trigger must be Delta**

- **eventData[0].actualValue must be Faulted**

- **eventData[0].component.name must be ChargingStation**

- **eventData[0].variable.name must be AvailabilityState**

- **evse.id <omitted>**

- **connector.id <omitted>**

* Step 5:

At least one of the following messages must be sent:

Message: **StatusNotificationRequest**

- **connectorStatus Available**

Message: **NotifyEventRequest**

- **eventData[0].trigger must be Delta**

- **eventData[0].actualValue must be Available**

- **eventData[0].component.name must be Connector**

- **eventData[0].variable.name must be AvailabilityState**

- **evse.id <not omitted>**

- **connector.id <not omitted>**

Message: **NotifyEventRequest**

- **eventData[0].trigger must be Delta**

- **eventData[0].actualValue must be Available**

- **eventData[0].component.name must be ChargingStation**

- **eventData[0].variable.name must be AvailabilityState** + * Step 8:

Message **GetDisplayMessagesResponse**

- **status Accepted**

* Step 9:

Message **NotifyDisplayMessagesRequest**

- **requestId <Generated requestId>**

- **state Faulted**

Post scenario validations:

- N/a

6.2.108. Page 555 - (2024-11) - Remove StatusNotificationRequest from Authorized reusable state Main B steps

Main B (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a RequestStartTransactionResponse	1. The OCTT sends a RequestStartTransactionRequest with idToken.idToken <Configured valid_idtoken_idtoken> idToken.type <Configured valid_idtoken_type> evseld <Configured evseld>
3. The Charging Station sends an AuthorizeRequest	4. The OCTT responds with an AuthorizeResponse with idTokenInfo.status Accepted
<p><u>Note(s):</u></p> <p>- This step needs to be executed when AuthCtrlr.AuthorizeRemoteStart is true, unless (AuthEnabled is implemented with mutability ReadOnly AND the value is set to false) OR the idToken is cached. In case the idToken is used for a reservation, sending the AuthorizeRequest message is optional.</p>	
<Removed>	
5. The Charging Station sends a TransactionEventRequest	6. The OCTT responds with a TransactionEventResponse
<p><u>Note(s):</u></p> <p>- This step needs to be executed when TxStartPoint contains Authorized OR the transaction already started. So in the case TxStartPoint contains ParkingBayOccupancy or (EVConnected, in the case this testcase was initiated from state EVConnectedPreSession.)</p>	
<p><u>Note(s):</u></p> <p>- The first TransactionEventRequest sent after authorization contains the idToken field. The TransactionEventResponse of this request message contains idTokenInfo with status Accepted</p>	

6.2.109. Page 560 - (2025-04) - Reusable states StopAuthorized & Deauthorized

Based on TWG and CWG discussions the transaction validations of the Test system have been made more flexible and validate accordingly based on the different TxStartPoint and TxStopPoint combinations. However, the reusable states described at part 6 did not reflect this yet.

State	StopAuthorized
System under test	Charging Station
Description	This state will prepare the Charging Station, so that it is in a state where the charging session is authorized to stop. This can be done in two ways (Configurable at Test System): A. Using local authorization B. Using a RequestStopTransactionRequest

Main A (Test scenario)	
Charging Station	CSMS
<u>Notes(s):</u> The tool will wait for <Configured Transaction Duration> seconds	
<u>Manual Action:</u> Present the same idToken as used to start the transaction.	
1. The Charging Station sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse With idTokenInfo.status is Accepted
<u>Note(s):</u> This step is optional	
3. The Charging Station sends a TransactionEventRequest	4. The Test System responds with a TransactionEventResponse With idTokenInfo.status is Accepted

Tool validations

* Step 1:

Message: **TransactionEventRequest**

- **triggerReason** must be *StopAuthorized*

- **idToken** omit OR

- **idToken.idToken** <Configured valid_idtoken_idtoken> AND

- **idToken.type** <Configured valid_idtoken_type>

* Step 3:

Message: **TransactionEventRequest**

- **triggerReason** must be *ChargingStateChanged*

- **transactionInfo.chargingState** must be *EVConnected*

If TxStopPoint contains Authorized or PowerPathClosed or EnergyTransfer

Then the last of the two **TransactionEventRequest** messages from step 1 and 3 needs to contain:

- **eventType** must be *Ended*

- **transactionInfo.stoppedReason** must be *Local* or omitted

Else

Then both **TransactionEventRequest** messages need to contain:

- **eventType** must be *Updated*

Main B (Test scenario)

Charging Station	CSMS
2. The Charging Station responds with a RequestStopTransactionResponse	1. The Test System sends a RequestStopTransactionRequest with transactionId <transactionId provided by the Charging Station in TransactionEventRequest >
3. The Charging Station sends a TransactionEventRequest	4. The Test System responds with a TransactionEventResponse
Note(s): This step is optional	
5. The Charging Station sends a TransactionEventRequest	6. The Test System responds with a TransactionEventResponse With idTokenInfo.status is Accepted

Tool validations

* Step 2:

Message: **RequestStopTransactionResponse**

- **status** must be *Accepted*

* Step 3:

Message: **TransactionEventRequest**

- **triggerReason** must be *RemoteStop*

* Step 5:

Message: **TransactionEventRequest**

- **triggerReason** must be *ChargingStateChanged*

- **transactionInfo.chargingState** must be *EVConnected*

If TxStopPoint contains Authorized or PowerPathClosed or EnergyTransfer

Then the last of the two **TransactionEventRequest** messages from step 3 and 5 needs to contain:

- **eventType** must be *Ended*

- **transactionInfo.stoppedReason** must be *Remote*

Else

Then both **TransactionEventRequest** messages need to contain:

- **eventType** must be *Updated*

Post scenario validations:

State is *StopAuthorized*

New reusable state Deauthorized. This is not a change to the testcases it already existed on the background, but is now formally defined at part 6.

6.2.110. Deauthorized

State	Deauthorized
System under test	Charging Station
Description	This reusable state will set the Charging Station to a state in which the transaction will be deauthorized.
Prerequisite	Reusable State Authorized is executed.

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): Continues executing transaction reusable states in order until the first TransactionEventRequest is received based on the configured TxStartPoint.

Main (Test scenario)	
Charging Station	CSMS
1. The Charging Station sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse With idTokenInfo.status is <i>Invalid</i>
3. The Charging Station sends a TransactionEventRequest	4. The Test System responds with a TransactionEventResponse
Note(s): - Step 3 and 4 are relevant when AuthCtrlr.StopTxOnInvalidId is true	
6. The Charging Station sends a UnlockConnectorResponse	5. The Test System sends a UnlockConnectorRequest Note(s): - Step 5 and 6 are executed when the connector is locked and the transaction gets deauthorized.

Tool validations
<ul style="list-style-type: none"> * Step 1: Message: TransactionEventRequest - idToken.idToken <Configured valid_idtoken_idtoken> AND - idToken.type <Configured valid_idtoken_type> * Step 3: Message: TransactionEventRequest - triggerReason must be <i>Deauthorized</i> - transactionInfo.chargingState must be <i>EVConnected</i> or <i>SuspendedEVSE</i> If TxStopPoint contains <i>Authorized</i> or <i>PowerPathClosed</i> or <i>EnergyTransfer</i> Then : - eventType must be <i>Ended</i> - stoppedReason must be <i>DeAuthorized</i> Else: - eventType must be <i>Updated</i>

6.2.111. Page 566 - (2025-02) - Reusable state RenegotiateChargingLimits

Main (Test scenario)	
Charging Station	CSMS
[...]	
7. The Charging Station sends a TransactionEventRequest	8. The Test System responds with a TransactionEventResponse
Note: Steps 7 and 8 are optional, but can also repeat until chargingState is Charging.	

Tool validations
[...]
* Step 5: Message: NotifyEVChargingScheduleRequest) - evsId <Configured evsId>
* Step 7: Message: TransactionEventRequest - triggerReason must be <i>ChargingStateChanged</i> - transactionInfo.chargingState must be <i>Charging</i> - evsId <Configured evsId>
Post scenario validations: N/a

6.2.112. Page 573/151 - (2025-04) - Removed Main steps B from IdTokenCached reusable state and added IdTokenCached15118

The described steps at step B were incorrect and the originally intended split does not exist in any of the currently existing testcases.

In addition, a separate IdTokenCached15118 reusable state has been added for ISO 15118 sessions. The affected testcases are TC_C_54_CS and TC_C_55_CS.

IdTokenCached

State	IdTokenCached
System under test	Charging Station
Description	An idToken is stored in the Authorization Cache of the Charging Station.

Before (Preparations)	
Configuration State:	N/a
Memory State:	N/a
Reusable State(s):	N/a

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>ParkingBayoccupied</i>	
2. Execute Reusable State <i>Authorized</i>	
Note(s): Step 3 and onwards are executed in case the idToken at step 2 was Accepted.	
3. Execute Reusable State <i>EVConnectedPreSession</i>	
4. Execute Reusable State <i>EnergyTransferStarted</i>	
5. Execute Reusable State <i>StopAuthorized</i>	
6. Execute Reusable State <i>EVDisconnected</i>	
7. Execute Reusable State <i>ParkingBayUnoccupied</i>	

Tool validations
N/a

IdTokenCached15118

State	IdTokenCached15118
System under test	Charging Station
Description	A 15118-idToken is stored in the Authorization Cache of the Charging Station.

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>ParkingBayoccupied</i>	
2. Execute Reusable State <i>EVConnectedPreSession</i>	
3. Execute Reusable State <i>Authorized15118</i>	
4. Execute Reusable State <i>EnergyTransferStarted</i>	
5. Execute Reusable State <i>StopAuthorized (Remote)</i>	
6. Execute Reusable State <i>EVDisconnected</i>	
7. Execute Reusable State <i>ParkingBayUnoccupied</i>	

Tool validations
N/a

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is true
Test case Id	TC_C_54_CS
...	

Before (Preparations)
Configuration State: ...
Memory State: <i>CertificateInstalled</i> for certificateType V2GRootCertificate <i>CertificateInstalled</i> for certificateType MORootCertificate <i>RenewV2GChargingStationCertificate</i> (If none are present, when checking with GetInstalledCertificates.certificateType = V2GCertificateChain) <i>IdTokenCached15118</i> for <Configured valid ISO 15118 IdToken> (If implemented) <i>IdTokenLocalAuthList</i> for <Configured valid ISO 15118 IdToken> (If implemented)
Reusable State(s): N/a

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is false
Test case Id	TC_C_55_CS
...	

Before (Preparations)
Configuration State:
...
Memory State:
<p><i>CertificateInstalled</i> for certificateType <i>V2GRootCertificate</i></p> <p><i>CertificateInstalled</i> for certificateType <i>MORootCertificate</i></p> <p><i>RenewV2GChargingStationCertificate</i> (If none are present, when checking with GetInstalledCertificates.certificateType = <i>V2GCertificateChain</i>)</p> <p><i>IdTokenCached15118</i> for <Configured valid ISO 15118 IdToken> (If implemented)</p> <p><i>IdTokenLocalAuthList</i> for <Configured valid ISO 15118 IdToken> (If implemented)</p>
Reusable State(s):
N/a

6.2.113. Page 574 - (2025-04) - **IdTokenLocalAuthList** memory state - set Enable to true if implemented

State	IdTokenLocalAuthList
System under test	Charging Station
Description	An valid idToken is stored in the Local Authorization List of the Charging Station.

Before (Preparations)
Configuration State:
<i>LocalAuthListCtrlr.Enabled</i> is <i>true</i> (If implemented)
Memory State:

N/a

Reusable State(s):

N/a

6.2.114. Page 575 - (2024-09) - Reusable state **RenewChargingStationCertificate** expects a reconnection [784]

If a valid certificate is installed, then charging station must use it. This involves reconnecting to set up a new TLS with the new certificate. If the charging station does not do so automatically, then OCTT will force it by sending a Reset command.

State	RenewChargingStationCertificate
System under test	Charging Station
Description	The ChargingStationCertificate is renewed using A02/A03
...	

State	RenewChargingStationCertificate	
Main (Test scenario)	Charging Station	CSMS
	2. The Charging Station responds with a TriggerMessageResponse	1. The OCTT sends a TriggerMessageRequest With requestedMessage <i>SignChargingStationCertificate</i>
	3 The Charging Station sends a SignCertificateRequest	4. The OCTT responds with a SignCertificateResponse With status Accepted
	6. The Charging Station responds with a CertificateSignedResponse	5. The OCTT sends a CertificateSignedRequest With certificateChain <Certificate generated from the received CSR from step 3 and signed by the provided CSMS Root certificate> certificateType ChargingStationCertificate
	<i>If the certificate is valid, then Charging Station should reconnect with the new certificate. OCTT waits some time for a reconnection, and if that does not occur, will send a Reset command to Charging Station to force a reconnection.</i>	
	7. The Charging Station reconnects.	
	8 . If the reconnect was forced by a Reset: The Charging Station sends a BootNotificationRequest	9. OCTT responds with a BootNotificationResponse .
Tool validations	<ul style="list-style-type: none"> * Step 2: Message: TriggerMessageResponse - status must be Accepted * Step 3: Message: SignCertificateRequest - csr must contain <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.> * Step 6: Message: CertificateSignedResponse - status must be Accepted * Step 7: <i>Charging Station must reconnect with new certificate.</i> <p>Post scenario validations: N/a</p>	

6.2.115. Page 575 - (2025-02) - Reusable state RenewChargingStationCertificate must not do a Reset [5281]

This memory state was sending a Reset when Charging Station was dropping the connection in time, but according to A02.FR.08 a Reset is not needed. It is enough to just close the websocket connection.

State	RenewChargingStationCertificate
System under test	Charging Station
Description	The ChargingStationCertificate is renewed using A02/A03
...	

State	RenewChargingStationCertificate	
Main (Test scenario)	Charging Station	CSMS

	6. The Charging Station responds with a CertificateSignedResponse	5. The OCTT sends a CertificateSignedRequest With certificateChain <Certificate generated from the received CSR from step 3 and signed by the provided CSMS Root certificate> certificateType ChargingStationCertificate <i>If the certificate is valid, then Charging Station should reconnect with the new certificate. Test System waits some time for a reconnection, and if that does not occur, will drop the connection to force a reconnection.</i>
	7. The Charging Station reconnects.	
	8. If Charging Station rebooted: The Charging Station sends a BootNotificationRequest	9. Test System responds with a BootNotificationResponse .
	...	
	Post scenario validations: N/a	

6.3. CSMS

6.3.1. Page 593 - (2025-04) - TC_A_11_CSMS - Added post scenario validation for clarification

Note: This erratum extends erratum: [Page 593 - \(2024-09\) - TC_A_11_CSMS - Reconnect using new client certificate](#)

Test case name	Update Charging Station Certificate by request of CSMS - Success - Charging Station Certificate
Test case Id	TC_A_11_CSMS
...	

Tool validations
N/a
Post scenario validations: The Test System and the CSMS are connected.

6.3.2. Page 593 - (2024-09) - TC_A_11_CSMS - Reconnect using new client certificate

Note: This erratum is extended by erratum: [Page 593 - \(2025-04\) - TC_A_11_CSMS - Added post scenario validation for clarification](#)

The testcase is missing steps to reconnect using the new client certificate.

Test case name	Update Charging Station Certificate by request of CSMS - Success - Charging Station Certificate
Test case Id	TC_A_11_CSMS
...	...

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State RenewChargingStationCertificate	
2. The OCTT disconnects its current connection and reconnects to the CSMS with the new certificate.	3. The CSMS accepts the incoming connection request using the new certificate.

6.3.3. Page 596 - (2024-09) - TC_A_14_CSMS - Update Charging Station Certificate by request of CSMS - Invalid certificate

SecurityEventNotification(InvalidChargingStationCertificate) has been added.

Test case name	Update Charging Station Certificate by request of CSMS - Invalid certificate
Test case Id	TC_A_14_CSMS
...	

Test case name	Update Charging Station Certificate by request of CSMS - Invalid certificate	
Main (Test scenario)	<p>Charging Station</p> <p>2. The OCTT responds with a TriggerMessageResponse With status Accepted</p> <p>3 The OCTT sends a SignCertificateRequest With csr <Configured CSR> certificateType ChargingStationCertificate</p> <p>6. The OCTT responds with a CertificateSignedResponse With status Rejected</p> <p>7. The OCTT sends a SecurityEventNotificationRequest with type = InvalidChargingStationCertificate</p>	<p>1. The CSMS sends a TriggerMessageRequest</p> <p>4. The CSMS responds with a SignCertificateResponse</p> <p>5. The CSMS sends a CertificateSignedRequest</p> <p>8. The CSMS responds with a SecurityEventNotificationResponse</p>
Tool validations	...	

6.3.4. Page 597 - (2025-04) - TC_A_19_CSMS - Added main steps and clarified tool validations

Clarified the validation steps regarding how the CSMS rejects the connection and added reconnect at the end of the testcase, so it does not end without a connection.

Test case name	Upgrade Charging Station Security Profile - Accepted	
Test case Id	TC_A_19_CSMS	
...		

Main (Test scenario)		
Charging Station	CSMS	
<u>Manual Action:</u> Request the CSMS to set a new NetworkConnectionProfile with a security profile level one higher than currently configured		
2. The Test System responds with a SetNetworkProfileResponse With status Accepted	1. The CSMS sends a SetNetworkProfileRequest	
<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 Test System responds with a SetVariablesResponse with status Accepted	3. The CSMS sends a SetVariablesRequest	
<u>Manual Action:</u> Request the CSMS to reboot the Charging Station		
6. The Test System responds with a ResetResponse with status Accepted	5. The CSMS sends a ResetRequest	
7. The Test System reconnects to the CSMS using the new NetworkProfile, containing the upgraded security profile <Configured securityProfile + 1>.	8. The CSMS accepts the connection attempt.	
9. Execute Reusable State <i>Booted</i>		
10. The Test System reconnects to the CSMS using the original NetworkProfile, containing the lower security profile.	11. The CSMS shall not accept the connection attempt.	
<u>Note(s):</u> - This is done to ensure that the CSMS does not accept a connection using the lower security profile anymore.		

Main (Test scenario)	
<p>12. The Test System reconnects to the CSMS using the new NetworkProfile, containing the upgraded security profile <Configured securityProfile + 1></p> <p>Note(s):</p> <ul style="list-style-type: none"> - This is done to restore the connection before ending the testcase. 	13. The CSMS accepts the connection attempt.
Tool validations	

* Step 1:

Message **SetNetworkProfileRequest**

- **connectionData.messageTimeout** <Configured messageTimeout>
- **connectionData.ocppInterface** <Configured ocppInterface>
- **connectionData.ocppTransport** JSON
- **connectionData.ocppVersion** OCPP20
- **connectionData.securityProfile** <Configured securityProfile + 1>

* Step 3:

Message **SetVariablesRequest**

setVariableData:

- **variable.name** = "NetworkConfigurationPriority"
- **component.name** = "OCPPCommCtrlr"
- **attributeValue** = <contains configurationSlot provided at step 1>

* Step 11:

When upgrading a Charging Station to a higher security profile, a CSMS has several options regarding which endpoint to use. This affects the way the CSMS is able to detect it needs to reject the incoming connection attempt.

In case of having upgraded from security profile 2 to 3, but there is an incoming connection attempt using security profile 2:

When the same endpoint is used, then it depends on the CSMS endpoint configuration.

- When the CSMS does a full switch and only allows TLS handshakes when a client certificate is provided, then the TLS handshake is rejected.
- When the CSMS only requires this Charging Station to use a client certificate, then it accepts the TLS handshake (because it will be unable to detect which Charging Station is connecting) and it rejects the HTTP request to establish the WebSocket connection.

When a different port or a whole different endpoint is used for the upgrade, then on the original endpoint the CSMS accepts the TLS handshake and it rejects the HTTP request to establish the WebSocket connection (because this Charging Station is not allowed to connect with security profile 2 anymore).

In case of security profile 1, the case is always the same. The CSMS shall always reject the HTTP request to establish the WebSocket connection, because TLS is required for this Charging Station.

Post scenario validations:

The Test System and the CSMS are connected.

6.3.5. Page 597 - (2024-09) - TC_A_19_CSMS - Added additional information regarding the use of the client certificates

Added additional information regarding the use of the client certificates.

Test case name	Upgrade Charging Station Security Profile - Accepted
Test case Id	TC_A_19_CSMS
...	...

Before (Preparations)
Configuration State: N/a
Memory State: If configured <Security profile> is 2, then RenewChargingStationCertificate The OCTT uses this certificate during the TLS handshake when connecting with security profile 3.
Reusable State(s): N/a

6.3.6. Page 597 - (2024-09) - TC_A_19_CSMS - Removed validation of OcppCsmsUrl [O20-4355]

Validation of OcppCsmsUrl has been removed, because in some implementations the URL changes with the security profile.

Test case name	Upgrade Charging Station Security Profile - Accepted	
Test case Id	TC_A_19_CSMS	
...		
Main (Test scenario)	Charging Station	CSMS

Tool validations	<ul style="list-style-type: none"> * Step 1: Message SetNetworkProfileRequest <ul style="list-style-type: none"> - connectionData.messageTimeout <Configured messageTimeout> - connectionData.ocppCsmsUrl <Configured ocppCsmsUrl> - connectionData.ocppInterface <Configured ocppInterface> - connectionData.ocppTransport JSON - connectionData.ocppVersion OCPP20 - connectionData.securityProfile <Configured securityProfile + 1> * Step 3: Message SetVariablesRequest setVariableData: <ul style="list-style-type: none"> - variable.name = "NetworkConfigurationPriority" - component.name = "OCPPCommCtrlr" - attributeValue = <contains configurationSlot provided at step 1> 	
	Post scenario validations: - N/a	

6.3.7. Page 637 - (2024-11) - TC_C_50_CSMS - Changed reference to configured valid idToken to a specific eMAID idToken

For ISO 15118 plug & charge the Charging Station always needs to use an eMAID idToken, which equals the CN of the configured contract certificate.

Test case name	Authorization using Contract Certificates 15118 - Online - Local contract certificate validation - Accepted
Test case Id	TC_C_50_CSMS
Use case Id(s)	C07
Requirement(s)	C07.FR.04
System under test	CSMS
Description	The Charging Station is able to authorize with contract certificates when it supports ISO 15118.
Purpose	To verify if the CSMS is able to validate the certificate hash data and the provided eMAID.

Test case name	Authorization using Contract Certificates 15118 - Online - Local contract certificate validation - Accepted
Prerequisite(s)	<ul style="list-style-type: none"> - The configured eMAID is known by the CSMS as valid. - The configured contract certificate is valid. - The CN of the configured contract certificate equals the configured eMAID. - iso15118CertificateHashData has a responder URL that points to an OCSP service for OCTT. - CSMS does not have a cached OCSP response for the contract certificate.

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): State is <i>EVConnectedPreSession</i>

Main (Test scenario)	
Charging Station	CSMS
<p>1. The OCTT sends an AuthorizeRequest With idToken.idToken <Configured eMAID> idToken.type eMAID iso15118CertificateHashData contains <hashes from configured (V2G) certificate chain</p>	...
...	

6.3.8. Page 639 - (2024-09) - TC_C_52_CSMS - TC does not use <Configured contract_certificate>

OCTT already has a keystore that contains the certificate. The pdf should not mention the <Configured contract_certificate> as the testcase does not use it

Test case name	Authorization using Contract Certificates 15118 - Online - Central contract certificate validation - Accepted	
Test case Id	TC_C_52_CSMS	
...		
Prerequisite(s)	<ul style="list-style-type: none"> - The configured eMAID is known by the CSMS as valid. - The configured contract certificate is signed by the configured V2GRoot or MORoot certificate at the CSMS. - Contract certificate has a responder URL that points to an OCSP service for OCTT. - CSMS does not have a cached OCSP response for the contract certificate. 	
...		
Main (Test scenario)	Charging Station	CSMS
	<p>1. The OCTT sends an AuthorizeRequest With idToken.idToken <Configured valid_idtoken_idtoken> idToken.type <Configured valid_idtoken_type> iso15118CertificateHashData is absent certificate <i>from keystore</i></p>	<p>2. The CSMS sends an OCSP request to responder URL of certificate to check validity</p>
...		

6.3.9. Page 639 - (2025-02) - TC_C_52_CSMS - Certificate needs at least one subCA

Test case name	Authorization using Contract Certificates 15118 - Online - Central contract certificate validation - Accepted
Test case Id	TC_C_52_CSMS
...	...

Before (Preparations)
...

Main (Test scenario)	
Charging Station	CSMS
...	...

Tool validations
* Step 2: CSMS sends an OCSP request for certificate
* Step 3: Test System checks that received request for certificate is valid AND key type = ECDSA AND certificate chain contains at least one subCA
* Step 4: Message: AuthorizeResponse
- idTokenInfo.status Accepted
- certificateStatus Accepted
* Step 6: Message: TransactionEventResponse
- idTokenInfo.status Accepted
Post scenario validations:
N/a

6.3.10. Page 640 - (2025-04) - TC_D_01_CSMS - Missing tool validation that the **idTokenInfo** must be provided for all list entries

Test case name	Send Local Authorization List - Full
Test case Id	TC_D_01_CSMS
...	

Tool validations
* Step 1: Message SendLocalListRequest
- updateType Full
- versionNumber <Bigger than 0>
- localAuthorizationList <Not empty>
- localAuthorizationList[n].idTokenInfo <Not empty>
Post scenario validations:
- N/a

6.3.11. Page 712 - (2024-09) - TC_I_01_CSMS - Show EV Driver running total cost

Test case name	Show EV Driver running total cost during charging - costUpdatedRequest
Test case Id	TC_I_01_CSMS
...	

Test case name	Show EV Driver running total cost during charging - costUpdatedRequest	
Main (Test scenario)	<p>Charging Station</p> <p>...</p> <p>7. The OCTT sends a TransactionEventRequest With triggerReason is <i>MeterValuePeriodic</i> eventType is <i>Updated</i> timestamp <The intervals between the timestamps of the received Meter Value messages equals the configured sampled Meter Values interval> sampledValue.context is <i>Sample.Periodic</i></p> <p><u>Note(s):</u> - This step will be executed every <Configured sampled Meter Values interval> - The OCTT will end the testcase after two MeterValues.</p> <p>...</p>	<p>8. The CSMS responds with a TransactionEventResponse</p>
Tool validations	<p>...</p> <p>Post scenario validations: - N/a</p>	

6.3.12. Page 715 - (2025-02) - TC_I_02_CSMS - Added explicit information about CSMS tariff configuration and sending in needed metervalue

Before (Preparations)
Configuration State: N/a
Memory State: N/a - CSMS is configured with a tariff which is based on energy consumed.
Reusable State(s): state is <i>EVConnectedPostSession</i> - N/a

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>EVConnectedPreSession</i>	
- The TransactionEventRequest contains the MeterValue field. - sampledValue[0].value 1000 - sampledValue[0].context <i>Transaction.Begin</i>	
2. Execute Reusable State <i>Authorized</i>	
3. Execute Reusable State <i>EnergyTransferStarted</i>	
4. Execute Reusable State <i>StopAuthorized</i>	
5. Execute Reusable State <i>EVConnectedPostSession</i>	

Main (Test scenario)	
6. The Test System notifies the CSMS about the current state of the configured connector.	7. The CSMS responds accordingly.
<p>Message: StatusNotificationRequest</p> <ul style="list-style-type: none"> - connectorStatus Available <p>Message: NotifyEventRequest</p> <ul style="list-style-type: none"> - trigger Delta - actualValue "Available" - component.name "Connector" - variable.name "AvailabilityState" 	
8. The Test System sends a TransactionEventRequest with <ul style="list-style-type: none"> - triggerReason EVCommunicationLost - eventType Ended - transactionInfo.chargingState Idle - transactionInfo.stoppedReason EVDisconnected - meterValue[0].sampledValue[0].value 6000 - meterValue[0].sampledValue[0].context Transaction.End 	9. The CSMS responds with a TransactionEventResponse

6.3.13. Page 726-760 - (2025-04) - TC_K_XX_CSMS - Improved tool validations to be sure valid Charging Profiles are used

Test case name	Set Charging Profile - TxDefaultProfile - Specific EVSE
Test case Id	TC_K_01_CSMS
...	

Tool validations	
* Step 1:	
Message SetChargingProfileRequest evseld <Configured evseld> AND chargingProfile.stackLevel <Configured stackLevel> AND chargingProfile.chargingProfilePurpose TxDefaultProfile AND chargingProfile.chargingProfileKind Absolute AND chargingProfile.validFrom now AND chargingProfile.validTo now + <Configured Charging Schedule Duration> AND chargingProfile.chargingSchedule.startSchedule now AND chargingProfile.chargingSchedule.chargingRateUnit <Configured chargingRateUnit> AND chargingProfile.chargingSchedule.duration <Configured duration> AND chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0 AND chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6.0 or 6000.0 AND chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases> where <Configured numberPhases> not 3 OR chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases> or <omit> where <Configured numberPhases> 3	
Post scenario validations: - N/a	

Test case name	Set Charging Profile - TxProfile without ongoing transaction on the specified EVSE
Test case Id	TC_K_02_CSMS
...	

Tool validations

* Step 1:

Message **SetChargingProfileRequest**

- **evsId** <Configured evsId> AND
- **chargingProfile.chargingProfilePurpose** *TxProfile* AND
- **chargingProfile.stackLevel** <Configured stackLevel> AND
- **chargingProfile.chargingProfileKind** *Relative* AND
- **chargingProfile.chargingSchedule.startSchedule** must be omitted AND
- **chargingProfile.chargingSchedule.chargingRateUnit** <Configured chargingRateUnit> AND
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod** 0 AND
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.limit** 7.0 or 7000.0 AND
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases** <Configured numberPhases> where <Configured numberPhases> not 3 OR
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases** <Configured numberPhases> or <omit> where <Configured numberPhases> 3

Post scenario validations:

- N/a

Test case name	Replace charging profile - With chargingProfileId
Test case Id	TC_K_04_CSMS
...	

Tool validations

* Step 1:

Message **SetChargingProfileRequest**

chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0 AND
chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 8.0 (A) or 8000.0 (W)

The **chargingSchedule** contains only one **chargingSchedulePeriod**

* Step 3:

Message **SetChargingProfileRequest**

chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0 AND
chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6.0 (A) or 6000.0 (W)

The **chargingSchedule** contains only one **chargingSchedulePeriod**

* Step 1/3:

Message **SetChargingProfileRequest**

chargingProfile.id <Same id for both chargingProfiles>

chargingProfile.chargingSchedule.startSchedule must NOT be omitted.

chargingProfile.stackLevel <Configured stackLevel> AND

chargingProfile.chargingProfilePurpose <Equal value for both profiles> AND (*TxDefaultProfile* OR *ChargingStationMaxProfile*)

chargingProfile.chargingProfileKind <Equal value for both profiles> AND

If **chargingProfile.chargingProfilePurpose** is *TxDefaultProfile* then **chargingProfile.chargingProfileKind** must be *Absolute* OR

Recurring

If **chargingProfile.chargingProfilePurpose** is *ChargingStationMaxProfile* then **chargingProfile.chargingProfileKind** must be *Absolute*

If **chargingProfile.chargingProfileKind** is *Recurring* then **chargingProfile.recurrencyKind** must NOT be omitted, else omitted

The received Charging Profiles must comply with the requirements defined at part 2 specification.

Post scenario validations:

- N/a

Test case name	Clear Charging Profile - With chargingProfileId
Test case Id	TC_K_05_CSMS
...	

Tool validations	
* Step 1: Message ClearChargingProfileRequest chargingProfileId <Generated chargingProfileId> AND chargingProfileCriteria omit	
Post scenario validations: - N/a	

Test case name	Clear Charging Profile - With stackLevel/purpose combination for one profile
Test case Id	TC_K_06_CSMS
...	

Tool validations	
* Step 1: Message ClearChargingProfileRequest chargingProfileCriteria.chargingProfilePurpose TxDefaultProfile AND chargingProfileCriteria.stackLevel <Configured stackLevel> AND chargingProfileCriteria.evseld <Configured evseld> AND chargingProfileId must be omitted.	
Post scenario validations: - N/a	

Test case name	Clear Charging Profile - Without previous charging profile
Test case Id	TC_K_08_CSMS
...	

Tool validations	
* Step 1: Message ClearChargingProfileRequest chargingProfileCriteria.chargingProfilePurpose TxDefaultProfile AND chargingProfileCriteria.stackLevel <Configured stackLevel> AND chargingProfileCriteria.evseld <Configured evseld> AND chargingProfileId must be omitted.	
Post scenario validations: - N/a	

Test case name	Set Charging Profile - Not Supported
Test case Id	TC_K_15_CSMS
...	

Tool validations

* Step 1:

Message **SetChargingProfileRequest**
- **evseld** <Configured evseld> AND
- **chargingProfile.stackLevel** <Configured stackLevel> AND
- **chargingProfile.chargingProfilePurpose** **TxDefaultProfile** AND
- **chargingProfile.chargingProfileKind** **Absolute** AND
- **chargingProfile.chargingSchedule.startSchedule** <Not omitted> AND
- **chargingProfile.chargingSchedule.chargingRateUnit** <Configured ChargingRateUnit> AND
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod** 0 AND
- **chargingProfile.chargingSchedule.duration** <Configured duration>
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.limit** 6.0 or 6000.0 AND
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases** <Configured numberPhases> where <Configured numberPhases> not 3 OR
- **chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases** <Configured numberPhases> or <omit> where <Configured numberPhases> 3 +

Post scenario validations:

- N/a

Test case name	Get Charging Profile - EvsId 0
Test case Id	TC_K_29_CSMS
...	

Tool validations

* Step 1:

Message **GetChargingProfilesRequest**

- **evsId** 0 AND
- **chargingProfile.chargingProfilePurpose** <Configured chargingProfilePurpose> AND
Note : chargingProfilePurpose is included, because the chargingProfile field is required and may not be left empty.
- **chargingProfile.stackLevel** must be omitted AND
- **chargingProfile.chargingLimitSource** must be omitted AND
- **chargingProfile.chargingProfileId** must be omitted

Post scenario validations:

- N/a

Test case name	Get Charging Profile - EvsId > 0
Test case Id	TC_K_30_CSMS
...	

Tool validations

* Step 1:

Message **GetChargingProfilesRequest**

- **evsId** <Configured evsId> AND
Note : chargingProfilePurpose is included, because the chargingProfile field is required and may not be left empty.
- **chargingProfile.chargingProfilePurpose** <Configured chargingProfilePurpose> AND
- **chargingProfile.stackLevel** must be omitted AND
- **chargingProfile.chargingLimitSource** must be omitted AND
- **chargingProfile.chargingProfileId** must be omitted

Post scenario validations:

- N/a

Test case name	Get Charging Profile - No EvsId
Test case Id	TC_K_31_CSMS
...	

Tool validations

* Step 1:

Message **GetChargingProfilesRequest**

- **evseld** must be omitted AND

- **chargingProfile.chargingProfilePurpose** <Configured chargingProfilePurpose> AND

Note : chargingProfilePurpose is included, because the chargingProfile field is required and may not be left empty.

- **chargingProfile.stackLevel** must be omitted AND

- **chargingProfile.chargingLimitSource** must be omitted AND

- **chargingProfile.chargingProfileId** must be omitted

Post scenario validations:

- N/a

Test case name	Get Charging Profile - chargingProfileId
Test case Id	TC_K_32_CSMS
...	

Tool validations

* Step 1:

Message **GetChargingProfilesRequest**

- **evseld** must be omitted AND

- **chargingProfile.chargingProfilePurpose** must be omitted AND

- **chargingProfile.stackLevel** must be omitted AND

- **chargingProfile.chargingLimitSource** must be omitted AND

- **chargingProfile.chargingProfileId** <received chargingProfileId>

Post scenario validations:

- N/a

Test case name	Get Charging Profile - Evseld > 0 + stackLevel
Test case Id	TC_K_33_CSMS
...	

Tool validations

* Step 1:

Message **GetChargingProfilesRequest**

- **evseld** <Configured evseld> AND

- **chargingProfile.stackLevel** <Configured stackLevel> AND

- **chargingProfile.chargingProfilePurpose** must be omitted AND

- **chargingProfile.chargingLimitSource** must be omitted AND

- **chargingProfile.chargingProfileId** must be omitted

Post scenario validations:

- N/a

Test case name	Get Charging Profile - Evseld > 0 + chargingLimitSource
Test case Id	TC_K_34_CSMS
...	

Tool validations

* Step 1:

Message **GetChargingProfilesRequest**

- **evseld** <Configured evseld> AND

- **chargingProfile.chargingLimitSource** <Configured chargingLimitSource> AND

- **chargingProfile.stackLevel** must be omitted AND

- **chargingProfile.chargingProfilePurpose** must be omitted AND

- **chargingProfile.chargingProfileId** must be omitted

Tool validations	
Post scenario validations:	
- N/a	
Test case name	
Test case name	Get Charging Profile - EvsId > 0 + chargingProfilePurpose
Test case Id	TC_K_35_CSMS
...	
Tool validations	
* Step 1:	
Message GetChargingProfilesRequest	
- evsId <Configured evsId> AND	
- chargingProfile.chargingProfilePurpose <Configured chargingProfilePurpose> AND	
- chargingProfile.stackLevel must be omitted AND	
- chargingProfile.chargingLimitSource must be omitted AND	
- chargingProfile.chargingProfileId must be omitted	
Post scenario validations:	
- N/a	
Test case name	
Test case name	Get Charging Profile - EvsId > 0 + chargingProfilePurpose + stackLevel
Test case Id	TC_K_36_CSMS
...	
Tool validations	
* Step 1:	
Message GetChargingProfilesRequest	
- evsId <Configured evsId> AND	
- chargingProfile.chargingProfilePurpose <Configured chargingProfilePurpose> AND	
- chargingProfile.chargingLimitSource must be omitted AND	
- chargingProfile.stackLevel <Configured stackLevel> AND	
- chargingProfile.chargingProfileId must be omitted	
Post scenario validations:	
- N/a	
Test case name	
Test case name	Set Charging Profile - TxProfile with ongoing transaction on the specified EVSE
Test case Id	TC_K_60_CSMS
...	
Tool validations	
* Step 1:	
(Message: SetChargingProfileRequest)	
chargingProfile.chargingProfilePurpose is <i>TxProfile</i> AND	
chargingProfile.evsId is <Configured evsId> AND	
chargingProfile.transactionId <Generated transactionId> AND	
chargingProfile.chargingProfileKind is <i>Relative OR Absolute</i>	
If chargingProfileKind is <i>Relative</i> then chargingSchedule.startSchedule must be omitted.	
If chargingProfileKind is <i>Absolute</i> then chargingSchedule.startSchedule must NOT be omitted.	
The received Charging Profile must comply with the requirements defined at part 2 specification.	
Post scenario validations:	
N/a	

Test case name	Remote start transaction with charging profile - Success
Test case Id	TC_K_37_CSMS
...	

Tool validations	
* Step 1:	
Message: RequestStartTransactionRequest	
idToken.idToken <Configured valid idToken>	
idToken.type <Configured valid idToken type>	
evseld <Configured evseld>	
chargingProfile contains:	
chargingProfile.chargingProfilePurpose is <i>TxProfile</i>	
chargingProfile.transactionId is omitted	
chargingProfile.chargingProfileKind is <i>Relative</i> OR <i>Absolute</i>	
If chargingProfileKind is <i>Relative</i> then chargingSchedule.startSchedule must be omitted.	
If chargingProfileKind is <i>Absolute</i> then chargingSchedule.startSchedule must NOT be omitted.	
The received Charging Profile must comply with the requirements defined at part 2 specification.	
Post scenario validations:	
N/a	

Test case name	Set Charging Profile - Multiple Profiles
Test case Id	TC_K_70_CSMS
Use case Id(s)	n/a
Requirement(s)	n/a
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 set multiple Charging Profiles.
Prerequisite(s)	n/a

Tool validations	
* Step 1:	
Message SetChargingProfileRequest	
chargingProfile.chargingProfilePurpose <i>TxDefaultProfile</i>	
chargingProfile.chargingProfileKind is <i>Absolute</i> OR <i>Recurring</i>	
chargingProfile.chargingSchedule.startSchedule must NOT be omitted.	
If chargingProfile.chargingProfileKind is <i>Recurring</i> then chargingProfile.recurrencyKind must NOT be omitted.	
* Step 3:	
Message SetChargingProfileRequest	
chargingProfile.id <different id from chargingProfile at step 1>	
chargingProfile.chargingProfilePurpose <i>ChargingStationMaxProfile</i>	
chargingProfile.chargingProfileKind is <i>Absolute</i>	
chargingProfile.chargingSchedule.startSchedule must NOT be omitted.	
The received Charging Profiles must comply with the requirements defined at part 2 specification.	
Post scenario validations:	
- N/a	

6.3.14. Page 728 - (2024-09) - TC_K_03_CSMS - Not requiring validFrom/To fields in charging profile [O20-4592] and chargingProfileKind must be Absolute [O20-4591]

Test case name	Set Charging Profile - ChargingStationMaxProfile	
Test case Id	TC_K_03_CSMS	
...		
Main (Test scenario)	Charging Station	CSMS

Tool validations	<p>* Step 1:</p> <p>Message SetChargingProfileRequest</p> <p>evsId 0 AND</p> <p>chargingProfile.stackLevel <Configured stackLevel> AND</p> <p>chargingProfile.chargingProfilePurpose ChargingStationMaxProfile_ AND</p> <p>chargingProfile.chargingProfileKind Absolute OR Relative</p> <p>chargingProfile.chargingSchedule.chargingRateUnit <Configured ChargingRateUnit></p> <p>chargingProfile.chargingSchedule.duration <Configured duration></p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 8.0 or 8000.0</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases></p> <p>where <Configured numberPhases> not 3 OR</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases></p> <p>or <omit> where <Configured numberPhases> 3</p> <p>chargingProfile.validFrom <Not omitted></p> <p>chargingProfile.validTo <Not omitted></p> <p>chargingProfile.chargingSchedule.startSchedule <Not omitted></p>	
	<p>Post scenario validations:</p> <ul style="list-style-type: none"> - N/a 	

6.3.15. Page 733 - (2024-09) - TC_K_10_CSMS - Not requiring validFrom/To fields in charging profile [O20-4592]

Test case name	Set Charging Profile - TxDefaultProfile - All EVSE	
Test case Id	TC_K_10_CSMS	
...		
Main (Test scenario)	Charging Station	CSMS

Test case name	Set Charging Profile - TxDefaultProfile - All EVSE
Tool validations	<p>* Step 1:</p> <p>Message SetChargingProfileRequest</p> <p>evsId 0 AND</p> <p>chargingProfile.stackLevel <Configured stackLevel> AND</p> <p>chargingProfile.chargingProfilePurpose TxDefaultProfile AND</p> <p>chargingProfile.chargingProfileKind Absolute AND</p> <p>chargingProfile.validFrom <Not omitted> AND</p> <p>chargingProfile.validTo <Not omitted> AND</p> <p>chargingProfile.chargingSchedule.startSchedule <Not omitted> AND</p> <p>chargingProfile.chargingSchedule.chargingRateUnit <Configured ChargingRateUnit> AND</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0 AND</p> <p>chargingProfile.chargingSchedule.duration <Configured duration></p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6.0 or 6000.0 AND</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases></p> <p>where <Configured numberPhases> not 3 OR</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases></p> <p>or <omit> where <Configured numberPhases> 3</p>
	<p>Post scenario validations:</p> <ul style="list-style-type: none"> - N/a

6.3.16. Page 734 - (2024-09) - TC_K_15_CSMS - Not requiring validFrom/To fields in charging profile [O20-4592]

Test case name	Set Charging Profile - Not Supported	
Test case Id	TC_K_15_CSMS	
...		
Main (Test scenario)	Charging Station 2. The OCTT responds with RPC Framework: CALLERROR: NotSupported.	CSMS 1. The CSMS sends a SetChargingProfileRequest with: <p>evsId <Configured evsId> AND</p> <p>chargingProfile.stackLevel <Configured stackLevel></p> <p>AND</p> <p>chargingProfile.chargingProfilePurpose TxDefaultProfile AND</p> <p>chargingProfile.chargingProfileKind Absolute AND</p> <p>chargingProfile.validFrom <Not omitted> AND</p> <p>chargingProfile.validTo <Not omitted> AND</p> <p>chargingProfile.chargingSchedule.startSchedule <Not omitted> AND</p> <p>chargingProfile.chargingSchedule.chargingRateUnit <Configured ChargingRateUnit> AND</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.startPeriod 0 AND</p> <p>chargingProfile.chargingSchedule.duration <Configured duration></p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.limit 6.0 or 6000.0 AND</p> <p>chargingProfile.chargingSchedule.chargingSchedulePeriod.numberPhases <Configured numberPhases></p>
Tool validations	...	

6.3.17. Page 753 - (2024-09) - TC_K_55_CSMS, TC_K_57_CSMS, TC_K_58_CSMS, TC_K_59_CSMS Removed triggerReason = ChargingRateChanged [776]

A trigger reason ChargingStateChange must only be sent, when an external actor (not CSMS) changes the charging rate. Therefore, removed the sending of a triggerReason=ChargingStateChanged by OCTT. This does not affect tool validations, but it was incorrect behavior of OCTT.

The step that sends a TransactionEventRequest with triggerReason=ChargingRateChanged has been removed from:

- TC_K_55_CSMS
- TC_K_57_CSMS
- TC_K_58_CSMS
- TC_K_59_CSMS

6.3.18. Page 760 - (2024-11) - TC_K_70_CSMS - Updated tool validation chargingProfiles and added preparation step.

We require a CSMS to install multiple ChargingProfiles with the same purpose and for the same connectordId with a different stackLevel. However there are CSMSs that do the stacking themselves and are unable to do this. Therefore it was decided to use different purposes instead.

Test case name	Set Charging Profile - Multiple Profiles	
Test case Id	TC_K_70_CSMS	
...		
Before (Preparations)	...	
Charging State: State is <i>EnergyTransferStarted</i>		
Main (Test scenario)	Charging Station	CSMS
	2. The OCTT responds with a SetChargingProfileResponse with status Accepted	1. The CSMS sends a SetChargingProfileRequest with chargingProfilePurpose TxDefaultProfile
	4. The OCTT responds with a SetChargingProfileResponse with status Accepted	3. The CSMS sends a SetChargingProfileRequest with chargingProfilePurpose ChargingStationMaxProfile
Tool validations	<p>* Step 1: Message SetChargingProfileRequest chargingProfile.chargingProfilePurpose TxDefaultProfile</p> <p>* Step 3: Message SetChargingProfileRequest chargingProfile.id <different id for both chargingProfiles> chargingProfile.chargingProfilePurpose ChargingStationMaxProfile</p> <p>Post scenario validations: - N/a</p>	

6.3.19. Page 824 - (2025-04) - TC_N_62_CSMS - Check only single identifier is provided

Added validation that idToken and customerCertificate are omitted.

TC_N_62_CSMS: Clear Customer Information - Clear and report - customerIdentifier

Test case name	Clear Customer Information - Clear and report - customerIdentifier
Test case Id	TC_N_62_CSMS

Test case name	Clear Customer Information - Clear and report - customerIdentifier			
...	...			
Before (Preparations)				
..				
Main (Test scenario)				
Charging Station	CSMS			
...	...			
Tool validations				
<ul style="list-style-type: none"> * Step 1: <p>Message CustomerInformationRequest</p> <ul style="list-style-type: none"> - report true - clear true - customerIdentifier "OpenChargeAlliance" - idToken is omitted - customerCertificate is omitted 				
Post scenario validations:				
<ul style="list-style-type: none"> - N/a 				
<h3>6.3.20. Page 830 - (2024-11) - TC_N_46_CSMS - Updated tool validation sendLocalListRequest</h3>				
Test case name	Clear Customer Information - Update Local Authorization List			
Test case Id	TC_N_46_CSMS			
...	...			
Before (Preparations)	<p>...</p> <p>Memory State: A local authorization list with <Configured valid_idtoken_idtoken> is configured.</p> <p>...</p>			
Main (Test scenario)	Charging Station	CSMS		
		
Tool validations	<ul style="list-style-type: none"> * Step 1: <p>Message CustomerInformationRequest</p> <ul style="list-style-type: none"> - report true AND - clear true AND - idToken.idToken <Configured valid_idtoken_idtoken> - idToken.type <Configured valid_idtoken_type> <ul style="list-style-type: none"> * Step 5: <p>Message SendLocalListRequest</p> <ul style="list-style-type: none"> - updateType Differential - versionNumber <Bigger than currently configured in OCTT> - localAuthorizationList <Contains only the configured valid_idtoken_idtoken, without idTokenInfo> <p>OR</p> <ul style="list-style-type: none"> - updateType Full - localAuthorizationList <Does NOT contain configured valid_idtoken_idtoken> <p>Post scenario validations:</p> <ul style="list-style-type: none"> - N/a 			

6.3.21. Page 830 - (2025-02) - TC_N_46_CSMS - Aligning configuration variable usage

Main (Test scenario)	
Charging Station	CSMS
Manual action : Trigger CSMS to CustomerInformationRequest to both report and clear token <Configured valid_idtoken_idtoken> and <Configured valid_idtoken_type>	
2. The Test System responds with a CustomerInformationResponse with status Accepted	1. The CSMS sends a CustomerInformationRequest
3. The Test System sends a NotifyCustomerInformationRequest	4. The CSMS responds with a NotifyCustomerInformationResponse .
Manual action : If not triggered automatically, trigger CSMS to send SendLocalListRequest with version = <configured local list version> + 1 and updateType = Differential and localAuthorizationList = [{idToken = {<Configured_valid_idtoken>, <Configured valid_idtoken_type>}}]	
6 The Test System responds with a SendLocalListResponse with status Accepted	5. The CSMS sends a SendLocalListRequest
Note(s): If the Local Authorization List is too big for one message, step 5 and 6 will be repeated	
Tool validations	
<ul style="list-style-type: none"> * Step 1: Message CustomerInformationRequest <ul style="list-style-type: none"> - report true AND - clear true AND - idToken.idToken <Configured valid_idtoken_idtoken> - idToken.type <Configured valid_idtoken_type> * Step 5: Message SendLocalListRequest <ul style="list-style-type: none"> - updateType Differential - versionNumber <configured local list version> + 1 - localAuthorizationList <Contains only the configured valid_idtoken_idtoken, without idTokenInfo> - localAuthorizationList[0].idToken contains <configured_valid_idtoken_idtoken> and <configured valid_idtoken_type> - localAuthorizationList[0].idTokenInfo <omitted> 	
Post scenario validations:	
<ul style="list-style-type: none"> - All messages have been received 	

6.3.22. Page 854 - (2025-02) - TC_O_27_CSMS - Fixing validations to be more specific for test case

Tool validations

* Step 1:

Message **SetDisplayMessageRequest**

- **message.id** is <Generated Id>

- **message.startDateTime** is <Configured startDateTime>

- **message.transactionId** is present

- **message.state** is <omitted>

- **message.startDateTime** is <Configured startDateTime>

- **message.endDateTime** is <omitted>

- **message.transactionId** is <Generated transactionId from Before>

Post scenario validations:

- N/a

6.3.23. Page 855 - (2025-02) - TC_O_28_CSMS - Fixing validations to be more specific for test case

A test case needs to start in Preparations phase:

Before (Preparations)

Configuration State:

N/a

Memory State:

N/a

Reusable State:

State is *EnergyTransferStarted*

Tool validations

* Step 1:

Message **SetDisplayMessageRequest**

- **message.id** is <Generated Id>

- **message.priority** is <Configured Priority>

- **message.endDateTime** is <Configured endDateTime>

- **message.state** is <Configured State>

- **message.transactionId** is present

- **message.state** is <omitted>

- **message.startDateTime** is <omitted>

- **message.endDateTime** is <Configured endDateTime>

- **message.transactionId** is <Generated transactionId from Before>

Post scenario validations:

- N/a