# OCPP 2.0.1

## Part 2 - Errata v2.0

v2.0, 2022-12-15

# Table of Contents

1. Disclaimer . . . 1
2. Scope . . . 2
    2.1. Terminology and Conventions . . . 2
3. General . . . 3
    3.1. (v1) Requirements take precedence over text [14] . . . 3
    3.2. (v1) Error in changelog [381] . . . 3
    3.3. (v1) Respond to request before sending result messages [397] . . . 3
    3.4. (v2) ChargingState = EVDetected should be read as EVConnected [545] . . . 3
    3.5. (v2) Recommendation on restoring a connection [567] . . . 4
4. Use case A Security . . . 4
    4.1. Page 19 - (v1) Note about having correct date to validate certificates [496] . . . 4
    4.2. Page 20 - (v1) Type of BasicAuthPassword [489] . . . 4
    4.3. Page 20 - (v1) Note about changed password encoding between OCPP 1.6 and 2.0.1 [431] . . . 5
    4.4. Page 20 - (v1) Unnecessary precondition for A00.FR.205 [429] . . . 5
    4.5. Page 20 - (v1) Implicit requirement for CSMS made explicit [440] . . . 5
    4.6. Page 23 - (v1) Wrong precondition in requirements [444] . . . 6
    4.7. Page 24 - (v1) Provision for accepting Charging Station with expired certificate [401] . . . 7
    4.8. Page 29 - (v1) Recommendations for handling expired manufacturer certificate [400] . . . 7
    4.9. Page 31 - (v1) Additional requirements for updating a Charging Station certificate . . . 8
    4.10. Page 33 - (v1) Clarification for reconnection with new certificate [149] . . . 9
5. Use case B Provisioning . . . 10
    5.1. Page 45 - (v1) Hearbeat interval must be greater than zero [370] . . . 10
    5.2. Page 54 - (v1) Return first instance when no instance specified [476] . . . 10
    5.3. Page 54 - (v2) Return only specified instance with GetVariablesRequest [608] . . . 10
    5.4. Page 55 - (v2) Use case B06: Requirement missing when exceeding ItemsPerMessageGetVariables [605] . . . 11
    5.5. Page 56 - (v1) Use case B07: Requirement for CSMS to request FullInventory is missing [511] . . . 11
    5.6. Page 56 - (v1) Use case B08 Get Custom Report: conflicting requirements [355] . . . 12
    5.7. Page 59 - (v2) Use case B08: Requirement missing when exceeding ItemsPerMessageGetReport [615] . . . 12
    5.8. Page 59 - (v2) Use case B08: GetReport handling of empty variables or instances [565] . . . 12
    5.9. Page 62 - (v2) Use case B10: Requirement note should have been a separate requirement and missing preconditions . . . 13
    5.10. Page 63 - (v2) Use case B11: allow Unavailable notification during reset [581] . . . 13
    5.11. Page 66 - (v1) Incomplete preconditions in B12 [442] . . . 14
6. Use case C Authorization . . . 14
    6.1. Page 74 - (v1) Part of requirement is actually a precondition [439] . . . 14
    6.2. Page 74 - (v2) C01.FR.06 AuthorizeResponse must return groupIdToken [641] . . . 15
    6.3. Page 75 - (v1) Requirement to avoid authorizing multiple idTokens in a transaction [514] . . . 15
    6.4. Page 83 - (v1) Sequence diagram use case C05 [435] . . . 16
    6.5. Page 84 - (v1) Description and remarks of use case C05 [503, 504] . . . 16
    6.6. Page 89 - (v2) Use case C07: *certificate* in AuthorizeRequest contains certificate chain [533] . . . 17
    6.7. Page 94 - (v2) Improved requirements for GroupIdToken [601] . . . 18
    6.8. Page 94 - (v1) Requirement C09.FR.11 is partly incorrect [513] . . . 19
    6.9. Page 101 - (v2) C13.FR.02/04 are about authorizing, instead of starting [640] . . . 19
    6.10. Page 105 - (v1) ChargingState in requirement C15.FR.03 [464] . . . 20
    6.11. Page 105 - (v2) Incorrect TxStartPoint PowerPathClosed reference in requirement precondition . . . 20
7. Use case E Transactions . . . 21
    7.1. Page 114 - (v2) Improved definition of transaction [538] . . . 21
    7.2. Page 115 - (v1) Setting for OCPP 1.6 transaction compatibility [516] . . . 22
    7.3. Page 116 - (v1) Using *seqNo* in TransactionEventRequest when EVSE is not known [525] . . . 22
    7.4. Page 116 - (v2) Solution for ever-increasing seqNo per EVSE may lead to race condition and eventual overflow [592] . . . 23
    7.5. Page 116 - (v2) Clarification for optional fields in TransactionEventRequest [101] . . . 23
    7.6. Page 119 - (v2) Use case E: references to AlignedDataSignReadings instead of SampledDataSignReadings [626] . . . 24
    7.7. Page 122 - (v2) Obsolete references to power relay for PowerPathClosed [600] . . . 26
    7.8. Page 123 - (v2) E01.FR.12 TransactionEventResponse must also return groupIdToken [642] . . . 28
    7.9. Page 123 - (v2) Requirements about chargingState and SuspendedEV [582] . . . 28
    7.10. Page 125 - (v2) Use case E02: Invalid remark about multiple authorizations [568] . . . 28
    7.11. Page 125 - (v2) Sequence diagram error: idToken.id shown more than once [591] . . . 29

  7.12. Page 126 - (v1) Requirement E02.FR.05 new note about NotifyEvent [437] 29
  7.13. Page 126 - (v1) Precondition E02.FR.06 is incomplete [438] 30
  7.14. Page 126 - (v1) Missing requirement about authorization during transaction [514][547] 30
  7.15. Page 126 - (v2) Broken table caused a requirement to be not shown 30
  7.16. Page 130 - (v1) Precondition E03.FR.04 is incomplete [435] 31
  7.17. Page 139 - (v2) Handling of EVConnectionTimeout better defined [627] 31
  7.18. Page 136 - (v2) E05.FR.02 should not contain PowerPathClosed [604] 33
  7.19. Page 137 - (v2) ChargingState in requirement E05.FR.09 [596][647] 33
  7.20. Page 140 - (v2) Use case E06: scenario S4 is not valid [537] 34
  7.21. Page 142 - (v2) Use case E06 is missing a common stopping scenario [540] 34
  7.22. Page 143 - (v1) Confusing precondition in E06.FR.06 [506] 35
  7.23. Page 143 - (v1) Requirements for eventType=Started do not belong in use case E06/E07 [453] 35
  7.24. Page 143 - (v2) Use case E07 should use TxStopPoint = Authorized [541] 36
  7.25. Page 144 - (v1) Use case E07, sequence diagram is not complete 38
  7.26. Page 145 - (v2) Report *idToken* for stopping authorization [583] 40
  7.27. Page 162 - (v1) Misspelled field name in requirements and remark(s) [452] 41
8. Use case F Remote Control 42
  8.1. Page 169 - (v2) Use case F01: requirements unclear about when transaction starts [580] 42
  8.2. Page 169 - (v1) Requirements for rejecting request are missing [449, 509] 44
  8.3. Page 170 - (v1) Requirement missing for *remoteStartId* [517] 44
  8.4. Page 170 - (v2) Use case F01/F02: ChargingProfile in RequestStartTransaction [558] 44
  8.5. Page 171 - (v1) Sequence diagram error [435] 45
  8.6. Page 171 - (v2) Use case F02: sequence diagram for TxStartPoint=EVConnected [579] 47
  8.7. Page 172 - (v1) Some requirements of use case E03 apply also to F02 [441] 48
  8.8. Page 172 - (v2) Relevant F01 requirements copied to F02 Remote Start First [590] 49
  8.9. Page 174 - (v2) Use case F03: replace TxStartPoint by TxStopPoint [557] 50
  8.10. Page 174 - (v2) Use case F03: Some requirements only valid for TxStopPoint = EVConnected [563] 51
  8.11. Page 180 - (v1) Requirement F06.FR.07 description improvement [451] 51
  8.12. Page 180 - (v1) Note to requirement F06.FR.11 is contradicting F06.FR.12 52
  8.13. Page 180 - (v1) Requirement F06.FR.12: evseId cannot be 0 [450] 52
9. Use case G Availability 52
  9.1. Page 184 - (v2) Use case G01: added alternative scenario using NotifyEvent [530] 53
  9.2. Page 184 - (v2) Missing requirements for StatusNotification [569] 53
  9.3. Page 184 - (v2) StatusNotifications for multiple connectors on an EVSE [530] 54
  9.4. Page 188 - (v1) Precondition of G03.FR.05 is incomplete [368] 54
10. Use case H Reservation 54
  10.1. Page 197 - (v1) Merging two requirements [445] 54
  10.2. Page 197 - (v1) Missing requirement about reserving for *evseId* [472, 505] 55
  10.3. Page 197 - (v1) Requirement H01.FR.20 needs to be split [447, 472] 55
  10.4. Page 197 - (v2) Using field *reservationId* instead of *id* in ReserveNow [584] 56
  10.5. Page 203 - (v2) incorrect dataType reference AvailabilityType should be operationalStatus 56
11. Use case I Tariff And Cost 56
  11.1. Page 207 - (v1) Use case I02: Show EV Driver Running Total Cost During Charging in TransactionEventResponse [361] 57
12. Use case J Meter Values 58
  12.1. Page 214 - (v1) Limit the amount of meter values in TransactionEvent(Ended) [371] 58
  12.2. Page 215 - (v2) section 2.2: Incorrect sentence in description for AlignedDataTxEndedInterval [624] 58
  12.3. Page 216 - (v1) section 2.3: Including phases for register meter values [328] 59
  12.4. Page 218 - (v1) Missing requirement for dealing with AlignedDataDuringIdle [457] 60
  12.5. Page 218 - (v1) Wrong placement of Aligned/SampledDataSignReadings 60
  12.6. Page 220 - (v2) Use case J02: improved definition of J01.FR.11 [535] 60
  12.7. Page 221 - (v1) Use case J03 description of dealing with ISO 15118 signed metering receipts is unclear [352, 353] 61
13. Use case K Smart Charging 62
  13.1. Page 231 - (v2) Chapter K.2.6.1: error in JSON examples [552] 62
  13.2. Page 233 - (v1) Use case K01: Recommendation to not limit duration of TxProfile [519] 63
  13.3. Page 233 - (v2) Transaction does not fall back to TxDefaultProfile when TxProfile ends early [603] 63
  13.4. Page 234 - (v1) Use case K01: SetChargingProfile cannot replace external constraints. [507] 64
  13.5. Page 234 - (v2) Use case K01: Improved explanation of TxDefaultProfile on evseId #0 [593] 64
  13.6. Page 236 - (v2) Use case K01: *numberPhases*, *phaseToUse* does not match EVSE capabilities [577] 66
  13.7. Page 236 - (v1) Use case K01: Missing requirement for *startSchedule* [510] 68

13.8. Page 235 - (v2) Definition of start of a relative charging profile [638] . . . 69
13.9. Page 238 - (v2) Use case K02: Incorrect note of K02.FR.03 [597] . . . 70
13.10. Page 245 - (v1) Use case K05: Remote Start Transaction with Charging Profile [354] . . . 70
13.11. Page 245 - (v2) Use case K05: added requirements from F01/F02 [558] . . . 70
13.12. Page 250 - (v1) Unclear precondition of K08.FR.06 [462] . . . 71
13.13. Page 251 - (v1) Requirements K09 GetChargingProfiles [406] . . . 71
13.14. Page 252 - (v1) Use case K10 requirements are incomplete [502] . . . 73
13.15. Page 253 - (v1) K01.FR.34 refers to ChargingSchedulePeriodType, but should be ChargingScheduleType [363] . . . 73
13.16. Page 254 - (v1) Charging profile id's for external constraints profiles [365] . . . 74
13.17. Page 259 - (v1) section K15: Recommendation for NotifyEVChargingScheduleRequest [349] . . . 74
13.18. Page 259 - Improved requirement definitions regarding NotifyEVChargingScheduleRequest [349] . . . 76
13.19. Page 260 - (v1) Use case K15: Improved error handling description . . . 76
13.20. Page 260 - (v1) Note to requirement K15.FR.07 about composite schedule . . . 77
13.21. Page 261 - (v1) Requirement K15.FR.17 has incomplete precondition [366] . . . 77
14. Use case L Firmware Management . . . 78
14.1. Page 271, 277 - (v2) Sequence diagrams allow reboot before status=Installed [639] . . . 78
14.2. Page 273 - (v2) Firmware update process graph is incomplete [634] . . . 80
14.3. Page 273 - (v1) Missing requirement if firmware verification fails [455] . . . 81
14.4. Page 274, 278 - (v1) Changed note of L01.FR.13 [456] . . . 81
14.5. Page 274 - (v2) Changed preconditions for DownloadScheduled [614] . . . 82
14.6. Page 275, 279 - (v1) Requirement for DownloadFailed missing [384] . . . 82
14.7. Page 275, 279 - (v2) Requirement for Installed is unclear [634] . . . 83
14.8. Page 275 - (v2) Requirement for SecurityEvent FirmwareUpdated [648] . . . 83
14.9. Page 275 - (v2) Allow the Charging Station to report InstallRebooting after installing, but before activating the new firmware . . . 84
15. Use case M ISO15118 Certificate Management . . . 84
15.1. Page 285 - (v2) Chapter M: improved sequence diagram Figure 121 . . . 84
15.2. Page 288 - (v1) Improving definition of V2GRootCertificate [283] . . . 86
15.3. Page 292 - (v1) Use Cases M01 and M02, contract certificate pool [288] . . . 86
15.4. Page 294 - (v1) Some occurences of *typeOfCertificate* instead of *certificateType* [389] . . . 87
15.5. Page 296 - (v2) Use case M05: reference to sub-CA certificate instead of root certificate [388] . . . 87
15.6. Page 296 - (v1) Use case M04, A Charging Station should be allowed to prevent the deletion of the last certificate from a defined certificate type . . . 87
15.7. Page 296 - (v1) Requirement M04.FR.06 misses status code [471] . . . 88
15.8. Page 296 - (v2) Use case M04: Deletion of certificate also deletes its child certificates [309] . . . 88
15.9. Page 298 - (v2) Use case M05: missing requirement about replacing a certificate [617] . . . 88
15.10. Page 299 - (v2) M06.FR.04 refers to wrong status [576] . . . 89
16. Use case N Diagnostics . . . 89
16.1. Page 303 - (v1) Misspelled enumeration in N01.FR.10 [443] . . . 89
16.2. Page 303 - (v1) Requirements for GetLogRequest are incomplete [497] . . . 89
16.3. Page 303 - (v2) LogStatusNotification for AcceptedCanceled made explicit [594] . . . 90
16.4. Page 305 - (v1) Use case N02 Get Monitoring Report: conflicting requirements [355] . . . 90
16.5. Page 305 - (v2) Use case N02 GetMonitoringReport handling of empty variables or instances [565] . . . 90
16.6. Page 306 - (v2) Use case N03: SetMonitoringBaseRequest clarification [616] . . . 91
16.7. Page 307 - (v1) Misspelled type in requirement N04.FR.05 [372] . . . 92
16.8. Page 308 - (v1) Requirement N04.FR.10 is too restrictive [358] . . . 92
16.9. Page 308 - (v1) Precondition incomplete in N04.FR.11 [373] . . . 92
16.10. Page 308 - (v1) Precondition unclear in N04.FR.16 [374] . . . 93
16.11. Page 308 - (v2) Use case N04: Requirement missing that hardwired monitors cannot be changed [628] . . . 93
16.12. Page 308 - (v2) Use case N04: Recommend GetMonitoringReport after reboot [643] . . . 93
16.13. Page 310 - (v2) Use case N06: Requirement missing when exceeding ItemsPerMessage/BytePerMessage [620] . . . 94
16.14. Page 310 - (v1) Error in requirement N06.FR.05 [369] . . . 94
16.15. Page 311 - (v2) Use case N07: remark about OfflineMonitoringEventQueuingSeverity [633] . . . 94
16.16. Page 311 - (v2) Typo in N07.FR.04 [633] . . . 95
16.17. Page 311 - (v1) N07.FR.06 uses variableMonitoringId [477] . . . 95
16.18. Page 312 - (v1) Wrong precondition in requirement N07.FR.14 [356] . . . 95
16.19. Page 312 - (v1) Added note to requirements N07.FR.16 and NR.FR.17 [359] . . . 96
16.20. Page 312 - (v1) Moved requirements about periodic monitors from N07 to N08 [367] . . . 96
17. Use case P Data Transfer . . . 98

  - 17.1. Page 331 - (v2) Added example for using JSON in DataTransfer data [574] 98
- 18. Messages 98
  - 18.1. Page 339 - (v2) Error in description of CertificateSignedRequest [425] 98
  - 18.2. Page 344 - (v1) Typographical error in FirmwareNotificationRequest [480] 98
  - 18.3. Page 345 - (v2) Enlarging *exiResponse* in Get15118EVCertificate [637] 99
  - 18.4. Page 349 - (v2) Better definition of *retries* [586] 100
  - 18.5. Page 351 - (v1) Remark at InstallCertificateRequest [283] 100
  - 18.6. Page 358 - (v2) Description of groupIdToken in RequestStartTransactionRequest [645] 100
  - 18.7. Page 365 - (v1) Wrong description of timestamp in StatusNotificationRequest [379] 101
  - 18.8. Page 366 - (v2) Description of IdToken in TransactionEventRequest [583] 101
- 19. Datatypes 101
  - 19.1. Page 371 - (v2) Description of CertificateHashDataType [542] 101
  - 19.2. Page 380 - (v1) Description of MessageInfoType [481] 102
  - 19.3. Page 382 - (v2) Description of OCSPRequestDataType [542] 102
  - 19.4. Page 386 - (v1) SetVariablesRequest can have empty string [488] 102
- 20. Enumerations 102
  - 20.1. Page 400 - (v1) Wrong description of EventTriggerEnumType Alerting [351] 103
  - 20.2. Page 404 - (v1) LocationEnumType Inlet description [494] 103
  - 20.3. Page 406 - (v2) Voltage: improved description [544] 103
  - 20.4. Page 409 - (v1) Enumeration value for OCPP 2.0.1 missing [501] 103
- 21. Referenced Components and Variables 103
  - 21.1. Page 420 - (v1) List of Components and Variables in XLS format [145] 103
  - 21.2. Page 420 - (v1) Section 2 Referenced Components and Variables: Added reference to part 1 103
  - 21.3. Page 422 - (v1) Improved description of OfflineThreshold [487] 105
  - 21.4. Page 423 - (v2) Variable UnlockOnEVSideDisconnect does not have an evse attribute 105
  - 21.5. Page 432 - (v2) DeviceDataCtrlr.Items/BytesPerMessage also applies to GetMonitoringReport [649] 105
  - 21.6. Page 426 - (v1) Variable ClockCtrlr.TimeAdjustmentReportingThreshold [492] 106
  - 21.7. Page 427 - (v1) Variable SecurityCtrlr.BasicAuthPassword [489] 106
  - 21.8. Page 428 - (v2) Improved description of OrganizationName [393] 107
  - 21.9. Page 428 - (v1) Note with AdditionalRootCertificateCheck variable [479] 107
  - 21.10. Page 429 - (v1) Improved description of AuthEnabled [485] 107
  - 21.11. Page 430 - (v1) New optional variable DisableRemoteAuthorization [486] 107
  - 21.12. Page 431 - (v1) New optional variable DisablePostAuthorize [484] 108
  - 21.13. Page 434 - (v1) Better description of TxStartPoint/TxStopPoint [348] 108
  - 21.14. Page 435 - (v1) Note to SampledDataSignReadings [526] 110
  - 21.15. Page 438 - (v2) Variable AlignedDataSendDuringIdle does not have evse attribute 110
  - 21.16. Page 438 - (v2) AlignedDataSignReadings only applies to meter values in TransactionEvent [625] 111
  - 21.17. Page 439 - (v1) Note to PublicKeyWithSignedMeterValue [460] 111
  - 21.18. Page 440 - (v2) Improved description of ChargingProfileMaxStackLevel [551] 111
  - 21.19. Page 442 - (v2) Typo in description of QueueAllMessages [546] 112
  - 21.20. Page 445 - (v1) New optional MonitoringCtrlr variables [357] 112
  - 21.21. Page 446 - (v1) CustomImplementationEnabled is located in wrong section [387] 112
  - 21.22. Page 447 - (v1) Reporting new connector types [478] 112
  - 21.23. Page 449 - (v1) New variables for ISO15118Ctrlr about protocol version in use [378] 113
  - 21.24. Page 449 - (v2) Variables about ISO15118Ctrlr protocol versions moved to ConnectedEV [598] 114
  - 21.25. Page 449 - (v1) New configuration variables for ISO15118Ctrlr [180, 443] 114
  - 21.26. Page 449 - (v2) Variable ISO15118Ctrlr.PnCEnabled description improved and additional variables added for certificate installation 115
  - 21.27. Page 449 - (v2) Setting Plug-and-Charge variables [564, 571] 116
- 22. Appendix 1 Security Events 117
  - 22.1. Page 3 - (v2) Certain security events must be marked as Critical [585] 117
  - 22.2. Page 3 - (v2) Unclear which security events are mandatory to be implemented and how this relates to Critical events 118
- 23. Appendix 3 Standardized Components 118
  - 23.1. Page 8 - (v1) DefaultMessageTimeout missing for OCPPCommCtrlr [482] 118
  - 23.2. Page 8 - (v1) Description of NetworkConfigurationPriority for OCPPCommCtrlr [483] 118
  - 23.3. Page 9 - (v1) BasicAuthPassword of SecurityCtrlr [490] 119
  - 23.4. Page 14 - (v1) New component ConnectedEV 119
  - 23.5. Page 14 - (v2) ConnectedEV: new EnergyCapacity variable [553] 120
- 24. Appendix 5 Reason Codes 120

  - 24.1. (v1) MissingDeviceModelInfo [383] . . . . . . 120
- 25. List of Referenced Components and Typical Variables . . . . . . 121

# 1. Disclaimer

Copyright © 2010 – 2022 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 |
|---|---|---|
| 2.0 | 2022-12-15 | Release of 2.0.1 errata version 2. |
| 1.0 | 2021-10-01 | Release of 2.0.1 errata. |

# 2. Scope

This document contains errata on "part 2: Specification" and "part 2: Appendices" of the OCPP 2.0.1 documentation. These errata have to be read as an addition to the release of OCPP 2.0.1.

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.

## 2.1. 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 version 2 of the errata has added a series of new errata descriptions to the original version 1. The errata of the original version are marked with "(v1)" in the section title. The newly added errata are marked with "(v2)".

Where possible 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]".

# 3. General

## 3.1. (v1) Requirements take precedence over text [14]

Whenever there is any (apparent) conflict between narrative text and requirements in the specification document, the requirements have precedence.

## 3.2. (v1) Error in changelog [381]

The OCPP 2.0 - 2.0.1 Changelog document states that A00.FR.605 and A00.FR.606 were added, but they were removed

## 3.3. (v1) Respond to request before sending result messages [397]

There are a few messages that do not provide their result in the response message, but send one or more messages that contain the result. The request MUST be acknowledged first in a response message, before the messages with the information that was requested are sent. The CSMS needs to know that a request was accepted, so that it can expect result messages.

The Charging Station needs to acknowledge the messages in the list below with a response message first, before sending the follow-up message shown after the arrow ( → ):

- GetReport → NotifyReport
- GetBaseReport → NotifyReport
- GetMonitoringReport → NotifyMonitoringReport
- GetDisplayMessages → NotifyDisplayMessage
- CustomerInformation → NotifyCustomerInformation
- GetChargingProfiles → ReportChargingProfiles
- GetLog → LogStatusNotification
- UpdateFirmware → FirmwareStatusNotification
- PublishFirmware → PublishFirmwareStatusNotification
- TriggerMessage(<message>) → <requested message>

## 3.4. (v2) ChargingState = EVDetected should be read as EVConnected [545]

In several sequence diagrams a TransactionEventRequest message with "chargingState = EVDetected" is shown, but this should be: "chargingState = EVConnected". (`EVDetected` is a TriggerReasonEnumType and not a ChargingStateEnumType).

This occurs in the following figures:

- Figure 27
- Figure 37
- Figure 65
- Figure 67
- Figure 101
- Figure 102
- Figure 121

## 3.5. (v2) Recommendation on restoring a connection [567]

A charging station may discover that the connection to CSMS is not functioning correctly when it gets a timeout to a request or when the websocket ping is not answered. In such a situation it is advised that the charging station drops the connection and then reconnects to CSMS. This will create a fresh session and will possibly connect to a different endpoint of a multi-instance CSMS, which may resolve the error.

# 4. Use case A Security

## 4.1. Page 19 - (v1) Note about having correct date to validate certificates [496]

In section 1.3 just before the start of section 1.3.1 add the following bullet:

- When the Charging Station does not have the correct date and time set, it cannot validate the server certificate. A solution for this might be to either use NTP, mobile network to set time automatically, or have an installer tool that sets the time before the first connection.

## 4.2. Page 20 - (v1) Type of BasicAuthPassword [489]

The variable `BasicAuthPassword` is defined as being of type "identifierString". This type, however, is case-insensitive, which is not desirable for passwords. We therefore define a new type "passwordString", which is a string with the same definition of a "identifierString", but case-sensitive.

### 4.2.1. Page 5 - New primitive datatype passwordString

The following primitive datatype is added to 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"<br>or any of the following limited set of symbols: * - _ = : + \| @ . |

### 4.2.2. Page 20 - (v1) BasicAuthPassword security profile 1

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | A00.FR.205 | A00.FR.203 | The password SHALL be stored in the `BasicAuthPassword` Configuration Variable. It SHALL be a randomly chosen identifierString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by identifierString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). |
| New text | A00.FR.205 | A00.FR.203 | 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). |

### 4.2.3. Page 21 - (v1) BasicAuthPassword security profile 2

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | A00.FR.304 | A00.FR.302 | The password SHALL be stored in the `BasicAuthPassword` Configuration Variable. It SHALL be a randomly chosen identifierString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by identifierString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). |
| 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 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). |

## 4.3. Page 20 - (v1) Note about changed password encoding between OCPP 1.6 and 2.0.1 [431]

### 4.3.1. Page 20 - (v1) Security Profile 1

In table 13 "Security Profile 1" at row #7 enter the following remark:

|  |  |  |
|---|---|---|
| 7 | **Remark(s)** | Please note, that the encoding of the basic authentication password in OCPP 2.0.1 (A00.FR.205) differs from how this was done in OCPP 1.6. |

### 4.3.2. Page 21 - (v1) Security Profile 2

In table 15 "Security Profile 2" at row #7 add the following remark to the existing text:

|  |  |  |
|---|---|---|
| 7 | **Remark(s)** | [...]<br>Please note, that the encoding of the basic authentication password in OCPP 2.0.1 (A00.FR.304) differs from how this was done in OCPP 1.6. |

## 4.4. Page 20 - (v1) Unnecessary precondition for A00.FR.205 [429]

The precondition "A00.FR.203" for requirement A00.FR.205 is not correct. It has been removed as shown in the following table.

*Changed requirement*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | A00.FR.205 | A00.FR.203 | The password SHALL be stored in the `BasicAuthPassword` Configuration Variable. It SHALL be a randomly chosen identifierString with a sufficiently high entropy, consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by identifierString). 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 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). |

## 4.5. Page 20 - (v1) Implicit requirement for CSMS made explicit [440]

### 4.5.1. Page 20 - (v1) Security Profile 1

The security requirement A00.FR.202 specifies that a Charging Station shall authenticate itself with CSMS, but there is no matching requirement for CSMS for this. It is implicitly there, of course. The following addition makes this requirement explicit.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| A00.FR.207 | A00.FR.202 | The CSMS SHALL validate that Charging Station identity and the Basic Authentication password match with username and password in the authorization header of the connection request. |

### 4.5.2. Page 22 - (v1) Security Profile 2

The security requirement A00.FR.302 specifies that a Charging Station shall authenticate itself with CSMS, but there is no matching requirement for CSMS for this. It is implicitly there, of course. The following addition makes this requirement explicit.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| A00.FR.324 | A00.FR.302 | The CSMS SHALL validate that Charging Station identity and the Basic Authentication password match with username and password in the authorization header of the connection request. |

## 4.6. Page 23 - (v1) Wrong precondition in requirements [444]

The requirements A00.FR.322 and A00.FR.323 refer to A00.FR.321 as precondition, but that should be A00.FR.320.

*Changed requirements*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | A00.FR.322 | A00.FR.321<br>AND<br>The CSMS detects that the Charging Station only allows connections using one of these suites | The CSMS SHALL terminate the connection. |
| New text | A00.FR.322 | **A00.FR.320**<br>AND<br>The CSMS detects that the Charging Station only allows connections using one of these suites | The CSMS SHALL terminate the connection. |
| Old text | A00.FR.323 | A00.FR.321<br>AND<br>The Charging Station detects that the CSMS only allows connections using one of these suites | The Charging Station SHALL trigger an InvalidTLSCipherSuite security event AND terminate the connection (See part 2 appendices for the full list of security events). |
| Old text | A00.FR.323 | **A00.FR.320**<br>AND<br>The Charging Station detects that the CSMS only allows connections using one of these suites | The Charging Station SHALL trigger an InvalidTLSCipherSuite security event AND terminate the connection (See part 2 appendices for the full list of security events). |

## 4.7. Page 24 - (v1) Provision for accepting Charging Station with expired certificate [401]

A situation can occur where a Charging Station that has been operational for some time, becomes disconnected for a longer period of time. There may be roadworks, for example, that have caused the Charging Station to be taken offline. If this lasts for a long time, then the Charging Station certificate may expire before it is online again.

In that specific situation a CSO can instruct the CSMS to accept the Charging Station in a `Pending` state even though it has an expired certificate. The CSMS will then immediately execute use case A02 to update the certificate.

**NOTE** This is not a required feature for a CSMS.

This leads to the following new requirement.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| A00.FR.429 | If Charging Station certificate has been expired AND<br>CSMS has been explicitly configured to accept a connection by this **specific** Charging Station with an expired certificate. | CSMS MAY accept this Charging Station in a BootNotification - Pending state (use case B02) after which it SHALL immediately execute A02 - Update Charging Station Certificate by request of CSMS to renew the certificate. |

A minor change to the precondition of A00.FR.407 is needed to allow this.

*Changed requirements*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | A00.FR.407 | If the Charging Station does not own a valid certificate, or if the certification path is invalid | The CSMS SHALL terminate the connection. |
| New text | A00.FR.407 | **NOT A00.FR.429 AND**<br>If the Charging Station does not own a valid certificate, or if the certification path is invalid | The CSMS SHALL terminate the connection. |
| Old text | A00.FR.408 | A00.FR.407 | It is RECOMMENDED to log a security event in the CSMS. |
| New text | A00.FR.408 | A00.FR.407 **OR A00.FR.429** | It is RECOMMENDED to log a security event **`InvalidChargingStationCertificate`** in the CSMS. |

## 4.8. Page 29 - (v1) Recommendations for handling expired manufacturer certificate [400]

Manufacturers can install a manufacturer certificate during production of the Charging Station for connecting with security profile 3. Even with certificates that are valid for several years, a situation can occur where a Charging Station is stored for so long (e.g. as warehoused inventory), that the certificate is no longer valid when it is installed.

A requirement is added to allow connection by a Charging Station with an outdated manufacturer certificate, since that needs to be replaced anyway. If a CSMS does not support this, then the only option is to have the certificate be replaced by an engineer on-site.

**NOTE** This requirement is slightly more relaxed than A00.FR.429, because it is only valid for first time installation with a manufacturer certificate.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| A00.FR.807 | A00.FR.804 AND<br>Charging Station manufacturer certificate has expired | The CSMS MAY accept a connection by Charging Station in a `Pending` state after the BootNotification and immediately execute use case A02 - Update Charging Station Certificate by request of CSMS to install a new valid CSO certificate. |

# 4.9. Page 31 - (v1) Additional requirements for updating a Charging Station certificate

There was no back-off mechanism described for when the Charging Station never receives the signed certificate from CSMS, that was generated from the provided CSR.

The following requirements are added to address that.

*New requirements*

| ID | Precondition | Requirements |
|---|---|---|
| A02.FR.17 | When the CSMS accepted the SignCertificateRequest for a CSR AND the Charging Station did not yet receive a CertificateSignedRequest for this CSR AND the number of seconds configured at CertSigningWaitMinimum has expired | The Charging Station SHALL send a new SignCertificateRequest for the CSR. Optionally, this CSR MAY be for a newly generated key pair. |
| A02.FR.18 | A02.FR.17 | The Charging Station SHALL double the previous back-off time, starting with the number of seconds configured at CertSigningWaitMinimum, every time the back-off time expires without having received the CertificateSignedRequest for this CSR. |
| A02.FR.19 | A02.FR.18 AND<br>The maximum number of increments is reached | The Charging Station SHALL stop resending the SignCertificateRequest, until it is requested by the CSMS via a TriggerMessageRequest for `SignChargingStationCertificate`, `SignV2GCertificate` or `SignCombinedCertificate`. |
| A03.FR.17 | When the CSMS accepted the SignCertificateRequest for a CSR AND the Charging Station did not yet receive a CertificateSignedRequest for this CSR AND the number of seconds configured at CertSigningWaitMinimum has expired | The Charging Station SHALL send a new SignCertificateRequest for the CSR. Optionally, this CSR MAY be for a newly generated key pair. |
| A03.FR.18 | A03.FR.17 | The Charging Station SHALL double the previous back-off time, starting with the number of seconds configured at CertSigningWaitMinimum, every time the back-off time expires without having received the CertificateSignedRequest for this CSR. |
| A03.FR.19 | A03.FR.18 AND<br>The maximum number of increments is reached | The Charging Station SHALL stop resending the SignCertificateRequest, until it is requested by the CSMS via a TriggerMessageRequest for `SignChargingStationCertificate`, `SignV2GCertificate` or `SignCombinedCertificate`. |
| A02.FR.20 | A02.FR.07 | The Charging Station SHALL NOT initiate the back-off mechanism and resend the SignCertificateRequest, until this is requested by the CSMS via a TriggerMessageRequest for `SignChargingStationCertificate`, `SignV2GCertificate` or `SignCombinedCertificate`. |
| A02.FR.21 | When the Charging Station receives a SignCertificateResponse with status *Rejected*, in response to a SignCertificateRequest with certificateType *V2GCertificate* | It is RECOMMENDED to turn off ISO15118PnCEnabled until the Charging Station has been rebooted. |

*Changed requirements*

| Version | ID | Precondition | Requirements |
|---|---|---|---|
| Old | A02.FR.07 | If the certificate is not valid. | The Charging Station SHALL discard the certificate, and trigger an InvalidChargingStationCertificate security event (See part 2 appendices for the full list of security events). |
| New | A02.FR.07 | If the certificate is not valid. | The Charging Station SHALL **respond to the CertificateSignedRequest with status *Rejected* AND** discard the certificate AND trigger an *InvalidChargingStationCertificate* security event (See part 2 appendices for the full list of security events). |

| Version | ID | Precondition | Requirements |
|---|---|---|---|
| Old | A03.FR.07 | If the certificate is not valid. | The Charging Station SHALL discard the certificate, and trigger an InvalidChargingStationCertificate security event (See part 2 appendices for the full list of security events). |
| New | A03.FR.07 | If the certificate is not valid. | The Charging Station SHALL **respond to the CertificateSignedRequest with status *Rejected* AND** discard the certificate AND trigger an *InvalidChargingStationCertificate* security event (See part 2 appendices for the full list of security events). |

### 4.9.1. Page 427 - (v1) New SecurityCtrlr variables

`CertSigningWaitMinimum`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | SecurityCtrlr | |
| **Variable** | **variableName** | CertSigningWaitMinimum | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **unit** | seconds |
| | | **dataType** | integer |
| **Description** | This configuration variable defines how long the Charging Station has to wait before generating another CSR, in the case the CSMS accepts the SignCertificateRequest, but never returns the signed certificate back. This value will be doubled after every attempt. The amount of attempts is configured at CertSigningRepeatTimes If the certificate signing process is slow, this setting allows the CSMS to tell the Charging Station to allow more time. | | |

`CertSigningRepeatTimes`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | SecurityCtrlr | |
| **Variable** | **variableName** | CertSigningRepeatTimes | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | integer |
| **Description** | This variable can be used to configure the amount of times the Charging Station SHALL double the previous back-off time, starting with the number of seconds configured at CertSigningWaitMinimum, every time the back-off time expires without having received the CertificateSignedRequest containing the from the CSR generated signed certificate. When the maximum number of increments is reached, the Charging Station SHALL stop resending the SignCertificateRequest, until it is requested by the CSMS using a TriggerMessageRequest. | | |

## 4.10. Page 33 - (v1) Clarification for reconnection with new certificate [149]

### 4.10.1. Page 33 - (v1) A02.FR.08

| | | |
|---|---|---|
| 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. |
| New 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).** |

### 4.10.2. Page 36 - (v1) A03.FR.08

| | | |
|---|---|---|
| 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. |
| New 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).** |

# 5. Use case B Provisioning

## 5.1. Page 45 - (v1) Hearbeat interval must be greater than zero [370]

The heartbeat interval that is returned by CSMS in a BootNotificationResponse must obviously be greater than zero.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | B01.FR.04 | When the CSMS responds with BootNotificationResponse with the status *Accepted*. | The Charging Station SHALL adjust the heartbeat interval in accordance with the interval from the response message. | |
| New text | B01.FR.04 | When the CSMS responds with BootNotificationResponse with the status `Accepted` **AND**<br>***interval* > 0** | The Charging Station SHALL adjust the heartbeat interval in accordance with the interval from the response message. | |

### 5.1.1. Page 421 - (v1) Adding *minLimit* to HeartbeatInterval configuration variable

The description of `OCPPCommCtrlr.HeartbeatInterval` needs an addition to the *dataType* definition:

- *minLimit* = 1

This ensures that the heartbeat interval is always greater than zero.

## 5.2. Page 54 - (v1) Return first instance when no instance specified [476]

**WARNING** This errata entry has been replaced by Page 54 - (v2) Return only specified instance with GetVariablesRequest [608].

~~Chapter 4.1 in part 1 Topology & Architecture states: "Each distinct component instance is uniquely identified by an (optional) componentInstance addressing key. When no componentInstance is provided, then the default or only instance of a component is referenced."~~

~~However, a requirement for this is missing in B06 for GetVariables. This is especially important when a Charging Station provides an instance name to an EVSE as a way to name it:~~
~~e.g EVSE at evse = 1 with instance = "High power left".~~

~~When CSMS requests variables from EVSE at evse = 1, then Charging Station should return this instance, even if CSMS does not specify the instance, because this is the only instance. The same applies to variables.~~

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| ~~B06.FR.14~~ | B06.FR.01 AND<br>a value for *instance* is provided in the *component* and/or *variable* in GetVariableData | Only the specified instance of that component and/or variable is returned in GetVariableResult. |
| ~~B06.FR.15~~ | B06.FR.01 AND<br>no value or an empty string is provided for *instance* in the *component* and/or *variable* in GetVariableData | The first (or only) instance of that component and/or variable is returned in GetVariableResult. |

## 5.3. Page 54 - (v2) Return only specified instance with GetVariablesRequest [608]

The errata entry of Page 54 - (v1) Return first instance when no instance specified [476] is problematic, because the order of instances is not defined. It is therefore not possible to return a defined "first instance". It is also not possible to return all instances when no instance is provided, because of requirement B06.FR.01.

This leads to the following change:

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| B06.FR.14 | B06.FR.01 AND<br>a value for *instance* is provided in the *component* and/or *variable* in GetVariableData | Charging Station SHALL return the specified instance of that component and/or variable in GetVariableResult. |
| B06.FR.15 | B06.FR.01 AND<br>no value or an empty string is provided for *instance* in the *component* and/or *variable* in GetVariableData AND<br>a component and/or variable without an *instance* does not exist | Charging Station SHALL return the *attributeStatus* `UnknownComponent` or `UnknownVariable` in the GetVariableResult entry for GetVariableData. |

## 5.4. Page 55 - (v2) Use case B06: Requirement missing when exceeding ItemsPerMessageGetVariables [605]

Requirement B06.FR.05 states that CSMS shall not send more GetVariableData elements in a GetVariablesRequest then allowed by configuration variable DeviceDataCtrlr.ItemsPerMessage[GetVariables].

Similarly, the size of the message cannot be larger than DeviceDataCtrlr.BytesPerMessage[GetVariables].

Requirements for the charging station on how to respond to such a situation are missing. Since a memory-constrained charging station may not be able to properly process such a message, the requirement allows a charging station in such a situation to return a CALLERROR.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| B06.FR.16 | Charging Station receives a GetVariablesRequest with more GetVariableData elements than allowed by `ItemsPerMessageGetVariables` | The Charging Station MAY respond with a CALLERROR(OccurenceConstraintViolation) |
| B06.FR.17 | Charging Station receives a GetVariablesRequest with a length of more bytes than allowed by `BytesPerMessageGetVariables` | The Charging Station MAY respond with a CALLERROR(FormatViolation) |

## 5.5. Page 56 - (v1) Use case B07: Requirement for CSMS to request FullInventory is missing [511]

The flexibility of the device model depends on the fact that the charging station decides which components and variables are present and the CSMS discovers this by requesting a FullInventory report. This also ensures that variables like MaxElements or MaxBytes for Get/SetVariables are known. Requirements B07.FR.08 and B07.FR.11 require the charging station to report it to CSMS. So, implicitly the requirement for a CSMS to request the report is there, but it needs to be made explicit.

The requesting of a `FullInventory` report is typically performed during use cases "Cold Boot Charging Station" (B01 and B02).

*New requirements*

| ID | Precondition | Requirement definition | Remark |
|---|---|---|---|
| B07.FR.14 | When a Charging Station connects to CSMS for the first time OR whenever CSMS suspects that the device model of the Charging Station has changed (e.g. after a firmware update or hardware change) | CSMS SHOULD request a GetBaseReportRequest with *reportBase* = `FullInventory` to retrieve a complete list of all its device model components and variables. | It is not mandated, because implementations may exist that are based on a known set of charging stations with fixed device models that will not change. |

## 5.6. Page 56 - (v1) Use case B08 Get Custom Report: conflicting requirements [355]

Two following two requirements of use case B08 are conflicting:

| ID | Precondition | Requirement definition |
|---|---|---|
| B08.FR.01 | When the Charging Station receives a getReportRequest for supported *criteria* | The Charging Station SHALL send a getReportResponse with Accepted |
| B08.FR.15 | When the Charging Station receives a GetReportRequest with a combination of criteria which results in an empty result set. | The Charging Station SHALL respond with a GetReportResponse(*status*=`EmptyResultSet`). |

To fix this, the precodition of requirement B08.FR.01 needs to be changed as follows:

*Changed requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| B08.FR.01 | **NOT B08.FR.15 AND**<br>When the Charging Station receives a getReportRequest for supported *criteria* | The Charging Station SHALL send a getReportResponse with Accepted |

## 5.7. Page 59 - (v2) Use case B08: Requirement missing when exceeding ItemsPerMessageGetReport [615]

Requirement B08.FR.06 states that CSMS shall not send more ComponentVariableType elements in a GetReportRequest then allowed by configuration variable DeviceDataCtrlr.ItemsPerMessage[GetReport].

Similarly, the size of the message cannot be larger than DeviceDataCtrlr.BytesPerMessage[GetReport].

Requirements for the charging station on how to respond to such a situation are missing. Since a memory-constrained charging station may not be able to properly process such a message, the requirement allows a charging station in such a situation to return a CALLERROR.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| B08.FR.17 | Charging Station receives a GetReportRequest with more ComponentVariableType elements than allowed by `ItemsPerMessageGetReport` | The Charging Station MAY respond with a CALLERROR(OccurenceConstraintViolation) |
| B08.FR.18 | Charging Station receives a GetReportRequest with a length of more bytes than allowed by `BytesPerMessageGetReport` | The Charging Station MAY respond with a CALLERROR(FormatViolation) |

## 5.8. Page 59 - (v2) Use case B08: GetReport handling of empty variables or instances [565]

The GetReportRequest takes *componentCriteria* and *componentVariables* as filters that limit the scope of the report. (See for example B08.FR.05, B08.FR.11). This implies that if no *variable* is specified, all variables of the component are reported. Similarly, if no *instance* is specified, then all instances of component or variable are reported, and if no *evse* is provided, than components over all EVSEs are reported.

This is made explicit in the following requirements.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| B08.FR.19 | When Charging Station receives a GetReportRequest with *componentVariable* elements in which *component.instance* and/or *component.evse* are missing | The Charging Station SHALL report for every instance and/or EVSE of the *component* in *componentVariable*. |
| B08.FR.20 | When Charging Station receives a GetReportRequest with *componentVariable* elements in which *variable* is missing | The Charging Station SHALL report for every *variable* of the *component* in *componentVariable*. |
| B08.FR.21 | When Charging Station receives a GetReportRequest with *componentVariable* elements in which *variable* is present, but *instance* is missing | The Charging Station SHALL report for every instance of the *variable* of the *component* in *componentVariable*. |

## 5.9. Page 62 - (v2) Use case B10: Requirement note should have been a separate requirement and missing preconditions

The note of requirement B10.FR.03 will be removed and a separate requirement for this will be created, that makes it more clear. And preconditions will be added to requirement B10.FR.03 and B10.FR.04.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | B10.FR.03 | When connecting fails | The Charging Station SHALL make the number of attempts as configured in `NetworkProfileConnectionAttempts` per entry of `NetworkConfigurationPriority`. | If after the number of attempts the connection fails, the Charging Station should go back to the old NetworkConnectionProfile. |
| New text | B10.FR.03 | **B10.FR.04 AND**<br>When connecting fails | The Charging Station SHALL make the number of attempts as configured in `NetworkProfileConnectionAttempts` per entry of `NetworkConfigurationPriority`. | |
| Old text | B10.FR.04 | After a reboot | The Charging Station SHALL begin connecting to the first entry of `NetworkConfigurationPriority` | |
| New text | B10.FR.04 | **B10.FR.01 OR B09.FR.01 AND**<br>After a reboot | The Charging Station SHALL begin connecting to the first entry of `NetworkConfigurationPriority` | |

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| B10.FR.07 | B10.FR.03 AND<br>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". |

## 5.10. Page 63 - (v2) Use case B11: allow Unavailable notification during reset [581]

Some charging stations automatically send a StatusNotification(Unavailable) when they perform a reset. This is allowed as long as the status is restored after the reset.

*Changed requirements*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | B11.FR.03 | B11.FR.01<br>AND no *evseId* parameter is supplied<br>AND<br>ResetResponse was *Accepted*. | The Charging Station SHALL start a reboot. |
| New text | B11.FR.03 | B11.FR.01<br>AND no *evseId* parameter is supplied<br>AND<br>ResetResponse was *Accepted*. | The Charging Station **MAY send a StatusNotification(Unavailable) and** SHALL start a reboot. |
| Old text | B11.FR.08 | B11.FR.01<br>AND an *evseId* parameter is supplied<br>AND<br>ResetResponse was *Accepted*. | The Charging Station SHALL start a reboot of EVSE that is referred to by *evseId* parameter. |
| New text | B11.FR.08 | B11.FR.01<br>AND an *evseId* parameter is supplied<br>AND<br>ResetResponse was *Accepted*. | The Charging Station **MAY send a StatusNotification(Unavailable) for the EVSE and** SHALL start a reboot of EVSE that is referred to by *evseId* parameter. |

## 5.11. Page 66 - (v1) Incomplete preconditions in B12 [442]

The use case is about reset during ongoing transaction, but this is not mentioned as a precondition.

*Changed requirements*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | B12.FR.01 | When the Charging Station receives a ResetRequest(OnIdle) | The Charging Station SHALL respond with a ResetResponse(Scheduled), to indicate whether the Charging Station will attempt to reset itself or EVSE after all transactions on Charging Station or EVSE have ended. |
| New text | B12.FR.01 | When the Charging Station receives a ResetRequest(OnIdle)<br>**AND a transaction is ongoing** | The Charging Station SHALL respond with a ResetResponse(Scheduled), to indicate whether the Charging Station will attempt to reset itself or EVSE after all transactions on Charging Station or EVSE have ended. |
| Old text | B12.FR.02 | When the Charging Station receives a ResetRequest(Immediate) | The Charging Station SHALL respond with a ResetResponse(Accepted), to indicate whether the Charging Station will attempt to reset itself or EVSE. |
| New text | B12.FR.02 | When the Charging Station receives a ResetRequest(Immediate)<br>**AND a transaction is ongoing** | The Charging Station SHALL respond with a ResetResponse(Accepted), to indicate whether the Charging Station will attempt to reset itself or EVSE. |

# 6. Use case C Authorization

## 6.1. Page 74 - (v1) Part of requirement is actually a precondition [439]

The description of requirement C01.FR.03 contains conditions, which belong in the precondition column. The phrase "When stopping a transaction" is not quite correct, because it is about presenting an idToken with the intention to stop charging.

*Changed requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | C01.FR.03 | When stopping a transaction | The Charging Station SHALL NOT send an AuthorizeRequest when<br>(a) the IdToken used for stopping the transaction is the same as the IdToken that started the transaction OR<br>(b) when the IdToken used for stopping the transaction is in the Local Authorization List or the Authorization Cache AND is valid AND has the same GroupIdToken as the IdToken that started the transaction. |  |
| New text | C01.FR.03 | When an idToken is presented during a transaction that has been authorized<br>AND<br>(a) the presented idToken is the same as the idToken that started the authorization<br>OR<br>(b) when the presented idToken is in the Local Authorization List or Authorization Cache AND is valid AND has the same GroupIdToken as the IdToken that started the authorization. | The Charging Station SHALL end the authorization of the transaction, without first sending an AuthorizeRequest | The idToken that started the authorization can always be used to end the authorization.<br>Ending authorization will end delivery of energy. Depending on the TxStopPoint ending of the authorization may also end the transaction. |

## 6.2. Page 74 - (v2) C01.FR.06 AuthorizeResponse must return groupIdToken [641]

The definition of C01.FR.06 used the verb "MAY", because a *groupIdToken* does not have to exist for an *idToken*. However, this can be mistaken to mean that it is completely optional to return the *groupIdToken*. If it exists, then it must be returned, so that the charging station can enter this information in the local authorization cache.

*Change requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | C01.FR.06 |  | AuthorizeResponse sent by the CSMS to a Charging Station MAY include groupIdToken. |  |
| New text | C01.FR.06 | **When CSMS receives an AuthorizeRequest for an *idToken***<br>**AND**<br>**the idToken has an associated groupIdToken.** | AuthorizeResponse sent by the CSMS to a Charging Station **SHALL include the associated groupIdToken.** |  |

## 6.3. Page 75 - (v1) Requirement to avoid authorizing multiple idTokens in a transaction [514]

If the `TxStopPoint` does not contain **Authorized**, then the transaction remains active after charging was stopped by presenting the *idToken* for a second time. There is no requirement that forbids that a new authorization is done for **another** *idToken* than the original one. This was never intended and will most likely not be supported by a CSMS. We therefore add a requirement to disallow authorization of a multiple *idTokens* during a transaction.

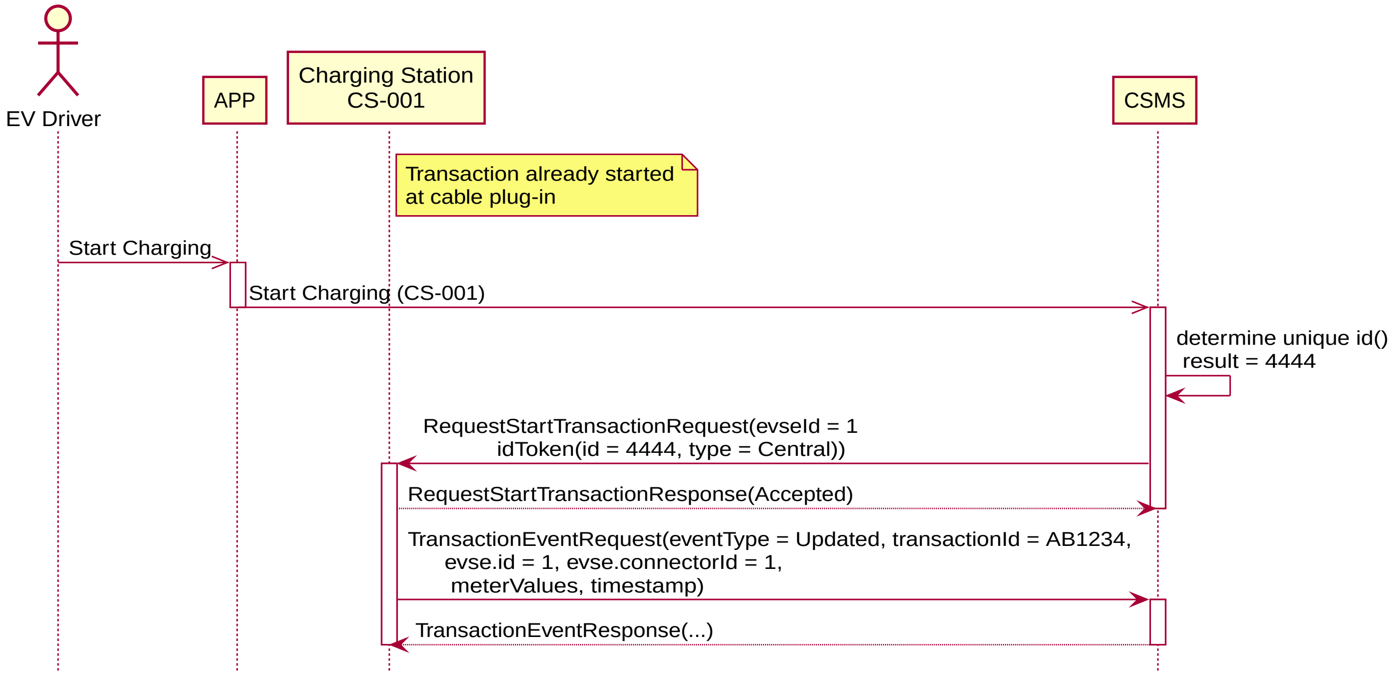
**NOTE** This errata is closely related to [jira514], which deals with the situation when a new *idToken* is authorized, despite the requirements in this section.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| C01.FR.23 | When a transaction is still active, that had been authorized earlier by an idToken, but which is now no longer authorized for charging AND<br>a new idToken is presented to the Charging Station for authorization, that **differs** from the inital idToken | The Charging Station SHOULD not allow the authorization of a different idToken. | Multiple *idTokens* for a transaction are most likely not supported by a CSMS. |
| C01.FR.24 | When a transaction is still active, that had been authorized earlier by an idToken, but which is now no longer authorized for charging AND<br>Charging Stations sends an AuthorizeRequest for a new idToken, that **differs** from the inital idToken of the transaction | The CSMS is RECOMMENDED to respond with an AuthorizeResponse with *idTokenInfo.status* = `NotAtThisTime` for this idToken. | If a second authorization is done by Charging Station then CSMS can reject the *idToken*. |

## 6.4. Page 83 - (v1) Sequence diagram use case C05 [435]

Sequence diagram Figure 26 shows a StatusNotification(Occupied) being sent after the RequestStartTransaction. This is not correct, because cable has already been plugged-in this use case begins. The sequence diagram has been modified to reflect this.


EV Driver
APP
Charging Station
CS-001
CSMS
Transaction already started
at cable plug-in
Start Charging
Start Charging (CS-001)
determine unique id()
result = 4444
RequestStartTransactionRequest(evseId = 1
idToken(id = 4444, type = Central))
RequestStartTransactionResponse(Accepted)
TransactionEventRequest(eventType = Updated, transactionId = AB1234,
evse.id = 1, evse.connectorId = 1,
meterValues, timestamp)
TransactionEventResponse(...)

## 6.5. Page 84 - (v1) Description and remarks of use case C05 [503, 504]

### 6.5.1. Error in description of use case C05

The description mentioned that a TransactionEvent(Updated) was sent to notify that the cable had been plugged in, but that is not correct, because cable was already connected prior to the use case.

Step #6 is removed and step #7 has changed.

| No. | Type | Description |
|---|---|---|
| | *Scenario description* | **1.** The EV Driver uses his app to start a charging.<br>**2.** The app sends a start request to the CSMS.<br>**3.** The CSMS determines an IdToken. It can generate a unique id to be used as IdToken for this transaction or can use a token that is provided by the app (for example the ID of the contract of the user).<br>**4.** The CSMS sends a RequestStartTransactionRequest with the IdToken from the previous step to the Charging Station.<br>**5.** The Charging Station accepts the RequestStartTransactionRequest by sending a RequestStartTransactionResponse with Accepted.<br>**7. The Charging Station starts charging and sends a TransactionEventRequest (eventType = Updated) to notify the CSMS that *chargingState* has changed.** |

## 6.5.2. Remarks use case C05

Add the following text to the remark of use case C05.

| | | |
|---|---|---|
| 8 | **Remarks** | [...]<br><br>**This use case assumes that the configuration variable AuthorizeRemoteStart is *false*. See use cases F01 and F02 for requirements with AuthorizeRemoteStart.**<br><br>**Other idTokenTypes can also be used to remote start charging, such an *eMAID* of the user that is provided by the app.** |

# 6.6. Page 89 - (v2) Use case C07: *certificate* in AuthorizeRequest contains certificate chain [533]

The field *certificate* in AuthorizeRequest needs to contain the certificate chain and not just the contract certificate. This is suggested by C07.FR.02, but not clear in C07.FR.06 and the field description of AuthorizeRequest.

## 6.6.1. Page 89 - C07 requirements

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | C07.FR.06 | C07.FR.01 AND<br>If Charging Station is not able to validate a contract certificate, because it does not have the associated root certificate AND `CentralContractValidationAllowed` is *true* | The Charging Station SHALL pass the contract certificate to the CSMS in *certificate* attribute (in PEM format) of AuthorizeRequest for validation by CSMS. |
| New text | C07.FR.06 | C07.FR.01 AND<br>If Charging Station is not able to validate a contract certificate, because it does not have the associated root certificate AND `CentralContractValidationAllowed` is *true* | The Charging Station SHALL pass the contract certificate **chain** to the CSMS in *certificate* attribute (in PEM format) of AuthorizeRequest for validation by CSMS. |

## 6.6.2. Page 338 - AuthorizeRequest

The description of field *certificate* is updated as follows:

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | certificate | string[0..5500] | 0..1 | Optional. The X.509 certificated presented by EV and encoded in PEM format. |

|  | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| New text | certificate | string[0..5500] | 0..1 | Optional. The X.509 **certificate chain** presented by EV and encoded in PEM format. **Order of certificates in chain is from leaf up to (but excluding) root certificate.** |

## 6.7. Page 94 - (v2) Improved requirements for GroupIdToken [601]

The requirements of C09 for handling an idToken authorization based on a matching groupIdToken imply that the idToken that started the transaction must exist in the authorization cache or the local authorization list, but that does not have to be the case.

- Requirement C09.FR.06 is superfluous as this is already covered by C01.FR.02.
- Requirement C09.FR.08 is superfluous as this is already covered by C01.FR.05.

*Deleted requirements*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Deleted | C09.FR.06 | If an idToken presented by the EV Driver is not present in the Local Authorization List or Authorization Cache | The Charging Station SHALL send AuthorizeRequest to the CSMS to request authorization. |
| Deleted | C09.FR.08 | If an IdToken is present in the Local Authorization List or Authorization Cache. | The Charging Station MAY send AuthorizeRequest to the CSMS. |

- Requirement C09.FR.05 is updated to reflect that only the newly presented idToken requires an AuthorizeRequest and only in case C09.FR.07 does not apply.
- C09.FR.07 has been reformulated in the same way as C01.FR.03 (see Page 74 - (v1) Part of requirement is actually a precondition [439].
- Requirement C09.FR.09 used "MAY" instead of "SHALL", because not every idToken has a groupIdToken. However, the intention is, that if there is a groupIdToken that it shall be returned.
- The above also applies to a TransactionEventResponse, for which a new requirement has been added to make this explicit.

*Changed requirements*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | C09.FR.05 | C09.FR.03 AND<br>If NOT both IdTokens with their corresponding GroupIdTokens are present in either the Local Authorization List or Authorization Cache. | The Charging Station SHALL send an AuthorizeRequest to the CSMS. |
| New text | C09.FR.05 | C09.FR.03 AND<br>**(NOT C09.FR.07) AND**<br>**If the newly presented IdToken with its corresponding GroupIdToken is not** present in either the Local Authorization List or Authorization Cache. | The Charging Station SHALL send an AuthorizeRequest to the CSMS. |
| Old text | C09.FR.07 | C09.FR.03 | The Charging Station SHALL NOT send an AuthorizeRequest when<br>(a) the IdToken used for stopping the transaction is the same as the IdToken that started the transaction OR<br>(b) when the IdToken used for stopping the transaction is in the Local Authorization List or the Authorization Cache AND is valid AND has the same GroupIdToken as the IdToken that started the transaction. |

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| New text | C09.FR.07 | When an idToken is presented during a transaction that has been authorized AND (a) the presented idToken is the same as the idToken that started the authorization<br>OR<br>(b) when the presented idToken is in the Local Authorization List or Authorization Cache AND is valid AND has the same GroupIdToken as the IdToken that started the authorization. | The Charging Station SHALL end the authorization of the transaction, without first sending an AuthorizeRequest |
| Old text | C09.FR.09 | If the CSMS accepts the IdToken. | AuthorizeResponse MAY include groupIdToken. |
| New text | C09.FR.09 | If the **IdToken in AuthorizeRequest has an associated groupIdToken** | AuthorizeResponse **from CSMS SHALL** include groupIdToken. |

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| C09.FR.12 | If a TransactionEventRequest contains an IdToken and idToken has an associated groupIdToken | TransactionEventResponse from CSMS SHALL include groupIdToken. |

## 6.8. Page 94 - (v1) Requirement C09.FR.11 is partly incorrect [513]

Requirement C09.FR.11 contains an error. When a token is presented to stop a transaction and the token is not authorized to do so, then stopping will simply be refused. No authorization status value will be returned, because there was no request message.

**NOTE** If the presented token does not exist in the authorization cache, then the charging station may send an AuthorizeRequest and in response to that, CSMS will send an authorization status value. This is covered by requirements in C01, like C01.FR.06 and C01.FR.07

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | C09.FR.11 | C09.FR.03 AND<br>A different IdToken is presented for stopping, which has the same GroupIdToken, but does not have status = `Accepted` | The Charging Station SHALL NOT stop the transaction and SHALL return an authorization status value indicating a reason for rejection. |
| New text | C09.FR.11 | C09.FR.03 AND<br>A different IdToken is presented for stopping, which has the same GroupIdToken, but does not have status = `Accepted` | **The Charging Station SHALL NOT stop the transaction.** |

## 6.9. Page 101 - (v2) C13.FR.02/04 are about authorizing, instead of starting [640]

Requirements C13.FR.02 and C13.FR.04 mention "allowed to start a transaction", but this must be "allowed to authorize a transaction", because the transaction may have started already, for example when the TxStartPoint contains `EVConnected` or `ParkingBayOccupancy`.

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | C13.FR.02 | If configuration variable `OfflineTxForUnknownIdEnabled` is false AND<br>The Charging Station is offline. | Only identifiers that are present in a Local Authorization List that have a status *Accepted* SHALL be allowed to start a transaction. | |

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New text | C13.FR.02 | If configuration variable `OfflineTxForUnknownIdEnabled` is false AND<br>The Charging Station is offline. | Only identifiers that are present in a Local Authorization List that have a status *Accepted* SHALL be allowed to **authorize** a transaction. | |
| Old text | C13.FR.04 | If configuration variable `OfflineTxForUnknownIdEnabled` is true AND<br>The Charging Station is offline. | Any identifier SHALL be allowed to start a transaction. | |
| New text | C13.FR.04 | If configuration variable `OfflineTxForUnknownIdEnabled` is true AND<br>The Charging Station is offline. | Any identifier SHALL be allowed to **authorize** a transaction. | |

## 6.10. Page 105 - (v1) ChargingState in requirement C15.FR.03 [464]

**WARNING** This entry has been updated in [jira-647].

This requirement states the *chargingState* should be `SuspendedEVSE` when the transaction is not stopped, but deauthorized, because the id token is not valid. If, for example, TxStopPoint is `EVConnected` the transaction remains active, but no more energy should be delivered.

However, the *chargingState* `SuspendedEVSE` is not quite correct when the transaction has become deauthorized. The *chargingState* should go back to `EVConnected`, because there is no authorization anymore.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | C15.FR.03 | C15.FR.02 AND<br>The authorization status in TransactionEventResponse is not *Accepted* AND<br>The transaction is still ongoing AND<br>`StopTxOnInvalidId` is *true* AND<br>`TxStopPoint` does NOT contain: (Authorized OR PowerPathClosed OR EnergyTransfer) | The Charging Station SHALL stop the energy transfer and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *Deauthorized* and chargingState set to *SuspendedEVSE*. | |
| New text | C15.FR.03 | C15.FR.02 AND<br>The authorization status in TransactionEventResponse is not *Accepted* AND<br>The transaction is still ongoing AND<br>`StopTxOnInvalidId` is *true* AND<br>`TxStopPoint` does NOT contain: (Authorized OR PowerPathClosed OR EnergyTransfer) | The Charging Station SHALL stop the energy transfer and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to `Deauthorized` and *chargingState* set to `SuspendedEVSE` **or preferably to `EVConnected`.** | Since the effect of setting *chargingState* to `SuspendedEVSE` or `EVConnected` both have the same effect of not delivering any energy, the use of `SuspendedEVSE` is still allowed in this situation in order to avoid breaking existing implementations that adhere to the original requirement.<br>Use of `SuspendedEVSE` in this situation will become deprecated in the next OCPP release. |

## 6.11. Page 105 - (v2) Incorrect TxStartPoint PowerPathClosed reference in requirement precondition

Requirement C15.FR.06 is incorrectly referencing to TxStartPoint PowerPathClosed in its precondition.

*Changed requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | C15.FR.06 | C15.FR.02 AND The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is set to *false* AND `MaxEnergyOnInvalidId` is not implemented or has been exceeded. `TxStopPoint` does NOT contain: (PowerPathClosed OR EnergyTransfer) | The Charging Station SHALL stop the energy delivery to the EV immediately and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *ChargingStateChanged* and chargingState set to *SuspendedEVSE* |  |
| New text | C15.FR.06 | C15.FR.02 AND The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is set to *false* AND `MaxEnergyOnInvalidId` is not implemented or has been exceeded. `TxStopPoint` does NOT contain: EnergyTransfer | The Charging Station SHALL stop the energy delivery to the EV immediately and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *ChargingStateChanged* and chargingState set to *SuspendedEVSE* |  |

# 7. Use case E Transactions

## 7.1. Page 114 - (v2) Improved definition of transaction [538]

In section E.1 Introduction add the following text in section 1.1 "Flexible transaction start/stop" just before the start of section 1.1.1.

> With the introduction in OCPP 2.0.1 of flexible start/stop points of a transaction, it is important to provide a definition of a transaction.
>
> > A transaction is the portion of a charging session that is recorded by CSMS. It is a single time frame with a start and stop time. This information can be used by the operator for billing.
>
> It is up to the Charging Station Operator to define the values for TxStartPoint and TxStopPoint (unless these are preset as read-only values in the charging station), but not all combinations make sense.
>
> The following three variants are most common:
>
> - If connection time is billed, then start and stop points should be `EVConnected`.
> - If time of use is billed, then the start points should be `EVConnected, Authorized` and the stop point `EVConnected`. (Such that upon authorization first, the charger is already seen as 'in use').
> - If charging time is billed, then start and stop points should be `PowerPathClosed`. (This starts as soon as charger is ready to provide power and stops when authorization is revoked or vehicle disconnected.) Pauses in between (i.e. SuspendedEV(SE)) do not end the transaction. Billing on the amount of energy or power can be done based on the meter values that are collected during the transaction.

| | |
|---|---|
| **WARNING** | Certain combinations of start and stop points can lead to a situation where a started transaction is never stopped. For example: when the start point is `ParkingBayOccupancy` and the stop point is `EVConnected`, then a transaction starts when an EV occupies the parking bay, but when the user never connects the EV, but simply drives away, then the transaction will remain open, because `ParkingBayOccupancy` is not configured as a stop point. |

## 7.2. Page 115 - (v1) Setting for OCPP 1.6 transaction compatibility [516]

The description in section E1.1.2 "OCPP 1.6 Transaction compatibility" suggests that the TxStartPoint and TxStopPoint `PowerPathClosed` refer to the closing and opening of the power relay. This is not correct. See Page 434 - (v1) Better description of TxStartPoint/TxStopPoint [348] for a better description of `PowerPathClosed`.

Replace the following paragraph:

| | |
|---|---|
| Old | In OCPP 1.x the moment a Charging Station should send StartTransaction.req was not defined very precise, generally this was done when the power path was closed: relay closed. Which should only be done after authorization. |
| New | In OCPP 1.x the moment a Charging Station should send StartTransaction.req was not defined very precise, generally this was done when **the Charging Station was ready to deliver energy: cable is connected and user is authorized.** |

Table 95 "The settings for an OCPP 1.6 compatible transaction" must be replaced by:

| | Configuration Variable | Values |
|---|---|---|
| Old | `TxStartPoint` | PowerPathClosed,EnergyTransfer |
| Old | `TxStopPoint` | EVConnected,Authorized,DataSigned,PowerPathClosed |
| New | `TxStartPoint` | PowerPathClosed |
| New | `TxStopPoint` | EVConnected, Authorized |

The addition of 'EnergyTransfer' in the TxStartPoint is optional and may be used for Charging Stations that do not require authorization to deliver energy.

The 'PowerPathClosed' condition as a TxStartPoint applies when both 'EVConnected' and 'Authorized' are true. The 'PowerPathClosed' condition as a TxStopPoint applies when either 'EVConnected' or 'Authorize' are false (or both are false). When 'EVConnected' and 'Authorized' are already present in TxStopPoint,then there is no need to add 'PowerPathClosed'.

## 7.3. Page 116 - (v1) Using *seqNo* in TransactionEventRequest when EVSE is not known [525]

| | |
|---|---|
| **WARNING** | This errata entry has been superseded by Page 116 - (v2) Solution for ever-increasing seqNo per EVSE may lead to race condition and eventual overflow [592] |

Section E 1.3.2 mentions the following about sequence numbers in TransactionEventRequests:

> In order to make it possible to know that all TransactionEventRequest messages about a transaction were received, OCPP uses *sequence numbers* in TransactionEventRequest messages. For every EVSE, the Charging Station maintains a counter of the number of TransactionEventRequest messages generated about that EVSE. When generating a new TransactionEventRequest message, the Charging Station includes the current value of the EVSE’s counter in the **seqNo** field of the request, and then increments the counter. With this mechanism, a CSMS can check if it has full information about a transaction by checking that:
>
> - It received a TransactionEventRequest about the start of the transaction, with a **seqNo** *a*
> - It received a TransactionEventRequest about the stop of the transaction, with a **seqNo** *o* greater than *a*.
> - It received a TransactionEventRequest about the transaction with **seqNo** *n* for every integer *n* between *a* and *o*

This implies that a separate *seqNo* counter is maintained for each EVSE. However, a transaction start point can be configured such that an EVSE is not yet known at the moment when a transaction starts, for example in the case where `TxStartPoint` is Authorized or ParkingBayOccupancy.

A solution for this, which does not conflict with above-mentioned requirements, is the following:

**seqNo start value**

If a transaction starts when the EVSE is not yet known, then use the largest *seqNo* counter among all EVSEs of the charging station as starting value for *seqNo* of the new transaction.

This may lead to 'holes' in the sequence numbering **between** transactions on some EVSEs, but it does not violate the requirements. All *seqNo* from start to end of a transaction will be continuously increasing and there are no duplicate *seqNo* on the EVSE.

| | |
|---|---|
| **NOTE** | Selecting the highest *seqNo* potentially goes wrong if one of the *seqNo* counters has wrapped around to 0 after reaching the maximimum counter value of 2,147,483,647. However, even when a TransactionEvent would be sent every second, it will still take 68 years before this maximum is reached. |

## 7.4. Page 116 - (v2) Solution for ever-increasing seqNo per EVSE may lead to race condition and eventual overflow [592]

The solution described in Page 116 - (v1) Using *seqNo* in TransactionEventRequest when EVSE is not known [525] has shown to be unpractical to implement and may eventually lead to a sequence number overflow. Since there is no good reason why transaction sequence numbers should be ever-increasing across transactions, we have decided to simplify this counter by letting each transaction start with sequence number zero (0).

This means that section 1.3.2.1 Sequence number generation is changed as follows:

> 1.3.2.1 Sequence number generation
>
> *This section is normative.*
>
> When a transaction starts, the Charging Station SHOULD set the *seqNo* field for the TransactionEventRequest message to 0. (Implementations with a continuously increasing *seqNo* are still allowed.)
>
> After each TransactionEventRequest Charging Station SHALL increase the *seqNo* by 1.

## 7.5. Page 116 - (v2) Clarification for optional fields in TransactionEventRequest [101]

*This section is informative.*

The TransactionEventRequest contains several optional fields. Some of these fields should only be sent once and should not be repeated in every TransactionEventRequest. The following summary points to the requirements related to these optional fields. It does not change any requirements.

***evse***

(E01.FR.16) The field *evse* is only provided in the first TransactionEventRequest that occurs after the EV has connected. It is not repeated in all future TransactionEventRequests.

***idToken***

(E03.FR.01) The field *idToken* is provided once in the first TransactionEventRequest that occurs after the transaction has been authorized.
(E07.FR.02) The field *idToken* is provided once in the TransactionEventRequest that occurs when the authorization of the transaction has been ended.
(C12.FR.02) The above is also the case when authorization was granted because the idToken is present in the authorization cache with a `Accepted` status.
(F02.FR.05): The above is also the case when the idToken is provided by a RequestStartTransactionRequest.

***reservationId***

(E03.FR.03/H01.FR.15) The field *reservationId* is only provided in the first TransactionEventRequest that occurs when the transaction has been authorized by the idToken for which a reservation existed in the charging station.
(F02.FR.06) The above is also the case when the idToken is provided by a RequestStartTransactionRequest.

***meterValue***

(E02.FR.09) The TransactionEventRequest(*eventType*=`Started`) must contain the meter values that have been configured in SampledDataCtrlr.TxStartedMeasurands.
(E02.FR.10) A TransactionEventRequest(*eventType*=`Updated`) must be sent at every interval configured in SampledDataCtrlr.TxUpdatedInterval and contain the meter values that have been configured in SampledDataCtrlr.TxUpdatedMeasurands. If TxUpdatedMeasurands == 0, then no meter values are sent.
(E06.FR.11) The TransactionEventRequest(*eventType*=`Ended`) must contain the meter values that have been configured in SampledDataCtrlr.TxEndedMeasurands. If SampledDataCtrlr.TxEndedInterval == 0, then only the values taken at start and end of the transaction are included.

***transactionInfo.chargingState***

(E02.FR.13) Whenever the charging state changes, the Charging Station must send a TransactionEventRequest containing *chargingState*.
A TransactionEventRequest with *triggerReason* = `ChargingStateChanged` must contain *chargingState*.

***transactionInfo.stoppedReason***

(C15.FR.04, E05.FR.10, E05.FR.08/09, E07.FR.06) The *stoppedReason* must be provided in the TransactionEventRequest(*eventType*=`Ended`), unless the value is `Local`, in which case it may be omitted.
(F03.FR.03, F03.FR.10, F04.FR.03) The above also applies to transactions that are stopped by a RequestStopTransactionRequest, however in this case the *stoppedReason* value must be `Remote`.

***transactionInfo.remoteStartId***

(C05.FR.03, F01.FR.25, F02.FR.01) The *remoteStartId* must be sent in the next TransactionEventRequest after the RequestStartTransactionRequest with the same *remoteStartId*.

## 7.6. Page 119 - (v2) Use case E: references to AlignedDataSignReadings instead of SampledDataSignReadings [626]

The text in all use cases of chapter E Transactions is referring to the configuration variable `AlignedDataSignReadings` where it must be `SampledDataSignReadings`.

The configuration variable `AlignedDataSignReadings` is replaced by `SampledDataSignReadings` in the following places:

### 7.6.1. Page 119 - Scenario S4

| **S4** | *Scenario objective* | **To start a transaction when the meter has provided the first signed meter values before starting with charging.** |
|---|---|---|
| ... | | |
| | **Prerequisite(s)** | No transaction is ongoing on the EVSE.<br>Configuration Variable: `TxStartPoint` contains: DataSigned (Not: ParkingBayOccupancy, EVConnected or Authorized).<br>The Charging Station has a meter that can sign measured values<br>Configuration Variable: ~~`AlignedDataSignReadings`~~ `SampledDataSignReadings` set to *true*. |

### 7.6.2. Page 122 - Requirement E01.FR.04

| ID | Precondition | Requirement definition |
|---|---|---|
| E01.FR.04 | `TxStartPoint` contains: DataSigned<br>AND<br>The Charging Station has a meter that can sign measured values<br>AND<br>Configuration Variable:<br>~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings`set to *true*.<br>AND<br>The Charging Station has retrieved a signed meter value<br>AND<br>No transaction has started yet | The Charging Station SHALL start a transaction and send a TransactionEventRequest (eventType = Started) to the CSMS. |

## 7.6.3. Page 126 - Requirement E02.FR.14

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E02.FR.14 | ~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

## 7.6.4. Page 130 - Requirement E03.FR.10

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E03.FR.10 | ~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

## 7.6.5. Page 133 - Requirement E04.FR.10

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E04.FR.10 | ~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

## 7.6.6. Page 136 - Requirement E05.FR.06

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E05.FR.06 | ~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

## 7.6.7. Page 145 - Requirement E07.FR.12

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E07.FR.12 | ~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

## 7.6.8. Page 149 - Requirement E08.FR.12

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E08.FR.12 | ~~`AlignedDataSignReadings`~~<br>`SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

### 7.6.9. Page 152 - Requirement E09.FR.08

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E09.FR.08 | ~~`AlignedDataSignReadings`~~ `SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

### 7.6.10. Page 155 - Requirement E10.FR.07

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E10.FR.07 | ~~`AlignedDataSignReadings`~~ `SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. | |

### 7.6.11. Page 157 - Requirement E11.FR.08

| ID | Precondition | Requirement definition |
|---|---|---|
| E11.FR.08 | ~~`AlignedDataSignReadings`~~ `SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. |

### 7.6.12. Page 159 - Requirement E12.FR.10

| ID | Precondition | Requirement definition |
|---|---|---|
| E12.FR.10 | ~~`AlignedDataSignReadings`~~ `SampledDataSignReadings` is *true* | The Charging Station SHALL retrieve signed meter values and put them in the *signedMeterValue* field of sampledValues. |

## 7.7. Page 122 - (v2) Obsolete references to power relay for PowerPathClosed [600]

PowerPathClosed has been referred to as "power relay closed/opened" at several locations in the specification. This association is wrong, since PowerPathClosed is the logical "AND" of the TxStartPoints "EVConnected" and "Authorized", which together are required to allow delivery of power.

### 7.7.1. Page 120 - Use case E01: S5 Scenario objective

The table describing scenario is referring to "power relay closed". The first two rows are replaced as follows:

| | | | |
|---|---|---|---|
| Old text | S5 | *Scenario objective* | To start a transaction when all preconditions are available to start charging, but energy does not yet have to be transfered (for example: power relay closed). |
| New text | S5 | *Scenario objective* | To start a transaction when all preconditions **have been met to start charging (authorized and connected), but energy does not yet have to be transfered.** |
| Old text | | *Scenario description* | **1.** The EV Driver is authorized by the Charging Station and/or CSMS.<br>**2.** The Charging Station closes the power relay.<br>**3.** The Charging Station sends a TransactionEventRequest (eventType = Started) notifying the CSMS about a transaction that has started.<br>**4.** The CSMS responds with a TransactionEventResponse, confirming that the TransactionEventRequest was received. |
| New text | | *Scenario description* | **1.** The EV Driver is authorized by the Charging Station and/or CSMS.<br>**2.** The Charging Station **is connected to the EV**<br>**3.** The Charging Station sends a TransactionEventRequest (eventType = Started) notifying the CSMS about a transaction that has started.<br>**4.** The CSMS responds with a TransactionEventResponse, confirming that the TransactionEventRequest was received. |

The text "close power relay" and self-referencing arrow in Figure 44 must be removed.

## 7.7.2. Page 122 - E01.FR.05

PowerPathClosed is treated as the logical AND of "EVConnected" and "Authorized".

| **NOTE** | PowerPathClosed == EVConnected && Authorized |
|---|---|

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | E01.FR.05 | `TxStartPoint` contains:<br>PowerPathClosed<br>AND<br>The Charging Station closes the power relay<br>AND<br>No transaction has started yet on this EVSE | The Charging Station SHALL start a transaction and send a TransactionEventRequest (eventType = Started) to the CSMS. |
| New text | E01.FR.05 | `TxStartPoint` contains:<br>PowerPathClosed<br>AND<br>**The EV Driver is authorized AND The Charging Station has connection with the EV**<br>**AND**<br>No transaction has started yet on this EVSE | The Charging Station SHALL start a transaction and send a TransactionEventRequest (eventType = Started) to the CSMS. |

## 7.7.3. Page 141 - Use case E06: Scenario S5

The table describing scenario is referring to "power relay opened". The first two rows are replaced as follows:

| | | | |
|---|---|---|---|
| Old text | S5 | *Scenario objective* | Stop a transaction when the power path is no longer closed. (For example: power relay opened.) |
| New text | S5 | *Scenario objective* | Stop a transaction when the **EV driver is no longer authorized and/or the EV is disconnected.** |
| Old text | | *Scenario description* | **1.** The Charging Station opens the power relay (for any reason).<br>**2.** The Charging Station sends a TransactionEventRequest (eventType = Ended) notifying the CSMS about a transaction that has ended.<br>**3.** The CSMS responds with a TransactionEventResponse, confirming that the TransactionEventRequest was received. |
| New text | | *Scenario description* | **1.** The Charging Station **is disconnected from EV and/or the EV driver is no longer authorized.**<br>**2.** The Charging Station sends a TransactionEventRequest (eventType = Ended) notifying the CSMS about a transaction that has ended.<br>**3.** The CSMS responds with a TransactionEventResponse, confirming that the TransactionEventRequest was received. |

The text "open power relay" and self-referencing arrow in Figure 54 must be removed.

## 7.7.4. Page 143 - E06.FR.06

PowerPathClosed is treated as the logical OR of "EVConnected" and "Authorized" in the context of a TxStopPoint.

| **NOTE** | For a TxStopPoint the negation of a situation is used, because it is about the ending of a situation, e.g. "NOT Authorized".<br>NOT PowerPathClosed == NOT (EVConnected && Authorized) == NOT EVConnected OR NOT Authorized. |
|---|---|

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | E06.FR.06 | `TxStopPoint` contains:<br>PowerPathClosed<br>AND<br>Power relay is opened | The Charging Station SHALL stop the transaction and send a TransactionEventRequest (eventType = Ended) to the CSMS. |
| New text | E06.FR.06 | `TxStopPoint` contains:<br>PowerPathClosed<br>AND<br>**( EV Driver is no longer authorized OR Charging Station is no longer connected to EV )** | The Charging Station SHALL stop the transaction and send a TransactionEventRequest (eventType = Ended) to the CSMS. |

## 7.8. Page 123 - (v2) E01.FR.12 TransactionEventResponse must also return groupIdToken [642]

TransactionEventResponse returns an *idTokenInfo* field when the TransactionEventRequest contains an *idToken*. The *groupIdToken* field within *idTokenInfo* is optional, because a *idToken* does not need to have an associated *groupIdToken*. However, if it exists, then the *groupIdToken* must be returned as part of the *idTokenInfo* field.

*Changed requirement*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | E01.FR.12 | E01.FR.11 | The CSMS SHALL send a TransactionEventResponse that includes an authorization status value. |
| New text | E01.FR.12 | E01.FR.11 | The CSMS SHALL send a TransactionEventResponse that includes **in *idTokenInfo*** an authorization status value **and the *groupIdToken* if one exists for the *idToken*.** |

## 7.9. Page 123 - (v2) Requirements about chargingState and SuspendedEV [582]

A requirement about reporting a change in chargingState is present in E02.FR.13, but should have been part of use case E01. Two recommendations about dealing with temporary suspension of energy transfer by EV are added as new "SHOULD" requirements.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| E01.FR.18 | If the charging state changes | The Charging Station SHALL send a TransactionEventRequest including the *chargingState* element. |
| E01.FR.19 | When EV temporarily suspends the energy transfer | The Charging Station SHOULD send a TransactionEventRequest with *chargingState* = `SuspendedEV` |
| E01.FR.20 | E01.FR.19 AND<br>The Charging Station is not able to handle temporary suspension of energy transfer | The Charging Station SHOULD send a TransactionEventRequest with *chargingState* = `EVConnected`. |

## 7.10. Page 125 - (v2) Use case E02: Invalid remark about multiple authorizations [568]

Use case E02 contains an invalid remark about the possiblity of having multiple authorizations in a transaction. This is in conflict with Page 75 - (v1) Requirement to avoid authorizing multiple idTokens in a transaction [514].

|  |  |  |  |
|---|---|---|---|
| Old text | **8** | **Remark(s)** | If the Charging Station has implemented an Authorization Cache, then upon receipt of TransactionEventResponse, the Charging Station updates the cache entry.<br>It is now possible and allowed to send IdTokenType in more than 1 TransactionEventRequest. The CSMS has to be able to handle/process multiple IdTokenType per transaction. It is up to the CSO how they use this information (for billing purposes).<br>The scenario description and sequence diagram above are based on the Configuration Variable for start & stop transaction being configured as follows:<br>`TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer<br>This use-case is also valid for other configurations, but then the transaction might start at another moment, which might change the sequence in which message are sent. For more details see the use cases: E01 - Start Transaction options and E06 - Stop Transaction options. |
| New text | **8** | **Remark(s)** | If the Charging Station has implemented an Authorization Cache, then upon receipt of TransactionEventResponse, the Charging Station updates the cache entry.<br>~~It is now possible and allowed to send IdTokenType in more than 1 TransactionEventRequest. The CSMS has to be able to handle/process multiple IdTokenType per transaction. It is up to the CSO how they use this information (for billing purposes).~~<br>The scenario description and sequence diagram above are based on the Configuration Variable for start & stop transaction being configured as follows:<br>`TxStartPoint`: EVConnected, Authorized, DataSigned, PowerPathClosed, EnergyTransfer<br>This use-case is also valid for other configurations, but then the transaction might start at another moment, which might change the sequence in which message are sent. For more details see the use cases: E01 - Start Transaction options and E06 - Stop Transaction options. |

## 7.11. Page 125 - (v2) Sequence diagram error: idToken.id shown more than once [591]

The sequence diagram of Figure 47 shows the *idToken.id* twice in a TransactionEventRequest message.
First time immediately after authorization (*triggerReason*=Authorized); which is correct.
Second time at start of charging (*triggerReason*=ChargingStateChange); this is not correct. *idToken.id* should not be part of the message anymore.

## 7.12. Page 126 - (v1) Requirement E02.FR.05 new note about NotifyEvent [437]

In order to make clear that device model notifications can be sent instead of StatusNotifications a note has been added to the requirement below.

*Changed requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | E02.FR.05 | When a cable is plugged in | The Charging Station SHALL send a StatusNotificationRequest with status: *Occupied* |  |

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New text | E02.FR.05 | When a cable is plugged in | The Charging Station SHALL send a StatusNotificationRequest with status: *Occupied* | Alternatively, a NotifyEventRequest message for **component**( name = 'Connector', evse.id = <*x*>, evse.connectorId = <*y*> ), **variable**( name = 'AvailabilityState' ), and **actualValue** = 'Occupied' MAY be sent to signal that Connector <*y*> of EVSE <*x*> is now occupied. |

## 7.13. Page 126 - (v1) Precondition E02.FR.06 is incomplete [438]

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | E02.FR.06 | When a cable is plugged in | The Charging Station SHALL send a TransactionEventRequest. | |
| New text | E02.FR.06 | When a cable is plugged in **AND `TxStartPoint` contains EVConnected** | The Charging Station SHALL send a TransactionEventRequest. | |

## 7.14. Page 126 - (v1) Missing requirement about authorization during transaction [514][547]

An explicit requirement that the *idToken* must be supplied in the next TransactionEventRequest after successful authorization (like in E03.FR.01) is missing in E02.

*Added requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E02.FR.20 | When a transaction has not been authorized before AND the Charging Station authorizes an *idToken* to start charging | The next TransactionEventRequest from Charging Station SHALL contain the *idToken* and have *triggerReason* = `Authorized`. | If authorization is not successful, then no TransactionEventRequest is sent, because this event has no effect on the running transaction. (For authorization to stop charging, see E07). |

*) Requirement number in v1 was "E02.FR.19" by mistake.

## 7.15. Page 126 - (v2) Broken table caused a requirement to be not shown

This was the last requirement of Use case E02, however because of a formatting error it was not visible. Including the requirement as E02.FR.21, because another erratum already adds a requirement with number E02.FR.20. Including this requirement only provides additional guidance for Use case E02 as a duplicated requirement. It is not a new requirement, because this requirement already exists for other Use cases.

*Added requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| E02.FR.21 | When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration<br>AND<br>EVSE is not known at start of transaction | The Charging Station SHALL add the measurands for *eventType* = `Started` to the optional meterValue field with *context* = `Transaction.Begin` in the TransactionEventRequest(eventType = Updated) that occurs when charging starts. | |

# 7.16. Page 130 - (v1) Precondition E03.FR.04 is incomplete [435]

The requirement E03.FR.04 is not relevant, because the connector will not have been reported as **Occupied** because the connector has not yet been plugged in.

*Deleted requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Deleted | E03.FR.04 | When the EV Driver does not plug-in the Charging Cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut` | The Charging Station SHALL send a StatusNotificationRequest with **status** set to **Available**, to the CSMS. | |

Since requirement E03.FR.05 uses E03.FR.04 as its precondition, we need to update E03.FR.05 as follows:

*Changed requirement*

**WARNING** This requirement has been updated in Errata v2: Page 139 - (v2) Handling of EVConnectionTimeout better defined [627].

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | E03.FR.05 | E03.FR.04 | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. | |
| New text | E03.FR.05 | **When the EV Driver does not plug-in the charging cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut`** | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. | |

# 7.17. Page 139 - (v2) Handling of EVConnectionTimeout better defined [627]

Requirement E03.FR.05 states that the transaction shall be deauthorized when an EVConnectionTimeout occurs, i.e. when the user has authorized, but does not connect the charging cable in time. If, for example, the TxStartPoint is `Authorized`,`EVConnected` and the TxStopPoint is `EVConnected` (a common setting), then the transaction that was started upon authorization will not be ended by deauthorization.

If the user now drives away, then an open transaction remains forever on the charging station. If this transaction was tied to an EVSE, then that EVSE can no longer be used by anyone else. (Requirement C01.FR.23 does not allow any other id tokens to continue or stop this transaction.)

This situation can be avoided by ending the transaction upon an EVConnectionTimeout. This is not needed if the TxStopPoint contains `ParkingBayOccupancy`, because in that case a normal end of transaction will occur when the user leaves the parking bay.

## 7.17.1. Page 139 - Use case E03 Start Transaction - IdToken First

*Changed requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text Errata v1 | E03.FR.05 | When the EV Driver does not plug-in the charging cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut` | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. |  |
| New text | E03.FR.05 | When the EV Driver does not plug-in the charging cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut`<br>**AND**<br>**TxStopPoint does not contain `ParkingBayOccupancy`** | The Charging Station SHOULD **end** the transaction and send a TransactionEventRequest (**eventType = Ended, stoppedReason = Timeout,** *triggerReason* = `EVConnectionTimeout`) to the CSMS. | This requirement is an additional safety measure to make sure the transaction is ended when the `EVConnectionTimeOut` is triggered. However it is up to the CSMS to make sure that sensible TxStartPoint / TxStopPoint combinations are configured. E.g. if Authorized is used as TxStartPoint, it should also be used as TxStopPoint. |

A new requirement is add to handle the case where TxStopPoint contains `ParkingBayOccupancy`.

*New requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New | E03.FR.15 | When the EV Driver does not plug-in the charging cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut`<br>**AND**<br>**TxStopPoint contains `ParkingBayOccupancy`** | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. | Transaction will be ended normally when driver leaves the parking bay. |

## 7.17.2. Page 172 - Use case F02 Remote Start Transaction - Remote Start First

The above also applies to a remote start transaction when the RequestStartTransactionRequest is received before the user has plugged in.

*Changed requirements*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | F02.FR.07 | When the EV Driver does not plug-in the Charging Cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut`<br>AND<br>*status* of the connector is `Occupied` | The Charging Station SHALL send a StatusNotificationRequest with *status* set to `Available`, to the CSMS. |  |
| New text | F02.FR.07 | When the EV Driver does not plug-in the charging cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut`<br>**AND**<br>**TxStopPoint does not contain `ParkingBayOccupancy`** | The Charging Station SHALL **end the transaction and send a TransactionEventRequest (eventType = Ended, stoppedReason = Timeout,** *triggerReason* = `EVConnectionTimeout`) to the CSMS. | Otherwise the transaction would not be ended in case the TxStopPoint does not contain `Authorized`. |
| Old text | F02.FR.08 | F02.FR.07 | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. |  |

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New | F02.FR.08 | When the EV Driver does not plug-in the charging cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut`<br>**AND**<br>**TxStopPoint contains `ParkingBayOccupancy`** | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. | Transaction will be ended normally when driver leaves the parking bay. |

## 7.18. Page 136 - (v2) E05.FR.02 should not contain PowerPathClosed [604]

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | E05.FR.02 | E05.FR.01 AND The authorization status in TransactionEventResponse is not *Accepted*<br>AND<br>The transaction is still ongoing AND<br>`StopTxOnInvalidId` is set to *false* AND `MaxEnergyOnInvalidId` is not implemented<br>or has been exceeded. `TxStopPoint` does NOT contain: (PowerPathClosed OR EnergyTransfer) | The Charging Station SHALL stop the energy delivery to the EV immediately and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *ChargingStateChanged* and chargingState set to *SuspendedEVSE* | |
| New text | E05.FR.02 | E05.FR.01 AND The authorization status in TransactionEventResponse is not *Accepted*<br>AND<br>The transaction is still ongoing AND<br>`StopTxOnInvalidId` is set to *false* AND `MaxEnergyOnInvalidId` is not implemented<br>or has been exceeded. `TxStopPoint` does NOT contain: **EnergyTransfer** | The Charging Station SHALL stop the energy delivery to the EV immediately and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *ChargingStateChanged* and chargingState set to ***SuspendedEVSE*** | The transaction is not deauthorized, but transfer of energy stops, since `MaxEnergyOnInvalidId` has been exceeded or is not set. If TxStopPoint contains `EnergyTransfer` then this would have ended the transaction. |

## 7.19. Page 137 - (v2) ChargingState in requirement E05.FR.09 [596][647]

Requirement E05.FR.09 states that the *chargingState* should be `SuspendedEVSE` when the transaction is deauthorized, when the id token is not valid. However, the *chargingState* `SuspendedEVSE` is not correct for a deauthorized transaction. Since charging cannot be resumed, as there is no authorization anymore, *chargingState* should return to `EVConnected`.

Both *chargingStates*, `SuspendedEVSE` and `EVConnected`, have the same effect of not delivering any energy. Therefore, in order to avoid breaking existing implementations that adhere to the original requirement, the use of `SuspendedEVSE` is still allowed in this situation. Use of charging state `SuspendedEVSE` that is not followed by `EVConnected` in this situation, will become deprecated in the next OCPP release.

If the physical change of charging state in the Charging Station occurs a few seconds or milliseconds later than the trigger *Deauthorized*, then the *chargingState* change may be reported separately as a *triggerReason* = *ChargingStateChanged*.

(See also C15.FR.03 in Page 105 - (v1) ChargingState in requirement C15.FR.03 [464])

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | E05.FR.09 | E05.FR.01 AND<br>The authorization status in TransactionEventResponse is not *Accepted* AND<br>The transaction is still ongoing AND<br>`StopTxOnInvalidId` is *true* AND<br>`TxStopPoint` does NOT contain: (Authorized OR PowerPathClosed OR EnergyTransfer) | The Charging Station SHALL stop the energy transfer and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *Deauthorized* and chargingState set to *SuspendedEVSE*. | |
| New text | E05.FR.09 | E05.FR.01 AND<br>The authorization status in TransactionEventResponse is not *Accepted* AND<br>The transaction is still ongoing AND<br>`StopTxOnInvalidId` is *true* AND<br>`TxStopPoint` does NOT contain: (Authorized OR PowerPathClosed OR EnergyTransfer) | The Charging Station SHALL stop the energy transfer and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *Deauthorized* **and in the same or next TransactionEventRequest report *chargingState* set preferably to *EVConnected*, or alternatively to *SuspendedEVSE*.** | **If the physical change of charging state in the Charging Station occurs a few seconds or milliseconds later than the trigger *Deauthorized*, then the *chargingState* change may be reported separately as a *triggerReason* = *ChargingStateChanged*.**<br>Use of charging state *SuspendedEVSE* that is not followed by *EVConnected* in this situation will become deprecated in the next OCPP release. |

## 7.20. Page 140 - (v2) Use case E06: scenario S4 is not valid [537]

Scenario S4 to stop a transaction as a result of TxStopPoint = `DataSigned` is not valid anymore, because the TxStopPoint `DataSigned` was removed in Errata v1. (See Page 434 - (v1) Better description of TxStartPoint/TxStopPoint [348]).

| S4 | *Scenario objective* | ~~**Stop a transaction when the meter stops providing signed meter values.**~~ |
|---|---|---|

## 7.21. Page 142 - (v2) Use case E06 is missing a common stopping scenario [540]

Use case E06 is missing the most common stop scenario: stopping by presenting IdToken. The use case has a scenario S3, but that refers to the situation where an IdToken is **deauthorized** by CSMS, because it is invalid, which is a different situation. We need a new scenario that describes local stopping by IdToken. This is described as a special use case in E07, but should at least be mentioned in E06 as a stop option.

| S7 | *Scenario objective* | **Stop a transaction when EV driver ends authorization** |
|---|---|---|
| | *Scenario description* | **1.** The EV drivers presents an IdToken to end the charging.<br>**2.** The Charging Station sends a TransactionEventRequest (eventType = Ended) notifying the CSMS about a transaction that has ended.<br>**3.** The CSMS responds with a TransactionEventResponse, confirming that the TransactionEventRequest was received. |
| | **Prerequisite(s)** | A transaction is ongoing.<br>Configuration Variable: `TxStopPoint` contains: Authorized (or PowerPathClosed). |

| S7 | *Scenario objective* | **Stop a transaction when EV driver ends authorization** |
|---|---|---|
| | **Postcondition(s)** | **Successful postcondition:**<br>The transaction is ended and the CSMS is *Successfully* informed.<br><br>**Failure postcondition:**<br>The transaction is still ongoing. *or*<br>The CSMS is *not* informed. |

## 7.22. Page 143 - (v1) Confusing precondition in E06.FR.06 [506]

E06.FR.06 about TxStopPoint containing `PowerPathClosed` refers to the power relay being opened as a reason to stop the transaction. This is confusing, because there may be other situations were a power relay is opened, but the transaction is not ended. This can occur, for example, in case of a charging state `SuspendedEVSE`.

`PowerPathClosed` is in fact the combination of `EVConnected` and `Authorized`. As soon as one of them is no longer applicable, then `PowerPathClosed` is no longer applicable. This leads to the following rephrasing of the precondition.

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | E06.FR.06 | `TxStopPoint` contains: PowerPathClosed<br>AND<br>Power relay is opened | The Charging Station SHALL stop the transaction and send a TransactionEventRequest (eventType = Ended) to the CSMS. |
| New text | E06.FR.06 | `TxStopPoint` contains: PowerPathClosed<br>AND (<br>**Connection between Charging Station and EV is lost**<br>**OR**<br>**Authorization has ended or idToken is deauthorized )** | The Charging Station SHALL stop the transaction and send a TransactionEventRequest (eventType = Ended) to the CSMS. |

## 7.23. Page 143 - (v1) Requirements for eventType=Started do not belong in use case E06/E07 [453]

### 7.23.1. Page 143 - (v1) Requirement E06.FR.11 and E06.FR.17

Requirement E06.FR.11 erroneously refers to an *eventType* = `Started`, but that is not correct in this use case about stopping a transaction.

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | E06.FR.11 | When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration<br>AND<br>EVSE is known at start of transaction | The Charging Station SHALL add the configured measurands to the optional meterValue field with *context* = `Transaction.Begin` in the TransactionEventRequest(eventType = Started) sent to the CSMS to provide more details during the transaction. |
| New text | E06.FR.11 | **When configured to send meter data in the TransactionEventRequest (eventType = Ended), See: Meter Values - Configuration** | The Charging Station SHALL add the configured measurands to the optional *meterValue* field with ***context* = `Transaction.End`** in the TransactionEventRequest(eventType = Ended) sent to the CSMS to provide more details about transaction usage. |

The following requirement does not belong in this use case and must be removed:

*Deleted requirement*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Deleted | E06.FR.17 | When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration<br>AND<br>EVSE is not known at start of transaction | The Charging Station SHALL add the measurands for *eventType* = `Started` to the optional meterValue field with *context* = `Transaction.Begin` in the TransactionEventRequest(eventType = Updated) that occurs when charging starts. |

## 7.23.2. Page 145 - (v1) Requirement E07.FR.08 and E07.FR.13

Requirement E07.FR.08 erroneously refers to an *eventType* = `Started`, but that is not correct in this use case about stopping a transaction.

*Changed requirement*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | E07.FR.08 | When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration<br>AND<br>EVSE is known at start of transaction | The Charging Station SHALL add the configured measurands to the optional meterValue field with *context* = `Transaction.Begin` in the TransactionEventRequest (eventType = Started) sent to the CSMS to provide more details during the transaction. |
| New text | E07.FR.08 | **When configured to send meter data in the TransactionEventRequest (eventType = Ended), See: Meter Values - Configuration** | The Charging Station SHALL add the configured measurands to the optional *meterValue* field with ***context* = `Transaction.End`** in the TransactionEventRequest(eventType = Ended) sent to the CSMS to provide more details about transaction usage. |

The following requirement does not belong in this use case and must be removed:

*Deleted requirement*

|  | ID | Precondition | Requirement definition |
|---|---|---|---|
| Deleted | E07.FR.13 | When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration<br>AND<br>EVSE is not known at start of transaction | The Charging Station SHALL add the measurands for *eventType* = `Started` to the optional meterValue field with *context* = `Transaction.Begin` in the TransactionEventRequest(eventType = Updated) that occurs when charging starts. |

# 7.24. Page 143 - (v2) Use case E07 should use TxStopPoint = Authorized [541]

Use case E07 describes how a transaction can be stopped locally by presenting the IdToken again, but the use case assumes a TxStopPoint of `EVConnected`. As a result the transaction would not be stopped by presenting the IdToken. It would be stopped when unplugging the cable. This is confusing.

The scenario description of E07 is therefore changed as follows to show behavior for TxStopPoint = `Authorized` (or PowerPathClosed as this has the same sequence).

This matches the sequence diagrams from Page 144 - (v1) Use case E07, sequence diagram is not complete, which shows the behavior for each TxStopPoint as an option in an alt-box.

|  | No. | Type | Description |
|---|---|---|---|
| Old text |  | *Scenario description* | **1.** The EV Driver is authorized by the Charging Station and/or CSMS.<br>**2.** If the cable is not permanently attached, the Charging Station unlocks the cable.<br>**3.** The Charging Station sends a TransactionEventRequest (eventType = Updated) with trigger *StopAuthorized*<br>**4.** The CSMS responds with a TransactionEventResponse.<br>**5.** The EV Driver unplugs the cable (and drives away the EV).<br>**6.** The Charging Station sends a StatusNotificationRequest with status *Available*, notify the CSMS that the Connector is available again.<br>**7.** The CSMS responds with a StatusNotificationResponse.<br>**8.** The Charging Station sends a TransactionEventRequest (eventType = Ended)<br>**9.** The CSMS responds with a TransactionEventResponse. |
| New text |  | *Scenario description*<br>TxStopPoint = `Authorized` (or `PowerPathClosed`) | **1.** The EV Driver **presents IdToken a second time to end charging.**<br>**2.** The Charging Station sends a TransactionEventRequest (eventType = Ended) with *triggerReason* = `StopAuthorized` and *stoppedReason* = `Local`.<br>**3.** The CSMS responds with a TransactionEventResponse.<br>**4.** The Charging Station stops the energy transfer and if the cable is not permanently attached, the Charging Station unlocks the cable.<br>**5.** ~~The EV Driver unplugs the cable (and drives away the EV).~~<br>**6.** ~~The Charging Station sends a StatusNotificationRequest with status *Available*, notify the CSMS that the Connector is available again.~~<br>**7.** ~~The CSMS responds with a StatusNotificationResponse.~~<br>**8.** ~~The Charging Station sends a TransactionEventRequest (eventType = Ended)~~<br>**9.** ~~The CSMS responds with a TransactionEventResponse.~~ |
| Old text |  | *Alternative scenario(s)* | **1.** The Charging Station MAY unlock the cable (if not permanently attached) when the cable is disconnected at the EV. If supported, this functionality is reported and controlled by the Configuration Variable `UnlockOnEvSideDisconnect`.<br>**2.** The Charging Station MAY stop an ongoing transaction when the cable is disconnected at the EV. If supported, this functionality is reported and controlled by the Configuration Variable `StopTxOnEVSideDisconnect`.<br>E07 - Offline Stop Transaction<br>E08 - When cable disconnected on EV-side: Stop Transaction<br>E09 - When cable disconnected on EV-side: Suspend Transaction |
| New text |  | *Alternative scenario(s)*<br>TxStopPoint = `Authorized` (or `PowerPathClosed`) | **1.** The EV Driver presents IdToken a second time to end charging.<br>**2.** The Charging Station sends a TransactionEventRequest (eventType = Updated) with *triggerReason* = `StopAuthorized`<br>**3.** The CSMS responds with a TransactionEventResponse.<br>**4.** The Charging Station stops the energy transfer and if the cable is not permanently attached, the Charging Station unlocks the cable.<br>**5.** The Charging Station sends a TransactionEventRequest (eventType = Ended) with *triggerReason* = `ChargingStateChanged`, *transactionInfo.chargingState* = `EVConnected`<br>**6.** The CSMS responds with a TransactionEventResponse. |

For the associated sequence diagrams, see Page 144 - (v1) Use case E07, sequence diagram is not complete.

| | | | |
|---|---|---|---|
| Old text | **8** | **Remark(s)** | It is likely that the CSMS applies sanity checks to the data contained in TransactionEventRequest it received. The outcome of such sanity checks SHOULD NOT ever cause the CSMS to not respond with a TransactionEventResponse.<br>The scenario description and sequence diagram above are based on the Configuration Variable for stop transaction being configured as follows.<br>`TxStopPoint`: EVConnected<br>This use-case is also valid for other configurations, but then the transaction might stop at another moment, which might change the sequence in which message are sent. For more details see the use case: E06 - Stop Transaction options<br>The CSMS cannot prevent a transaction from stopping. |
| New text | **8** | **Remark(s)** | ~~It is likely that the CSMS applies sanity checks to the data contained in TransactionEventRequest it received. The outcome of such sanity checks SHOULD NOT ever cause the CSMS to not respond with a TransactionEventResponse.~~<br>~~The scenario description and sequence diagram above are based on the Configuration Variable for stop transaction being configured as follows.~~<br>~~`TxStopPoint`: EVConnected~~<br>~~This use-case is also valid for other configurations, but then the transaction might stop at another moment, which might change the sequence in which message are sent. For more details see the use case: E06 - Stop Transaction options~~<br>**The scenario descriptions are based on TxStopPoint containing Authorized or PowerPathClosed. The sequence diagrams also show behavior for other TxStopPoint values in the alt-blocks.**<br>The CSMS cannot prevent a transaction from stopping. |

## 7.25. Page 144 - (v1) Use case E07, sequence diagram is not complete

The sequence diagram for use case E07 shows the behavior for a Charging Station that has TxStopPoint configured as TxStopPoint = `EVConnected`, but this is not mentioned explicitly and may lead to confusion.

The following sequence diagrams show the alternative sequences when TxStopPoint is configured with different values, in the same diagram.

Figure 1 shows the sequence when the TransactionEventRequests are strictly reported by TxStopPoint configuration.
Figure 2 shows the sequence with a delayed TransactionEventRequest with eventType = Ended for TxStopPoint = Authorized OR TxStopPoint = PowerPathClosed.

| | |
|---|---|
| **NOTE** | The advantage of the alternative sequence from figure 2 is that the chargingState transition from Charging to EVConnected is reported when TxStopPoint is configured as Authorized or PowerPathClosed also. In some cases, this might be valuable information to be reported, thus making these TxStopPoint configurations more widely applicable. |

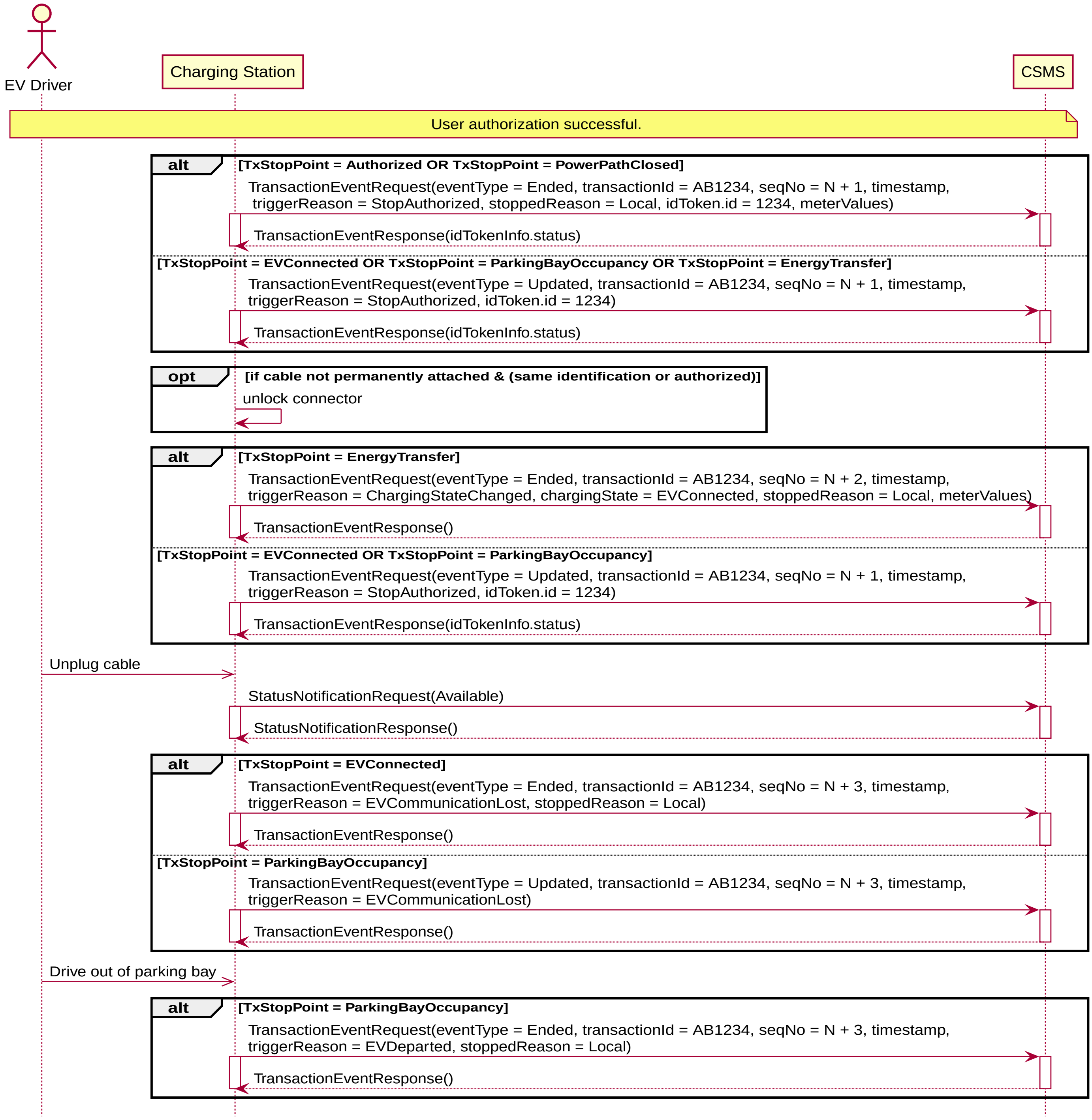

EV Driver
Charging Station
CSMS
User authorization successful.
alt
[TxStopPoint = Authorized OR TxStopPoint = PowerPathClosed]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 1, timestamp, triggerReason = StopAuthorized, stoppedReason = Local, idToken.id = 1234, meterValues)
TransactionEventResponse(idTokenInfo.status)
[TxStopPoint = EVConnected OR TxStopPoint = ParkingBayOccupancy OR TxStopPoint = EnergyTransfer]
TransactionEventRequest(eventType = Updated, transactionId = AB1234, seqNo = N + 1, timestamp, triggerReason = StopAuthorized, idToken.id = 1234)
TransactionEventResponse(idTokenInfo.status)
opt
[if cable not permanently attached & (same identification or authorized)]
unlock connector
alt
[TxStopPoint = EnergyTransfer]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 2, timestamp, triggerReason = ChargingStateChanged, chargingState = EVConnected, stoppedReason = Local, meterValues)
TransactionEventResponse()
[TxStopPoint = EVConnected OR TxStopPoint = ParkingBayOccupancy]
TransactionEventRequest(eventType = Updated, transactionId = AB1234, seqNo = N + 1, timestamp, triggerReason = StopAuthorized, idToken.id = 1234)
TransactionEventResponse(idTokenInfo.status)
Unplug cable
StatusNotificationRequest(Available)
StatusNotificationResponse()
alt
[TxStopPoint = EVConnected]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 3, timestamp, triggerReason = EVCommunicationLost, stoppedReason = Local)
TransactionEventResponse()
[TxStopPoint = ParkingBayOccupancy]
TransactionEventRequest(eventType = Updated, transactionId = AB1234, seqNo = N + 3, timestamp, triggerReason = EVCommunicationLost)
TransactionEventResponse()
Drive out of parking bay
alt
[TxStopPoint = ParkingBayOccupancy]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 3, timestamp, triggerReason = EVDeparted, stoppedReason = Local)
TransactionEventResponse()

*Figure 1. Sequence diagram with TransactionEventRequest reported strictly by TxStopPoint configuration*

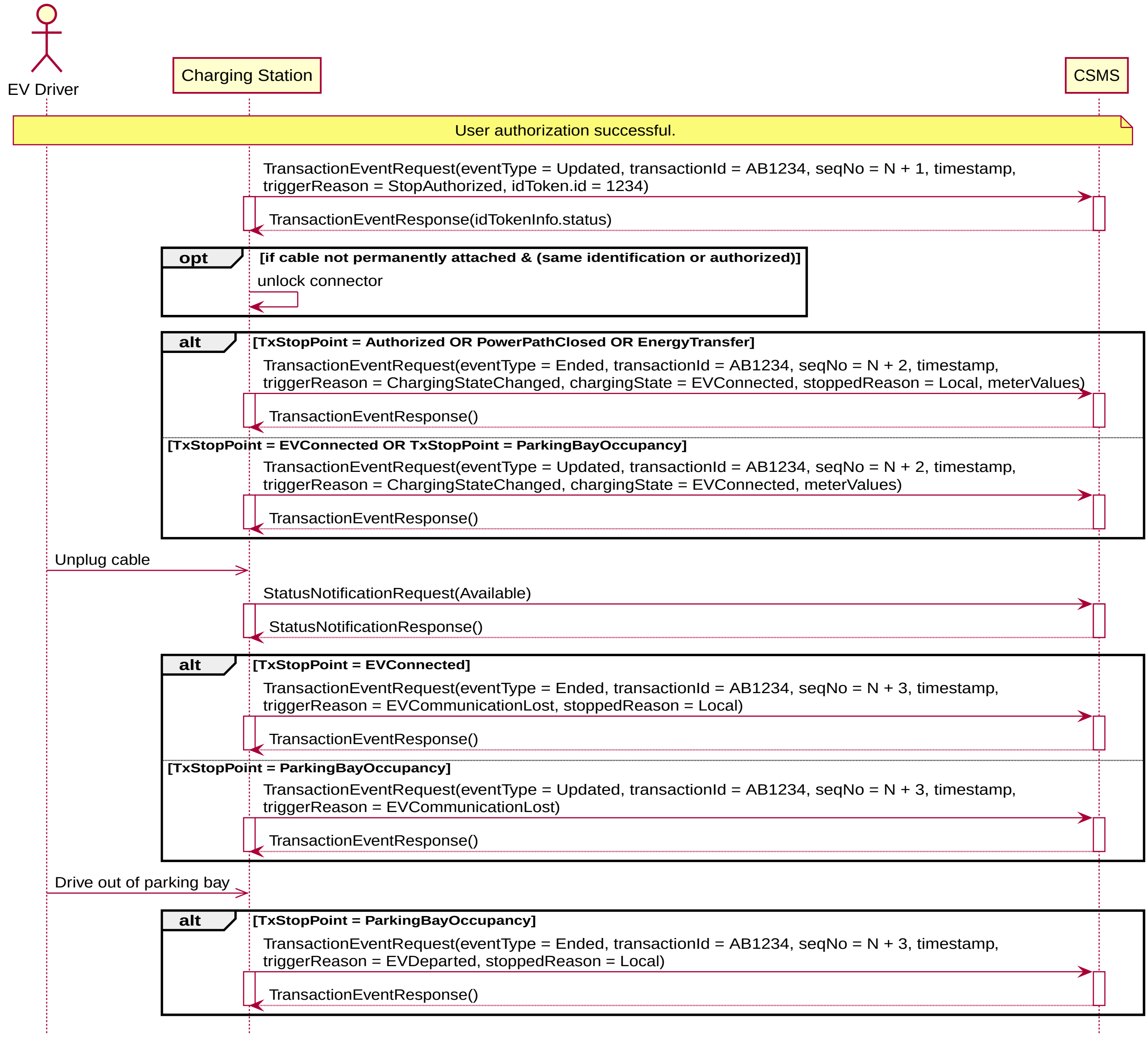

EV Driver
Charging Station
CSMS
User authorization successful.
TransactionEventRequest(eventType = Updated, transactionId = AB1234, seqNo = N + 1, timestamp, triggerReason = StopAuthorized, idToken.id = 1234)
TransactionEventResponse(idTokenInfo.status)
opt
[if cable not permanently attached & (same identification or authorized)]
unlock connector
alt
[TxStopPoint = Authorized OR PowerPathClosed OR EnergyTransfer]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 2, timestamp, triggerReason = ChargingStateChanged, chargingState = EVConnected, stoppedReason = Local, meterValues)
TransactionEventResponse()
[TxStopPoint = EVConnected OR TxStopPoint = ParkingBayOccupancy]
TransactionEventRequest(eventType = Updated, transactionId = AB1234, seqNo = N + 2, timestamp, triggerReason = ChargingStateChanged, chargingState = EVConnected, meterValues)
TransactionEventResponse()
Unplug cable
StatusNotificationRequest(Available)
StatusNotificationResponse()
alt
[TxStopPoint = EVConnected]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 3, timestamp, triggerReason = EVCommunicationLost, stoppedReason = Local)
TransactionEventResponse()
[TxStopPoint = ParkingBayOccupancy]
TransactionEventRequest(eventType = Updated, transactionId = AB1234, seqNo = N + 3, timestamp, triggerReason = EVCommunicationLost)
TransactionEventResponse()
Drive out of parking bay
alt
[TxStopPoint = ParkingBayOccupancy]
TransactionEventRequest(eventType = Ended, transactionId = AB1234, seqNo = N + 3, timestamp, triggerReason = EVDeparted, stoppedReason = Local)
TransactionEventResponse()

*Figure 2. Sequence diagram with delayed TransactionEventRequest eventType = Ended for TxStopPoint = Authorized OR PowerPathClosed*

## 7.26. Page 145 - (v2) Report *idToken* for stopping authorization [583]

Some requirements in E07 speak of "stopping a transaction" as result of presenting an IdToken. This is only correct when use case E07 uses a TxStopPoint containing `Authorized` or `PowerPathClosed`. We have rephrased "stopping a transaction" to "end authorization", which is correct in all cases. Depending on the TxStopPoint the ending of authorization may also cause the end of a transaction.

Requirement E07.FR.01 was missing a preconditioning, thus rendering it meaningless. It has been replaced with the precondition of C01.FR.03 about ending authorization by presenting the IdToken again.

Requirement E07.FR.02 implicitly assumes that the *idToken* that is presented by the user to end authorization, is reported in the next TransactionEventRequest message. This has been made explicit. (It is phrased as a "SHOULD" and not a "SHALL" to avoid breaking existing implementations.)

In Page 366 - (v2) Description of IdToken in TransactionEventRequest [583] the description of *idToken* has been improved to include this situation explicitly.

*Changed requirements*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | E07.FR.01 |  | The CSMS SHALL only inform the Charging Station it has received TransactionEventRequest. |  |
| New text | E07.FR.01 | **When an idToken is presented during a transaction that has been authorized AND**<br>**(a) the presented idToken is the same as the idToken that started the authorization**<br>**OR**<br>**(b) when the presented idToken is in the Local Authorization List or Authorization Cache AND is valid AND has the same GroupIdToken as the IdToken that started the authorization.** | **The Charging Station SHALL end the authorization of the transaction, without first sending an AuthorizeRequest** | **The idToken that started the authorization can always be used to end the authorization. Ending authorization will end delivery of energy. Depending on the TxStopPoint ending of the authorization may also end the transaction. (See C01.FR.03)** |
| Old text | E07.FR.02 | E07.FR.01 and when stopping a transaction. | The CSMS MAY send information about the IdTokenType used to stop the transaction. |  |
| New text | E07.FR.02 | E07.FR.01 | **The Charging Station SHALL send a TransactionEventRequest with *triggerReason* = `StopAuthorized` and SHOULD include the *idToken* used to stop authorization.** | The stopping *idToken* may differ from the starting *idToken*, when they share the same GroupId. |

The following requirements are incorrect in this use case and have been deleted.

*Deleted requirement*

|  | ID | Precondition | Requirement definition | Reason for deletion |
|---|---|---|---|---|
| Deleted | E07.FR.03 |  | The IdTokenType in the request message MAY be omitted when the Charging Station itself needs to stop the transaction. | **This use case is not about resetting or other ways of stopping.** |

## 7.27. Page 162 - (v1) Misspelled field name in requirements and remark(s) [452]

The field name *ongoing* must be *ongoingIndicator*.

*Changed requirements*

|  | ID | Precondition | Requirements |
|---|---|---|---|
| Old text | E14.FR.01 | The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND<br>It did not do a transaction with that *transactionId* | The Charging Station SHALL respond with *ongoing* = false AND *messagesInQueue* = false. |
| New text | E14.FR.01 | The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND<br>It did not do a transaction with that *transactionId* | The Charging Station SHALL respond with ***ongoingIndicator*** = false AND *messagesInQueue* = false. |
| Old text | E14.FR.02 | The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND<br>The transaction with that *transactionId* has not stopped yet | The Charging Station’s response SHALL have *ongoing* = true. |

|  | ID | Precondition | Requirements |
|---|---|---|---|
| New text | E14.FR.02 | The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND<br>The transaction with that *transactionId* has not stopped yet | The Charging Station's response SHALL have ***ongoingIndicator*** = true. |
| Old text | E14.FR.03 | The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND<br>The transaction with that *transactionId* has stopped | The Charging Station's response SHALL have *ongoing* = false. |
| New text | E14.FR.03 | The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND<br>The transaction with that *transactionId* has stopped | The Charging Station's response SHALL have ***ongoingIndicator*** = false. |
| Old text | E14.FR.06 | The Charging Station receives a GetTransactionStatusRequest without a *transactionId* | The Charging Station's response SHALL NOT have *ongoing* set. |
| New text | E14.FR.06 | The Charging Station receives a GetTransactionStatusRequest without a *transactionId* | The Charging Station's response SHALL NOT have ***ongoingIndicator*** set. |

*Changed remark(s)*

|  |  |  |  |
|---|---|---|---|
| Old text | **8** | **Remark(s)** | When the CSMS receives a GetTransactionStatusResponse with both fields (*ongoing* and *messagesInQueue*) set to false, this might mean that the transaction is finished and there are no more messages in the queue for this transaction, or the Charging Station doesn't know anything about this transaction (anymore). |
| New text | **8** | **Remark(s)** | When the CSMS receives a GetTransactionStatusResponse with both fields (***ongoingIndicator*** and *messagesInQueue*) set to false, this might mean that the transaction is finished and there are no more messages in the queue for this transaction, or the Charging Station doesn't know anything about this transaction (anymore). |

# 8. Use case F Remote Control

**NOTE** The use cases F01 and F02 in F Remote Control are based on a TxStartPoint containing `EVConnected` and `Authorized`, meaning that either connecting the cable or authorizing the idToken will start a transaction. (The remaining start points mentioned are not relevant for the use case).
Use case F03 has a TxStopPoint containing `EVConnected`.
Beware that requirements for these use cases regarding the starting and stopping of a transaction behave differently for different TxStartPoints and TxStopPoints.

## 8.1. Page 169 - (v2) Use case F01: requirements unclear about when transaction starts [580]

Use case F01 has a TxStartPoint containing `EVConnected`. This means that the transaction has already been started at the moment of cable plug-in and not when the remote start request is received. As a result the authorization of the idToken will not start at transaction, but allow the transfer of energy in the already started transaction.

**NOTE** The old text would have been valid for F01 if TxStartPoint had not contained `EVConnected`.

*Changed requirements*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | F01.FR.01 | If the value of `AuthorizeRemoteStart` = true. | The Charging Station SHALL behave as if in response to a local action at the Charging Station to start a transaction with the IdToken given in RequestStartTransactionRequest message. | This means that the Charging Station will first try to authorize the IdToken, using the Local Authorization List, Authorization Cache and/or an AuthorizeRequest. A transaction will only be started after authorization was obtained. |
| New text | F01.FR.01 | If the value of `AuthorizeRemoteStart` = true<br>**AND**<br>**Charging Station receives a RequestStartTransactionRequest** | The Charging Station SHALL behave as if in response to a local action at the Charging Station **to allow energy transfer after successful authorization of** the IdToken given in RequestStartTransactionRequest message. | **Charging Station will first respond to the request and then** try to authorize the IdToken, using the Local Authorization List, Authorization Cache and/or an AuthorizeRequest. **Energy transfer is only allowed** after authorization was obtained. |
| Old text | F01.FR.02 | If the value of `AuthorizeRemoteStart` = false. | The Charging Station SHALL immediately try to start a transaction for the IdToken given in RequestStartTransactionRequest message. | Note that after the transaction has been started, the Charging Station will send a TransactionEventRequest with the idToken to the CSMS, and the CSMS will check the authorization status of the IdToken when processing this TransactionEventRequest. |
| New text | F01.FR.02 | If the value of `AuthorizeRemoteStart` = false<br>**AND**<br>**Charging Station receives a RequestStartTransactionRequest** | The Charging Station SHALL **allow energy transfer** for the IdToken given in RequestStartTransactionRequest message **without checking authorization**. | **Charging Station will first respond to the request, and** send a TransactionEventRequest with the idToken to the CSMS, and the CSMS will check the authorization status of the IdToken when processing this TransactionEventRequest. |
| Old text | F01.FR.03 | After the transaction has been started. | The Charging Station SHALL send a TransactionEventRequest to the CSMS, and the CSMS will check the authorization status of the IdToken when processing this TransactionEventRequest. |  |
| New text | F01.FR.03 | **F01.FR.01 OR F01.FR.02** | The Charging Station SHALL send a TransactionEventRequest to the CSMS, and the CSMS will check the authorization status of the IdToken when processing this TransactionEventRequest. | **If CSMS returns an authorization status that is not `Accepted`, then Charging Station must stop energy transfer as per use case E05.** |

## 8.2. Page 169 - (v1) Requirements for rejecting request are missing [449, 509]

The use case F01 (and F02) assume that the EVSE is available for use by the RequestStartTransaction request, but that is not made explicit. The specified EVSE may be reserved, unavailable or already charging.

New requirements are added for that.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F01.FR.20 | If the RequestStartTransactionRequest does not contain an *evseId* AND the Charging Station is capable of selecting an EVSE | The Charging Station SHALL select an EVSE to be used as a value for *evseId* for the operation | See also F01.FR.07 if Charging Station does not support starting at an arbitrary EVSE. |
| F01.FR.21 | When the *evseId* for RequestStartTransactionRequest is `Reserved` for an *idToken* that differs from *idToken* in the request AND has no reservation for a *groupIdToken* | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | |
| F01.FR.22 | When the *evseId* for RequestStartTransactionRequest is `Reserved` for an *idToken* that differs from *idToken* in the request AND is `Reserved` for a *groupIdToken* that differs from *groupIdToken* in the request | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | EV is not allowed to use station if neither *idToken* nor *idGroupToken* match the reservation. |
| F01.FR.23 | When the *evse* for RequestStartTransactionRequest is `Unavailable` or `Faulted` | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | |
| F01.FR.24 | When the *evseId* for RequestStartTransactionRequest is `Occupied` AND this *evseId* has a transaction that has been authorized | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | Only an EVSE with no transaction or with a transaction that has not yet been authorized can be matched with the RequestStartTransactionRequest |

## 8.3. Page 170 - (v1) Requirement missing for *remoteStartId* [517]

A requirement similar to F02.FR.01, that a *remoteStartId* must be returned in a TransactionEvent, is missing in use case F01, but it is required according to requirement C05.FR.03.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F01.FR.25 | F01.FR.13 | The Charging Station SHALL put the *remoteStartId* in the next TransactionEventRequest it sends for the associated transaction. | |

## 8.4. Page 170 - (v2) Use case F01/F02: ChargingProfile in RequestStartTransaction [558]

Use cases F01 and F02 state that a charging station may ignore a charging profile from a RequestStartTransactionRequest if it does not support smart charging. However, it must be possible for a smart charging-capable charging station to reject an **invalid** charging profile.

### 8.4.1. Page 170: Use case F01

*Added note*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F01.FR.12 | If a Charging Station without support for Smart Charging receives a RequestStartTransactionRequest with a ChargingProfile. | The Charging Station SHALL ignore the specified ChargingProfile. | **The device model variable SmartChargingCtrlr.Enabled tells CSMS whether smart charging is supported.** |

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F01.FR.26 | If a Charging Station **with** support for Smart Charging receives a RequestStartTransactionRequest with an invalid ChargingProfile. | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected` and optionally with *reasonCode* = "InvalidProfile" or "InvalidSchedule". | The device model variable SmartChargingCtrlr.Enabled tells CSMS whether smart charging is supported. |

### 8.4.2. Page 172: Use case F02

**NOTE** Many F01 requirements have been copied to F02 in Page 172 - (v2) Relevant F01 requirements copied to F02 Remote Start First [590].

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F02.FR.27 | If a Charging Station **with** support for Smart Charging receives a RequestStartTransactionRequest with an invalid ChargingProfile. | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected` and optionally with *reasonCode* = "InvalidProfile" or "InvalidSchedule". | The device model variable SmartChargingCtrlr.Enabled tells CSMS whether smart charging is supported. |

### 8.4.3. Page 358 - (v1) Minor change to description of *remoteStartId*

Returning the *remoteStartId* of a RequestStartTransactionRequest is not optional. The description of the field *remoteStartId* needs to be updated accordingly.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old | **remoteStartId** | integer | 1..1 | Required. Id given by the server to this start request. The Charging Station might return this in the TransactionEventRequest, letting the server know which transaction was started for this request. Use to start a transaction. |
| New | **remoteStartId** | integer | 1..1 | Required. Id given by the server to this start request. The Charging Station **will** return this in the TransactionEventRequest, letting the server know which transaction was started for this request. |

## 8.5. Page 171 - (v1) Sequence diagram error [435]

The sequence diagram on Figure 66 shows that a StatusNotification(Occupied) is sent after the RequestStartTransaction, but before the cable has been plugged in. There is no requirement that the connector should be reported occupied already at this moment. Step 5 and 6 have therefore been deleted from the use case description and the StatusNotification now occurs in steps 8a and 8b after cable plug-in.

## 8.5.1. Use case F02 scenario description

Changes to the scenario description shown in bold (except for step numbering).

| No. | Type | Description |
|---|---|---|
| | *Scenario description* | **1.** An External Trigger triggers the remote start.<br>**2.** The CSMS sends RequestStartTransactionRequest to the Charging Station.<br>**3.** The Charging Station responds with RequestStartTransactionResponse to the CSMS.<br>**4.** The EV Driver is authorized by the CSMS, dependent on the Configuration Variable settings.<br>**5.** ~~The Charging Station sends StatusNotificationRequest to the CSMS to inform it about a Connector became *Occupied*.~~<br>**6.** ~~The CSMS sends StatusNotificationResponse to the Charging Station~~<br>**7.** The Charging Station sends a TransactionEventRequest (eventType = Started) notifying the CSMS about a transaction that has started<br>**8.** The cable is plugged in.<br>**8a. Charging Station sends a StatusNotificationRequest with *Occupied*.**<br>**8b. CSMS sends a StatusNotificationResponse to the Charging Station**<br>**9.** The energy offer is started.<br>**10.** The Charging Station sends a TransactionEventRequest (eventType = Updated, chargingState = Charging) message to inform the CSMS that the charging has started.<br>**11.** The CSMS sends TransactionEventResponse to the Charging Station |

## 8.5.2. Use case F02 sequence diagram, TxStartPoint=Authorized [549]

**NOTE** This an updated version of the diagram for Errata v2 for issue [549]. The original sequence diagram showed *evse* in the first TransactionEventRequest when there is not yet any cable plugged in.

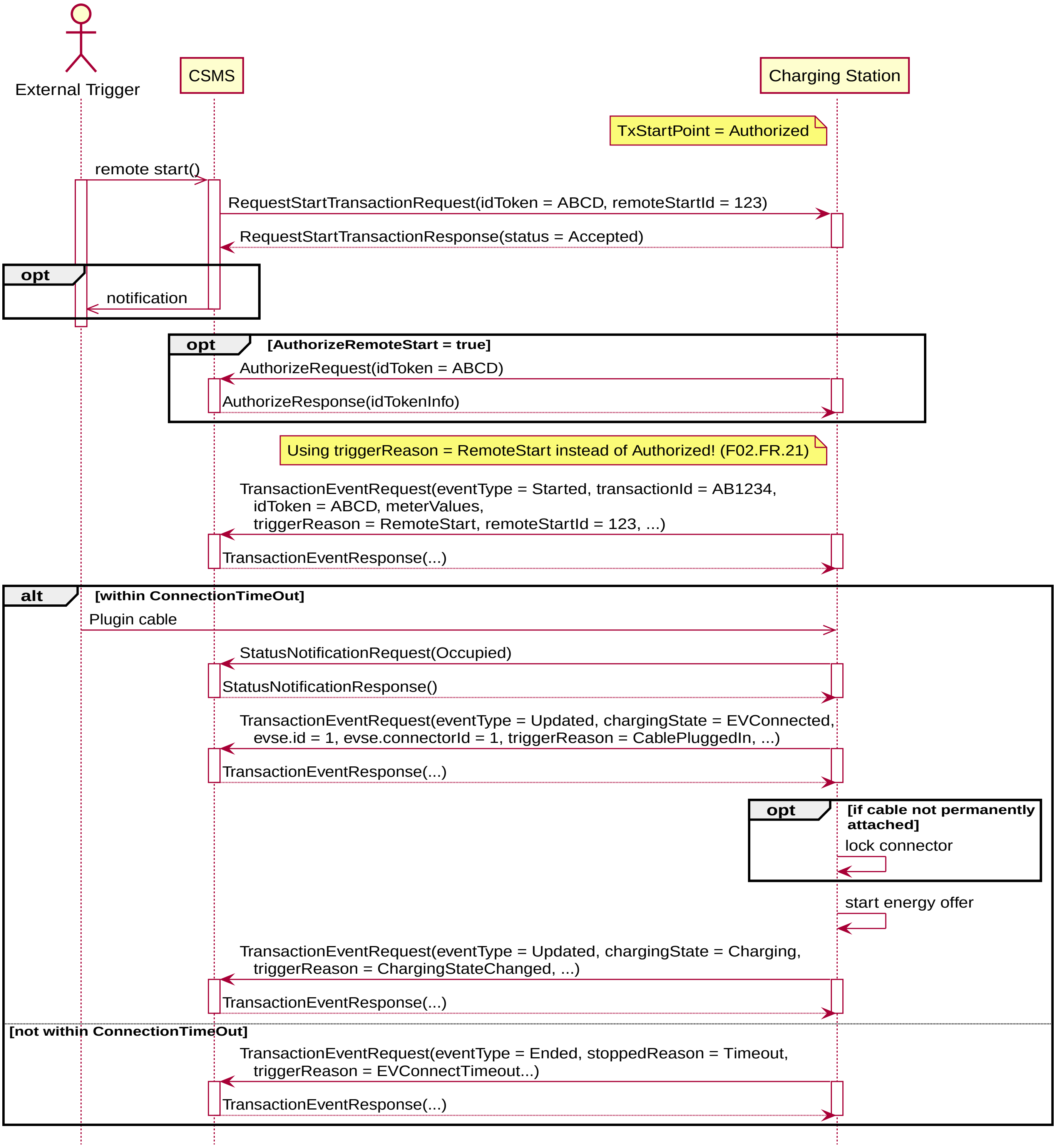

External Trigger
CSMS
Charging Station
TxStartPoint = Authorized
remote start()
RequestStartTransactionRequest(idToken = ABCD, remoteStartId = 123)
RequestStartTransactionResponse(status = Accepted)
opt
notification
opt
[AuthorizeRemoteStart = true]
AuthorizeRequest(idToken = ABCD)
AuthorizeResponse(idTokenInfo)
Using triggerReason = RemoteStart instead of Authorized! (F02.FR.21)
TransactionEventRequest(eventType = Started, transactionId = AB1234, idToken = ABCD, meterValues, triggerReason = RemoteStart, remoteStartId = 123, ...)
TransactionEventResponse(...)
alt
[within ConnectionTimeOut]
Plugin cable
StatusNotificationRequest(Occupied)
StatusNotificationResponse()
TransactionEventRequest(eventType = Updated, chargingState = EVConnected, evse.id = 1, evse.connectorId = 1, triggerReason = CablePluggedIn, ...)
TransactionEventResponse(...)
opt
[if cable not permanently attached]
lock connector
start energy offer
TransactionEventRequest(eventType = Updated, chargingState = Charging, triggerReason = ChargingStateChanged, ...)
TransactionEventResponse(...)
[not within ConnectionTimeOut]
TransactionEventRequest(eventType = Ended, stoppedReason = Timeout, triggerReason = EVConnectTimeout...)
TransactionEventResponse(...)

*Figure 3. Remote Start Transaction - Remote Start First with TxStartPoint=Authorized*

## 8.6. Page 171 - (v2) Use case F02: sequence diagram for TxStartPoint=EVConnected [579]

Use case F02 uses a TxStartPoint that contains `Authorized`. Since an authorization is done upon RequestStartTransactionRequest (it is like remotely presenting an idToken), this causes the transaction to be started.

If the TxStartPoint does not contain `Authorized`, but only `EVConnected`, then the sequence of events looks different, because the RequestStartTransactionRequest does not lead to starting a transaction as long as no cable is plugged in.

This errata entry clarifies this by adding a new sequence diagram for the situation where TxStartpoint = `EVConnected`.

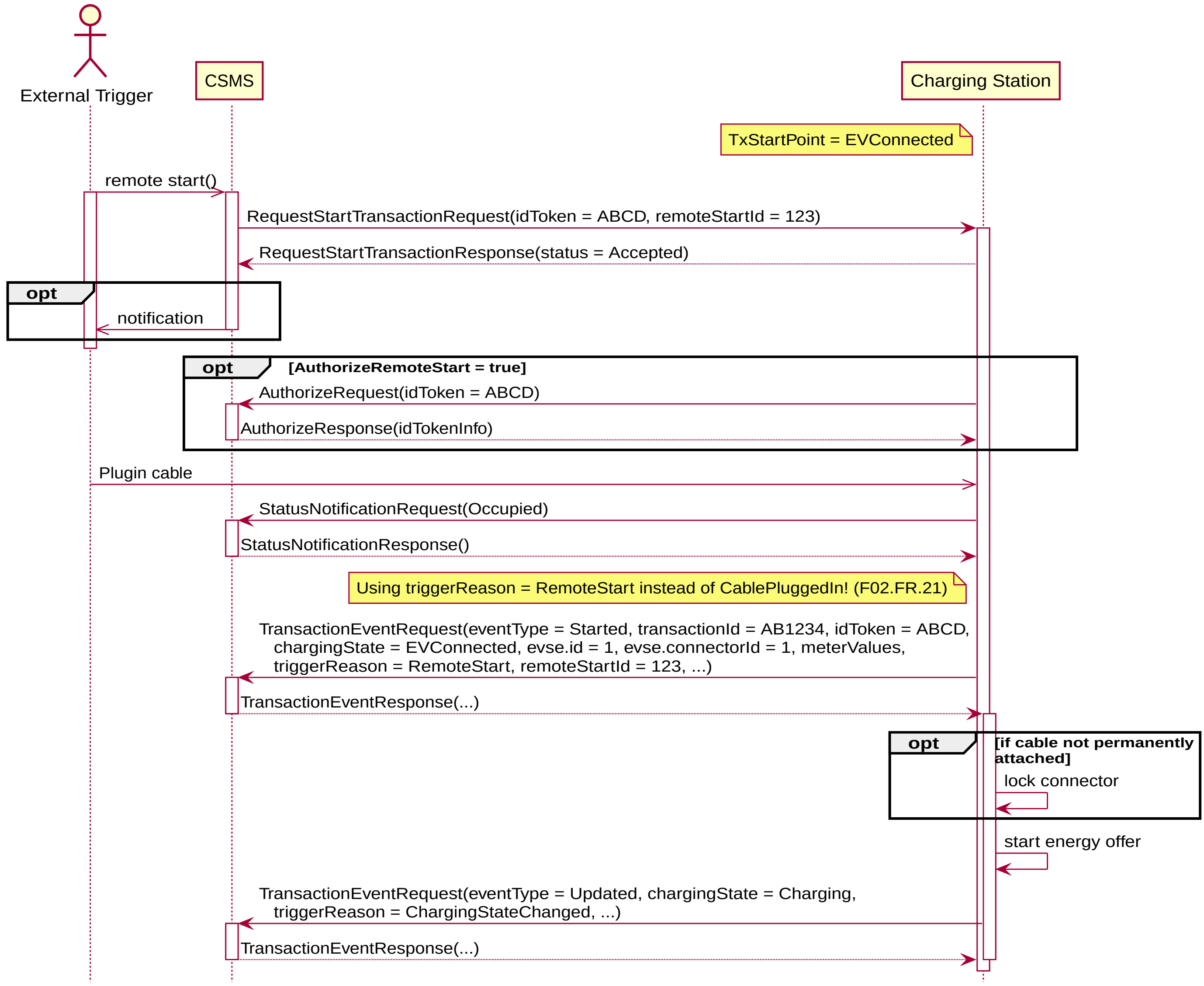

External Trigger
CSMS
Charging Station
TxStartPoint = EVConnected
remote start()
RequestStartTransactionRequest(idToken = ABCD, remoteStartId = 123)
RequestStartTransactionResponse(status = Accepted)
opt
notification
opt
[AuthorizeRemoteStart = true]
AuthorizeRequest(idToken = ABCD)
AuthorizeResponse(idTokenInfo)
Plugin cable
StatusNotificationRequest(Occupied)
StatusNotificationResponse()
Using triggerReason = RemoteStart instead of CablePluggedIn! (F02.FR.21)
TransactionEventRequest(eventType = Started, transactionId = AB1234, idToken = ABCD, chargingState = EVConnected, evse.id = 1, evse.connectorId = 1, meterValues, triggerReason = RemoteStart, remoteStartId = 123, ...)
TransactionEventResponse(...)
opt
[if cable not permanently attached]
lock connector
start energy offer
TransactionEventRequest(eventType = Updated, chargingState = Charging, triggerReason = ChargingStateChanged, ...)
TransactionEventResponse(...)

*Figure 4. Remote Start Transaction - Remote Start First with TxStartPoint=EVConnected*

## 8.7. Page 172 - (v1) Some requirements of use case E03 apply also to F02 [441]

**WARNING** F02.FR.07 has been updated in Errata v2: Page 139 - (v2) Handling of EVConnectionTimeout better defined [627].

The requirements E03.FR.01, E03.FR.04, E03.FR.05 from E03 "Start Transaction - idToken first" logically also apply to F02 "Remote Start Transaction - Remote Start First".

They are added as new requirements to F02.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F02.FR.05 | When the IdToken information is known. | The next TransactionEventRequest SHALL contain IdTokenType information. | |
| F02.FR.06 | This transaction ends a reservation for the specific IdToken. | The next TransactionEventRequest SHALL contain the reservationId. | See H. Reservation. |
| F02.FR.07 | When the EV Driver does not plug-in the Charging Cable before the timeout set by the Configuration Variable: `EVConnectionTimeOut` AND *status* of the connector is `Occupied` | The Charging Station SHALL send a StatusNotificationRequest with *status* set to `Available`, to the CSMS. | |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F02.FR.08 | F02.FR.07 | The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS. | |

## 8.8. Page 172 - (v2) Relevant F01 requirements copied to F02 Remote Start First [590]

A note below the requirements of F02 mentions that the F01 requirements also apply to F02. Not all F01 requirements can be applied as-is to F02, because the transaction starts at different points in time. This errata entry therefore adds all applicable F01 requirements as explicit new requirements for F02.

*New requirements (copied from F01 as appropriate)*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F02.FR.09 | If the value of `AuthorizeRemoteStart` = true AND Charging Station receives a RequestStartTransactionRequest | The Charging Station SHALL behave as if in response to a local action at the Charging Station to start a transaction after successful authorization of the IdToken given in RequestStartTransactionRequest message. | Charging Station will first respond to the request and then try to authorize the IdToken, using the Local Authorization List, Authorization Cache and/or an AuthorizeRequest.<br>A transaction is only started after authorization was obtained. |
| F02.FR.10 | If the value of `AuthorizeRemoteStart` = false AND<br>Charging Station receives a RequestStartTransactionRequest | The Charging Station SHALL start a transaction for the IdToken given in RequestStartTransactionRequest message without checking authorization. | Note that after the transaction has been started, the Charging Station will send a TransactionEventRequest with the idToken to the CSMS, and the CSMS will check the authorization status of the IdToken when processing this TransactionEventRequest. |
| F02.FR.11 | F02.FR.09 OR F02.FR.10 | The Charging Station SHALL send a TransactionEventRequest to the CSMS, and the CSMS will check the authorization status of the IdToken when processing this TransactionEventRequest. | |
| F02.FR.12 | | RequestStartTransactionRequest SHALL contain an IdToken, which Charging Station SHALL use, if it is able to start a transaction, in the TransactionEventRequest sent to the CSMS. | |
| F02.FR.13 | | The transaction SHALL be started in the same way as described in E03 - Start Transaction - Id Token First. | |
| F02.FR.14 | | RequestStartTransactionRequest MAY contain an evseId if the transaction is to be started on a specific EVSE. | When no evseId is provided, the Charging Station is in control of the EVSE selection. |
| F02.FR.15 | If the RequestStartTransactionRequest does not contain an evseId. | The Charging Station MAY reject the RequestStartTransactionRequest. | |
| F02.FR.16 | | The CSMS MAY include a ChargingProfile in the RequestStartTransactionRequest. | |
| F02.FR.17 | F02.FR.16 | The purpose of this ChargingProfile SHALL be set to TxProfile. | |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F02.FR.18 | F02.FR.16 | The Charging Station SHALL use this ChargingProfile for the transaction that is started by this RequestStartTransaction. | |
| F02.FR.19 | F02.FR.16 | The transactionId in the ChargingProfile SHALL NOT be set. | |
| F02.FR.20 | If a Charging Station without support for Smart Charging receives a RequestStartTransactionRequest with a ChargingProfile. | The Charging Station SHALL ignore the specified ChargingProfile. | The device model variable SmartChargingCtrlr.Enabled tells CSMS whether smart charging is supported. |
| F02.FR.21 | When a RequestStartTransactionRequest is received. | The next TransactionEventRequest SHALL contain *triggerReason*: `RemoteStart` and the *remoteStartId* from the RequestStartTransactionRequest. | This is to notify CSMS that this is the result of RequestStartTransaction.<br>Note, that if TxStartPoint=`EVConnected` the transaction will be started upon cable connection, but the *triggerReason* = `RemoteStart` must still be sent. The connection event is reported by the fact that *chargingState* = `EVConnected`. |
| F02.FR.22 | If the RequestStartTransactionRequest does not contain an *evseId* AND the Charging Station is capable of selecting an EVSE | The Charging Station SHALL select an EVSE to be used as a value for *evseId* for the operation | See also F02.FR.15 if Charging Station does not support starting at an arbitrary EVSE. |
| F02.FR.23 | When the *evseId* for RequestStartTransactionRequest is `Reserved` for an *idToken* that differs from *idToken* in the request AND has no reservation for a *groupIdToken* | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | |
| F02.FR.24 | When the *evseId* for RequestStartTransactionRequest is `Reserved` for an *idToken* that differs from *idToken* in the request AND is `Reserved` for a *groupIdToken* that differs from *groupIdToken* in the request | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | EV is not allowed to use station if neither *idToken* nor *idGroupToken* match the reservation. |
| F02.FR.25 | When the *evseId* for RequestStartTransactionRequest is `Unavailable` or `Faulted` | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | |
| F02.FR.26 | When the *evseId* for RequestStartTransactionRequest is `Occupied` AND this *evseId* has a transaction that has been authorized | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected`. | Only an EVSE with no transaction or with a transaction that has not yet been authorized can be matched with the RequestStartTransactionRequest |

## 8.9. Page 174 - (v2) Use case F03: replace TxStartPoint by TxStopPoint [557]

The remark of use case F03 refers to the configuration variable "TxStartPoint", but should refer to "TxStopPoint".

| Old text | 7 | **Remark(s)** | …<br>`TxStartPoint`: ParkingBayOccupancy, EVConnected<br>… |
|---|---|---|---|
| New text | 7 | **Remark(s)** | …<br>**`TxStopPoint`**: ParkingBayOccupancy, EVConnected<br>… |

## 8.10. Page 174 - (v2) Use case F03: Some requirements only valid for TxStopPoint = EVConnected [563]

Use case F03 is based on a TxStopPoint configuration that contains `EVConnected`. For a configuration where TxStopPoint contains `Authorized` or `PowerPathClosed` the behavior will be different. The following requirements have been worded differently, so that they remain valid with other TxStopPoint configurations.

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | F03.FR.02 | F03.FR.01 | The Charging Station SHALL stop the energy offer, unlock the cable and send a TransactionEventRequest (eventType = Updated) to the CSMS. | Cable unlocked if not permanently attached. |
| New text | F03.FR.02 | F03.FR.01 **AND TxStopPoint configuration does not cause the transaction to end (i.e., TxStopPoint does NOT contain `Authorized` or `PowerPathClosed`)** | The Charging Station SHALL stop the energy offer **and send a TransactionEventRequest (*eventType* = `Updated`, *triggerReason* = `RemoteStop`)** to the CSMS. | **For example when TxStopPoint = `EVConnected` the transaction will not be ended until EV is disconnected.** |
| Old text | F03.FR.03 | F03.FR.02 +<br>When the EV Driver unplugs the cable. | The Charging Station SHALL send a TransactionEventRequest (eventType = *Ended*, stoppedReason = *Remote*) to the CSMS. | |
| New text | F03.FR.03 | **F03.FR.01 AND TxStopPoint configuration causes the transaction to end (i.e., TxStopPoint contains `Authorized` or `PowerPathClosed`)** | The Charging Station SHALL send a **TransactionEventRequest (*eventType* = Ended, *triggerReason* = `RemoteStop`, *stoppedReason* = `Remote`)** to the CSMS. | For example when TxStopPoint = `EVConnected` and EV is disconnected after the RequestStopTransactionRequest. |

Requirement F03.FR.10 can be deleted, because it is already covered by F03.FR.03.

*Deleted requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Delete | F03.FR.10 | | The Charging Station SHALL include the stoppedReason element in the TransactionEventRequest (eventType = Ended). What reason to use is described in the description of reasonEnumType. | |

## 8.11. Page 180 - (v1) Requirement F06.FR.07 description improvement [451]

*Changed requirement*

|  | **ID** | **Precondition** | **Requirement definition** | **Note** |
|---|---|---|---|---|
| Old text | F06.FR.07 | If a Charging Station receives a TriggerMessageRequest with *requestedMessage* set to: *TransactionEvent* | The Charging Station SHALL send a TransactionEventRequest to the CSMS with the current status of the transaction, and the most recent measurements for all measurands configured in Configuration Variable: `SampledDataTxUpdatedMeasurands`. |  |
| New text | F06.FR.07 | If a Charging Station receives a TriggerMessageRequest with *requestedMessage* set to: *TransactionEvent* | The Charging Station SHALL send a TransactionEventRequest to the CSMS with ***triggerReason = Trigger*****,** ***transactionInfo*** **with at least the** ***chargingState*****, and** ***meterValue*** **with** the most recent measurements for all measurands configured in Configuration Variable: `SampledDataTxUpdatedMeasurands`. |  |

## 8.12. Page 180 - (v1) Note to requirement F06.FR.11 is contradicting F06.FR.12

*Changed requirement*

|  | **ID** | **Precondition** | **Requirement definition** | **Note** |
|---|---|---|---|---|
| Old text | F06.FR.11 | If the field evse is relevant but absent in the TriggerMessageRequest. | The Charging Station SHALL interpret this as "for all allowed evse values". | For example, a request for a statusNotification without evse is a request for multiple statusNotifications: a notification for each Connector of each EVSE. |
| New text | F06.FR.11 | If the field evse is relevant but absent in the TriggerMessageRequest. | The Charging Station SHALL interpret this as "for all allowed evse values". | **StatusNotifications can only be requested for a specific connector, see F06.FR.12/13** |

## 8.13. Page 180 - (v1) Requirement F06.FR.12: evseId cannot be 0 [450]

The field *evse.id* (from EVSEType) is always greater than zero, therefore the requirement F06.FR.12 with a precondition about *evse.id = 0* is wrong.

*Changed requirement*

|  | **ID** | **Precondition** | **Requirement definition** | **Note** |
|---|---|---|---|---|
| Old text | F06.FR.12 | If a Charging Station receives a TriggerMessageRequest with *requestedMessage* set to:<br>*StatusNotification* AND *evse.id* is set to 0 | The Charging Station SHALL respond with a TriggerMessageResponse with status *Rejected*. | StatusNotification messages can only be sent at connector level. |
| New text | F06.FR.12 | If a Charging Station receives a TriggerMessageRequest with *requestedMessage* set to:<br>*StatusNotification* AND<br>**(** ***evse*** **is omitted OR** ***evse.connectorId*** **is omitted )** | The Charging Station SHALL respond with a TriggerMessageResponse with status *Rejected*. | StatusNotification messages can only be sent at connector level. |

# 9. Use case G Availability

## 9.1. Page 184 - (v2) Use case G01: added alternative scenario using NotifyEvent [530]

Use case G01 describes how connector status can be reported using the StatusNotification message. A StatusNotification can only report connector status and cannot be used to report the status of an EVSE or charging station.

The availability status of a charging station, EVSE or connector can also be reported as a change to the variable "AvailabilityState" that exist in the device model of the charging station.

The alternative scenario is added as follows:

| No. | Type | Description |
|---|---|---|
| 1 | **Name** | Status Notification |
| ... | | |
| 4 | **Description** | ... |
| | *Alternative scenario* | **1.** Instead of a StatusNotificationRequest a Charging Station can send a NotifyEventRequest with *trigger* = `Delta` for *component.name* = "Connector" and the EVSE number in *evse.id* and the connector number in *evse.connectorId*, and *variable* = "AvailabilityState" with the value of the new status to the CSMS.<br>**1a.** Optionally, Charging Station can also include the state of *component* = "ChargingStation" and *component* = "EVSE" in the NotifyEventRequest.<br>**2.** The CSMS responds with NotifyEventResponse to the Charging Station. |

## 9.2. Page 184 - (v2) Missing requirements for StatusNotification [569]

The implicitly assumed requirement that a StatusNotificationRequest or NotifyEventRequest must be sent when the status of a plug changes, is missing. This is made explicit with the following requirements.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| G01.FR.03 | The connector is `Available` when an EV is connecting | The Charging Station SHALL send a StatusNotificationRequest with *connectorStatus* = `Occupied`<br>or a NotifyEventRequest for *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Occupied" and *trigger* = "Delta". |
| G01.FR.04 | The connector is `Occupied` when an EV is disconnecting AND<br>connector is not scheduled to become `Unavailable` (G03.FR.05) | The Charging Station SHALL send a StatusNotificationRequest with *connectorStatus* = `Available` when an EV is disconnected<br>or a NotifyEventRequest for *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Available" and *trigger* = "Delta". |
| G01.FR.05 | The connector is `Occupied` when an EV is disconnecting AND<br>connector is scheduled to become `Unavailable` (G03.FR.05) | The Charging Station SHALL send a StatusNotificationRequest with *connectorStatus* = `Unavailable` when an EV is disconnected<br>or a NotifyEventRequest for *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Unavailable" and *trigger* = "Delta". |
| G01.FR.06 | The connector is `Reserved` when an EV is connecting AND<br>EV driver presents an IdToken matching the reservation | The Charging Station SHALL send a StatusNotificationRequest with *connectorStatus* = `Occupied`<br>or a NotifyEventRequest for *component* = "Connector", *variable* = "AvailabilityState", *actualValue* = "Occupied" and *trigger* = "Delta". |
| G01.FR.07 | When a ChangeAvailabilityRequest leads to a connector status change | The Charging Station SHALL send a StatusNotificationRequest with the corresponding *connectorStatus*<br>or a NotifyEventRequest for *component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta" and the corresponding *actualValue* of "AvailabilityState". |

### 9.2.1. State transition overview for connecting/disconnecting

| Initial | EV connects | EV disconnects |
|---|---|---|
| Available | → Occupied | - |

| Initial | EV connects | EV disconnects |
|---|---|---|
| Occupied | - | → Available<br>( → Unavailable, if scheduled to become Unavailable) |
| Reserved | -<br>( → Occupied, only if authorized for reserved IdToken ) | - |
| Unavailable | - | - |
| Faulted | - | - |

## 9.3. Page 184 - (v2) StatusNotifications for multiple connectors on an EVSE [530]

The remark of use case G01 says:

> The Charging Station MAY use the *Unavailable* status internally for other purposes (e.g. while updating firmware or waiting for an initial *Accepted* RegistrationStatus). When one of the connectors on an EVSE is Reserved/Occupied, the CSMS has to take care of the status of the other connectors when presenting availability information to another system or user. The CSMS knows which connectors belong to the same EVSE.

On an EVSE with multiple connectors, when an EV is plugged in at one of the connectors (C1), then the other connector (C2) can no longer be used. According to the above remark the Charging Station will only send a StatusNotification(Occupied) for C1. No StatusNotification is sent for C2, because CSMS will know that C2 is no longer available at this moment.

A requirement is added to make this explicit.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| G01.FR.08 | When a connector of an EVSE becomes reserved or a cable is plugged-in AND<br>The EVSE has multiple connectors | The Charging Station SHOULD NOT send a StatusNotificationRequest for the other connector(s), even though they are no longer usable. |

## 9.4. Page 188 - (v1) Precondition of G03.FR.05 is incomplete [368]

The requirement G03.FR.03 conflicts with G03.FR.05 when a transaction is active. It is corrected as follows:

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | G03.FR.05 | When a transaction is in progress. | The Charging Station SHALL respond with availability status *Scheduled* to indicate that it is scheduled to occur after the transaction has finished. | |
| New text | G03.FR.05 | When a transaction is in progress **AND NOT G03.FR.03** | The Charging Station SHALL respond with availability status *Scheduled* to indicate that it is scheduled to occur after the transaction has finished. | |

# 10. Use case H Reservation

## 10.1. Page 197 - (v1) Merging two requirements [445]

The requirements H01.FR.11 and H01.FR.13 should be merged into one, because now they are incorrect in the situation where, for example, a Charging Station has two EVSEs of which one is *Occupied* and the other is *Reserved*. It should state that the Charging Station should return *Occupied* when each of the targeted EVSEs has a status *Reserved* or *Occupied*.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | H01.FR.11 | When receiving a ReserveNowRequest AND (all) targeted EVSEs have status *Reserved* | The Charging Station SHALL return *Occupied*. | |
| New text | H01.FR.11 | When receiving a ReserveNowRequest AND (all) targeted EVSEs have status *Reserved* **or *Occupied*** | The Charging Station SHALL return *Occupied*. | |

The following requirement becomes obsolete:

*Deleted requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Deleted | H01.FR.13 | When receiving a ReserveNowRequest AND (all) targeted EVSEs have status *Occupied* | The Charging Station SHALL return *Occupied*. | |

# 10.2. Page 197 - (v1) Missing requirement about reserving for *evseId* [472, 505]

A requirement for reserving a specific *evseId* (scenario S2) is missing.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| H01.FR.23 | If the Charging Station receives a ReserveNowRequest for *evseId* AND this EVSE is *Available* | The Charging Station SHALL respond with a ReserveNowResponse with status *Accepted* AND SHALL send a StatusNotificationRequest with *connectorStatus* = `Reserved` for all connectors of the EVSE. | If an EVSE is reserved, all of its connectors are reported as reserved. |

# 10.3. Page 197 - (v1) Requirement H01.FR.20 needs to be split [447, 472]

The handling of a maximum amount of reservations without *evseId* and reservations for a *connectorType* is actually slightly different and should not be handled in one requirement.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | H01.FR.20 | H01.FR.04 OR H01.FR.06<br>AND<br>amount of EVSEs available equals the amount of reservations | The Charging Station SHALL set all available EVSEs to *Reserved*. | |
| Old text | H01.FR.20 | **H01.FR.04**<br>AND<br>amount of EVSEs available equals the amount of reservations | The Charging Station SHALL send a StatusNotificationRequest with *connectorStatus* = `Reserved` for all connectors of the EVSE. | If an EVSE is reserved, all of its connectors are reported as reserved. |

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| H01.FR.24 | H01.FR.06<br>AND<br>amount of reservations for a specific *connectorType* equals the amount of available EVSEs with that specific *connectorType* | The Charging Station SHALL send a StatusNotificationRequest with *connectorStatus* = `Reserved` for all connectors of the EVSEs with the specific *connectorType*. | If an EVSE is reserved for a specific *connectorType*, all connectors on the EVSE are reported as reserved. |

## 10.4. Page 197 - (v2) Using field *reservationId* instead of *id* in ReserveNow [584]

Some requirements in H01 erroneously refer to *reservationId* instead of *id*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | H01.FR.02 | If the reservationId in the ReserveNowRequest matches a reservation in the Charging Station. | The Charging Station SHALL replace that reservation with the new reservation in the request. | |
| New text | H01.FR.02 | If the ***id*** in the ReserveNowRequest matches a reservation in the Charging Station. | The Charging Station SHALL replace that reservation with the new reservation in the request. | |
| Old text | H01.FR.03 | If the reservationId in the ReserveNowRequest does not match any reservation in the Charging Station. | The Charging Station SHALL return the status value *Accepted* if it succeeds in reserving an EVSE. | |
| New text | H01.FR.03 | If the ***id*** in the ReserveNowRequest does not match any reservation in the Charging Station. | The Charging Station SHALL return the status value *Accepted* if it succeeds in reserving an EVSE. | |

## 10.5. Page 203 - (v2) incorrect dataType reference AvailabilityType should be operationalStatus

| | Description |
|---|---|
| Old text | **1.** The CSMS sends ChangeAvailabilityRequest requesting a Charging Station to change the availability of an EVSE or Connector.<br>**2.** The Charging Station changes the availability to the EVSE/Connector to the requested AvailabilityType from the ChangeAvailabilityRequest.<br>**3**. Upon receipt of ChangeAvailabilityRequest, the Charging Station responds with ChangeAvailabilityResponse. In case that the status 'Scheduled' is reported in the ChangeAvailabilityResponse, a transaction was running and this will be finished first.<br>**4**. The Charging Station reports the status of the EVSE/Connector using a StatusNotification. |
| New text | **1.** The CSMS sends ChangeAvailabilityRequest requesting a Charging Station to change the availability of an EVSE or Connector.<br>**2.** The Charging Station changes the availability to the EVSE/Connector to the requested **operationalStatus** from the ChangeAvailabilityRequest.<br>**3**. Upon receipt of ChangeAvailabilityRequest, the Charging Station responds with ChangeAvailabilityResponse. In case that the status 'Scheduled' is reported in the ChangeAvailabilityResponse, a transaction was running and this will be finished first.<br>**4**. The Charging Station reports the status of the EVSE/Connector using a StatusNotification. |

# 11. Use case I Tariff And Cost

# 11.1. Page 207 - (v1) Use case I02: Show EV Driver Running Total Cost During Charging in TransactionEventResponse [361]

Use case I02 explains how running cost can be reported with the CostUpdatedRequest. This message is useful when reporting of running cost is not done in synchronisation with TransactionEvents. Since a TransactionEventResponse contains a *totalCost* field, this can also be used to provide running cost updates. This is not mentioned in the use case and its requirements. Therefore, the use case I02 is extended with the following *Alternative scenario*:

| No. | Type | Description |
|---|---|---|
| **4** | **Description** | While a transaction is ongoing, the driver wants to know how much the running total cost is, updated at a relevant interval. |
|  | Alternative scenario | **1.** Upon receipt of a TransactionEventRequest with *eventType* = `Updated` the CSMS returns the running cost corresponding to the *timestamp* and *meterValue* in the field *totalCost* in the TransactionEventResponse.<br>**2.** The Charging Station shows the current total cost to the EV Driver. |

## 11.1.1. Page 208 - (v1) I02 changed requirements

The requirements for I02 are extended to support using the TransactionEventResponse message to report a running cost.

*Changed requirement*

|  | ID. | Precondition | Requirements |
|---|---|---|---|
| Old text | I02.FR.01 |  | The CSMS SHALL send CostUpdatedRequest at a relevant interval/moment, this might depend on the charging speed, running cost, etc. |
| New text | I02.FR.01 |  | The CSMS SHALL send **either a** CostUpdatedRequest at a relevant interval/moment **or return the running cost in a TransactionEventResponse.** This might depend on the charging speed, running cost, etc. |

*New requirement*

| ID. | Precondition | Requirements |
|---|---|---|
| I02.FR.04 | When running cost is reported in TransactionEventResponse | The Charging Station SHALL show the current running cost to the EV Driver. |

## 11.1.2. Page 367 - (v1) TransactionEventResponse

|  | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **totalCost** | decimal | 0..1 | Optional. SHALL only be sent when charging has ended. Final total cost of this transaction, including taxes. In the currency configured with the Configuration Variable: `Currency`. When omitted, the transaction was NOT free. To indicate a free transaction, the CSMS SHALL send 0.00. |
| New text | **totalCost** | decimal | 0..1 | Optional. When *eventType* of TransactionEventRequest is `Updated`, then this value contains the *running cost*.<br>When *eventType* of TransactionEventRequest is `Ended`, then this contains the final *total cost* of this transaction, including taxes, in the currency configured with the Configuration Variable: `Currency`. Absence of this value does not imply that the transaction was free. To indicate a free transaction, the CSMS SHALL send a value of 0.00. |

# 12. Use case J Meter Values

## 12.1. Page 214 - (v1) Limit the amount of meter values in TransactionEvent(Ended) [371]

### 12.1.1. Section 2.1

The text in section 2.1 Transaction Meter Values describes how `SampledDataTxEndedMeasurands` can be used to configure which measurands are sent at the end of the transaction in a TransactionEventRequest(eventType=`Ended`) message. Unlike a TransactionEventRequest(eventType=`Updated`), which can be repeated if the amount of measurands is large, there is only one TransactionEventRequest for eventType=`Ended`. This means, that care should be taken to ensure that the amount of measurands that is expected at the end of a transaction fits in one message.

After this paragraph:

"`SampledDataTxEndedMeasurands` is a comma separated list that prescribes the sampled measurands to be included in the *meterValues* field of a TransactionEventRequest (eventType = Ended), these measurands have to be taken every `SampledDataTxEndedInterval` seconds from the start of the transaction, and will only be sent in the TransactionEventRequest (eventType = Ended)."

enter the following text:

> Care should be taken to ensure that the amount of measurands that is expected at the end of a transaction fits in one TransactionEventRequest(eventType=`Ended`) message. Keep the number of measurands in `SampledDataTxEndedMeasurands` to a minimum and configure a large interval in `SampledDataTxEndedInterval` to keep the number of samples small.

### 12.1.2. Requirements J02

A new requirement is needed to tell what to do in the event that the number of sampled measurands in the TransactionEventRequest(eventType=`Ended`) becomes too large.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| J02.FR.20 | When configured to send meter data in the TransactionEventRequest (eventType = Ended) AND amount of meter data is too much for one TransactionEventRequest (eventType = Ended) message | Charging Station MAY remove samples until it fits in a message. When removing samples, the Charging Station SHOULD remove intermediate samples first (for example: 2nd sample, 4th sample, 6th sample etc.). | Samples should be removed in a way that it does not affect billing. See also E06.FR.12. |

## 12.2. Page 215 - (v2) section 2.2: Incorrect sentence in description for AlignedDataTxEndedInterval [624]

The last sentence of the last paragraph on page 215 is not valid and has been removed: It is not possible to provide clock-aligned measurands for only start and end of the transaction, because then these measurands would no longer be clock-aligned.

> `AlignedDataTxEndedInterval` is the size of the clock-aligned data interval (in seconds). This defines the set of evenly spaced meter data aggregation intervals per day, starting at 00:00:00 (midnight) intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. ~~A value of "0" (numeric zero), by convention, is to be interpreted to mean that only the values taken at the start and end of a transaction should be transmitted (no intermediate values).~~

### 12.2.1. Page 439 - AlignedDataTxEndedInterval

### 12.2.2. `AlignedDataTxEndedInterval`

For the same reason as above, the last sentence of the description of AlignedDataTxEndedInterval has been removed.

| | | | |
|---|---|---|---|
| **Required** | yes | | |
| **Component** | **componentName** | AlignedDataCtrlr | |
| **Variable** | **variableName** | TxEndedInterval | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **unit** | seconds |
| | | **dataType** | integer |
| **Description** | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals.<br>When clock aligned data is being collected, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard. All "per-period" data (e.g. energy readings) should be accumulated (for "flow" type measurands such as energy), or averaged (for other values) across the entire interval (or partial interval, at the beginning or end of a transaction), and transmitted (if so enabled) at the end of the transaction in 1 TransactionEventRequest (eventType = Ended) message.<br>~~+ A value of "0" (numeric zero), by convention, is to be interpreted to mean that only the values taken at the *start* and *end* of a transaction should be transmitted (no intermediate values).~~ | | |

## 12.3. Page 216 - (v1) section 2.3: Including phases for register meter values [328]

The text in section 2.3 states that meter values need to be reported for all phases. This is not very convenient for register meter values, which are mostly used for billing purposes. A new configuration variable will allow the charging station to send only the total energy register value, without including each phase.

The following text is added to section 2.3:

> "When the configuration variable `SampledDataRegisterValuesWithoutPhases` has the value *true*, then meter values of measurand `Energy.Active.Import.Register` will only report the total energy over all phases without reporting the individual phase values."

Below follows a description of this variable of the SampleDataCtrlr component.

### 12.3.1. Page 439 - (v1) New configuration variable: SampledDataRegisterValuesWithoutPhases

This new configuration variable defaults to *false*, such that behavior is unchanged when it is not present.

**2.7.18 SampledDataRegisterValuesWithoutPhases**

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | SampledDataCtrlr | |
| **Variable** | **variableName** | RegisterValuesWithoutPhases | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | If this variable reports a value of *true*, then meter values of measurand `Energy.Active.Import.Register` will only report the total energy over all phases without reporting the individual phase values.<br>If this variable is absent or *false*, then the value for each phase is reported, possibly also with a total value (depending on the meter). | | |

## 12.4. Page 218 - (v1) Missing requirement for dealing with AlignedDataDuringIdle [457]

The configuration variable `AlignedDataDuringIdle` has the following description:

> If set to true, the Charging Station SHALL NOT send clock aligned meter values when a transaction is ongoing. When an EVSE is specified, it SHALL stop sending the clock aligned meter values for this EVSE when it has an ongoing transaction. When no EVSE is specified, it SHALL stop sending the clock aligned meter values when any transaction is ongoing on this Charging Station.

This behavior is made explicit with the following new requirements for use case J01.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| J01.FR.19 | If `AlignedDataSendDuringIdle` is set to true for an EVSE AND the specified EVSE has an ongoing transaction. | The Charging Station SHALL stop sending the clock aligned meter values for this EVSE. | |
| J01.FR.20 | If `AlignedDataSendDuringIdle` is set to true for a Charging Station AND the Charging Station has an ongoing transaction. | The Charging Station SHALL stop sending the clock aligned meter values for all EVSEs and the main power meter. | |

**NOTE** `AlignedDataSendDuringIdle` is the variable SendDuringIdle of the AlignedDataCtrlr. This controller can belong to an EVSE or exist at the top level, in which case it refers to all EVSEs in the Charging Station.

## 12.5. Page 218 - (v1) Wrong placement of Aligned/SampledDataSignReadings

Two requirements about Aligned/SampledDataSignReadings are in the wrong use case. They need to be swapped.

### 12.5.1. Page 218 - (v1) Requirement J01.FR.16 belongs to use case J02

Requirement J01.FR.16 is moved to J02 as requirement J02.FR.21.

### 12.5.2. Page 220 - (v1) Requirement J02.FR.15 belongs to use case J01

Requirement J02.FR.15 is moved to J01 as requirement J01.FR.21

## 12.6. Page 220 - (v2) Use case J02: improved definition of J01.FR.11 [535]

J02.FR.11 might suggest that every TransactionEventRequest(Updated) should have meter values included. That is not correct. Meter values shall only be included in TransactionEventRequest(Updated) messages sent on a periodic basis during the transaction.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | J02.FR.11 | When configured to send meter data in the TransactionEventRequest(eventType = Updated), See: Meter Values - Configuration | The Charging Station SHALL add the configured measurands to the optional meterValue field in the TransactionEventRequest(eventType = Updated) sent to the CSMS to provide more details during the transaction. | |

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New text | J02.FR.11 | When configured to send meter data in the TransactionEventRequest(eventType = Updated) **AND**<br>**When the interval in** **`SampledDataTxUpdatedInterval`** **has elapsed**<br>**(See: Meter Values - Configuration )** | The Charging Station SHALL **send a TransactionEventRequest(eventType = Updated with *triggerReason* =** **`MeterValuePeriodic`** **with the configured measurands in the *meterValue* field.** |  |

# 12.7. Page 221 - (v1) Use case J03 description of dealing with ISO 15118 signed metering receipts is unclear [352, 353]

Use case J03 describes how receipt of a meter value from the fiscal meter of an EVSE can be confirmed by the EV by signing the MeteringReceiptReq message towards the Charging Station. The use case description and requirement J03.FR.04 require that this meter value be sent to CSMS, but that is not correct, because the Charging Station already sends meter values to CSMS as part of TransactionEventRequests and at a frequency that differs from ISO 15118.

## 12.7.1. Use case J03 row #7

| No. | Type | Description |
|---|---|---|
| 7 | **Combined scenario description** | **15118**<br>**1.** The EV sends a ChargingStatusReq (in case of AC charging) message to the Charging Station.<br>**2.** The EV sends a MeteringReceiptReq to the Charging Station.<br>**OCPP**<br>**3.** Between the Charging Station and the CSMS, the TransactionEventRequest(eventType = Updated) message is being exchanged.<br>When sending a MeteringReceiptReq message the EV acknowledges that the data elements MeterInfo record, SessionID and the SAScheduleTupleID included in the ChargingStatusRes message prior to this request have been received from the Charging Station. This confirmation is implemented by applying a signature to the message body of the MeteringReceiptReq message. |

In the above shown row #7 of the use case the following text needs to be changed in order to be more complete.

|  |  |
|---|---|
| Old text | **15118**<br>**1.** The EV sends a ChargingStatusReq (in case of AC charging) message to the Charging Station.<br>**2.** The EV sends a MeteringReceiptReq to the Charging Station.<br>**OCPP**<br>**3.** Between the Charging Station and the CSMS, the TransactionEventRequest(eventType = Updated) message is being exchanged. |
| New text | **15118**<br>**1a.** The EV sends a ChargingStatusReq (in case of AC charging) message to the Charging Station, upon which EVSE returns a ChargingStatusRes containing the meter value from the fiscal meter.<br>**1b.** The EV sends a CurrentDemandReq (in case of DC charging) message to the Charging Station, upon which EVSE returns a CurrentDemandRes containing the meter value from the fiscal meter.<br>**2.** The EV sends a MeteringReceiptReq to the Charging Station to acknowledge receipt of the meter value.<br>~~**OCPP**~~<br>~~**3.** Between the Charging Station and the CSMS, the TransactionEventRequest(eventType = Updated) message is being exchanged.~~ |

## 12.7.2. Use case J03 row #10

|  |  |  |
|---|---|---|
| 10 | **Remark(s)** | The MeteringReceiptReq message in ISO 15118 only applies to ChargingStatusReq (AC), because CurrentDemandReq (for DC) does not contain meter values. |

This is incorrect. CurrentDemandRes does contain optional meter values, so it is possible to sign/confirm meter values with DC charging.

### 12.7.3. Requirement J03.FR.03

Requirement J03.FR.04 states that a Charging Station must send the fiscal meter value in a MeteringReceiptReq message, received from an ISO 15118 transaction, to CSMS. This is not correct. Fiscal meter values are, like any other meter values, sent to CSMS as described in use case J02.

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | J03.FR.04 | When the Charging Station receives ISO 15118 signed meter values | The Charging Station SHALL pass them to CSMS in a TransactionEventRequest (eventType = Updated) message. |
| New text | J03.FR.04 | When the Charging Station receives ISO 15118 signed **MeteringReceiptReq message from EV** | The Charging Station **SHOULD NOT pass the meter value from the MeteringReceiptReq message** to CSMS in a TransactionEventRequest (eventType = Updated) message. **Instead, Charging Station sends transaction-related meter values as described in use case J02.** |

**NOTE** The above does not imply that a Charging Station cannot require EV to send MeteringReceiptReq messages. An implementation at a Charging Station can be such, that every meter value from the fiscal meter that is send to CSMS (as per use case J02) must first have been acknowledged by a MeterReceiptReq from the EV.

# 13. Use case K Smart Charging

## 13.1. Page 231 - (v2) Chapter K.2.6.1: error in JSON examples [552]

The JSON examples on page 231 contained an error: the *chargingSchedule* field was not represented as an array and it was missing the *id* field.

The correct examples are shown below:

**(1) TxDefaultProfile, stack #1:** time-of-day limitation to 2 kW, recurring every day from 17:00h to 20:00h.

```
"chargingProfile": {
  "id": 10, "stackLevel": 1, "chargingProfilePurpose": "TxDefaultProfile",
  "chargingProfileKind": "Recurring", "recurrencyKind": "Daily",
  "chargingSchedule": [ {
    "id": 1, "startSchedule": "2020-01-09T17:00:00", "duration": 1080,
    "chargingRateUnit": "W",
    "chargingSchedulePeriod": [ { "startPeriod": 0, "limit": 2000 } ]
  } ]
}
```

**(2) TxDefaultProfile, stack #2:** overruling Sundays to no limit, recurring every week starting 2020-01-05.

```
"chargingProfile": {
  "id": 11, "stackLevel": 2, "chargingProfilePurpose": "TxDefaultProfile",
  "chargingProfileKind": "Recurring", "recurrencyKind": "Weekly",
  "chargingSchedule": [ {
    "id": 1, "startSchedule": "2020-01-05T00:00:00", "duration": 86400,
    "chargingRateUnit": "W",
    "chargingSchedulePeriod": [ { "startPeriod": 0, "limit": 999999 } ]
  } ]
}
```

**(3) TxDefaultProfile, stack #3:** overruling Christmas Day 2020 to no limit, fixed date 2020-12-25.
Note, that this profile is only valid in the year 2020.

```
"chargingProfile": {
  "id": 12, "stackLevel": 3, "chargingProfilePurpose": "TxDefaultProfile",
  "chargingProfileKind": "Absolute",
  "validFrom": "2020-01-01T00:00:00", "validTo": "2021-01-01T00:00:00",
  "chargingSchedule": [ {
    "id": 1, "startSchedule": "2020-12-25T00:00:00", "duration": 86400,
    "chargingRateUnit": "W",
    "chargingSchedulePeriod": [ { "startPeriod": 0, "limit": 999999 } ]
  } ]
}
```

## 13.2. Page 233 - (v1) Use case K01: Recommendation to not limit duration of TxProfile [519]

**WARNING** This errata has been superseded by Page 233 - (v2) Transaction does not fall back to TxDefaultProfile when TxProfile ends early [603]

A ChargingProfile of *chargingProfilePurpose* = `TxProfile` is only valid for the specified transaction and ceases to be valid when the transaction ends.

> It is recommended to omit the *duration* field of the ChargingSchedule, so that it automatically lasts until end of the transaction, because the behavior when a `TxProfile` ends before end of the transaction is **not** specified.

Chapter K3.5 Combining Charging Profile Purposes mentions that a `TxProfile` overrules a `TxDefaultProfile`, but this can be interpreted as either overruling for the entire transaction or as overruling for the duration of the `TxProfile`.

> In the event that a `TxProfile` is received with a *duration* shorter than the transaction, it is recommended to activate the `TxDefaultProfile` when `TxProfile` ends, if it exists and is valid at that point in time.

## 13.3. Page 233 - (v2) Transaction does not fall back to TxDefaultProfile when TxProfile ends early [603]

The recommendations given in Page 233 - (v1) Use case K01: Recommendation to not limit duration of TxProfile [519] above do not cover the full scope of the situation.

*K02 Remark addition:*

> It is recommended to omit the duration field of the ChargingSchedule from a TxProfile, so that it automatically lasts until the end of the transaction. If the TxProfile expires before the transaction ends, it falls back to the lowest limit of the active TxDefaultProfile and ChargingStationMaxProfile. If there are no other active profiles, it falls back to the local limit of the Charging Station.

*Paragraph 3.2 of chapter K. Smart Charging in OCPP 2.0.1:*

> A transaction-specific profile with purpose *TxProfile* overrules the TxDefaultProfile for the duration of the current transaction only or until the TxProfile expires, whichever occurs earlier.

For clarity the recommendation regarding the fall back is also added as a new requirement:

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K02.FR.08 | K02.FR.04 AND<br>The charging schedule of TxProfile ends, before the transaction ends, because the set duration or validTo period expired | The Charging Station SHALL fall back to using the lowest limit of the active TxDefaultProfile and ChargingStationMaxProfile. If there are no other active profiles, it falls back to the local limit of the Charging Station | |

## 13.4. Page 234 - (v1) Use case K01: SetChargingProfile cannot replace external constraints. [507]

A charging profile of purpose `ChargingStationExternalConstraints` is never managed by CSMS. It is created internally in a Charging Station when it receives an external limit, e.g. via IEC61850. This purpose is only used in ReportChargingProfiles to report such a limit. When reported, the external constraints profile has a charging profile id. This id shall not be used by CSMS in a SetChargingProfileRequest, because it cannot be overwritten.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | K01.FR.05 | When a SetChargingProfileRequest with an already known ChargingProfile.*id* is received | The Charging Station SHALL replace the existing ChargingProfile with the one specified. | External charging limits can change during transaction, therefore updates should be possible. |
| New text | K01.FR.05 | When a SetChargingProfileRequest with an already known ChargingProfile.*id* is received **AND the existing ChargingProfile does NOT have chargingProfilePurpose = `ChargingStationExternalConstraints`** | The Charging Station SHALL replace the existing ChargingProfile with the one specified. | **ChargingStationExternal Constraints profile cannot be replaced.** |

## 13.5. Page 234 - (v2) Use case K01: Improved explanation of TxDefaultProfile on evseId #0 [593]

**NOTE** The below only apples to TxDefaultProfile. It is not an issue for a ChargingStationMaxProfile, which is ONLY valid for #0.

The OCPP specification is ambiguous on how to deal with TxDefaultProfile charging profiles that are submitted to EVSE #0. It says that "The charging station SHALL apply this profile to all EVSEs". However, it is not clear what "apply" means, and not defined what happens when both EVSE #0 and EVSE #1 have a TxDefaultProfile.

The following describes what is meant by “apply” in this situation:

> A TxDefaultProfile charging profile on EVSE #0 is “owned by” EVSE #0, but has effect on all EVSEs.

This explicitly means that the TxDefaultProfile is **not copied** to all other EVSEs, because that would create problems with duplicate charging profile IDs leading to undefined results with ClearChargingProfile and GetChargingProfile.

## 13.5.1. Page 234 - Updated requirements for TxDefaultProfile on EVSE #0

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | K01.FR.14 | When a SetChargingProfileRequest with a TxDefaultProfile and evseId = 0 is received. | The Charging Station SHALL apply this profile to all EVSEs. | |
| New text | K01.FR.14 | When a SetChargingProfileRequest with a TxDefaultProfile and evseId = 0 is received AND<br>No other TxDefaultProfile with the same stackLevel is installed on any specific EVSE. | The Charging Station SHALL apply, **but not copy,** this profile to all EVSEs. | A TxDefaultProfile charging profile on EVSE #0 is “owned by” EVSE #0, but has effect on all EVSEs. |
| Old text | K01.FR.15 | When a SetChargingProfileRequest with a TxDefaultProfile and evseId > 0 is received. | The Charging Station SHALL only apply this profile to the specified EVSE. | |
| New text | K01.FR.15 | When a SetChargingProfileRequest with a TxDefaultProfile and evseId > 0 is received AND<br>No TxDefaultProfile with the same stackLevel is installed on EVSE #0. | The Charging Station SHALL only apply this profile to the specified EVSE. | |

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.52 | When a SetChargingProfileRequest with a TxDefaultProfile and *evseId* = 0 is received AND<br>A TxDefaultProfile with the same *stackLevel* is installed on a specific EVSE and its *chargingProfile.id* does NOT equal the received *chargingProfile.id* | The Charging Station SHALL respond with a SetChargingProfileResponse with status `Rejected` and optionally with *reasonCode* = `DuplicateProfile`. | |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.53 | When a SetChargingProfileRequest with a TxDefaultProfile and *evseId* > 0 is received AND A TxDefaultProfile with the same *stackLevel* is installed on EVSE #0 and its *chargingProfile.id* does NOT equal the received *chargingProfile.id* | The Charging Station SHALL respond with a SetChargingProfileResponse with status `Rejected` and optionally with *reasonCode* = `DuplicateProfile`. | |

## 13.5.2. Page 251 - Updated requirements for GetChargingProfileRequest

The updated note for K09.FR.05 in Errata v1, Page 251 - (v1) Requirements K09 GetChargingProfiles [406], is not correct in light of the improved description on how to deal with TxDefaultProfile charging profiles for EVSE #0.

*Changed requirement*

| | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old text (Errata v1) | K09.FR.05 | If *evseId* is set to 0 in GetChargingProfilesRequest | The Charging Station SHALL only report charging profiles installed on the Charging Station itself (the grid connection) that match all fields in *chargingProfile*. | For *evseId* = 0, you should only have a `ChargingStation MaxProfile` purpose (or `ChargingStation ExternalConstraints`), because a `TxProfile` is not allowed on *evseId* = 0 and a `TxDefaultProfile` for *evseId* = 0 is not applied to #0 but to all individual EVSEs (see K01.FR.14). |
| New text | K09.FR.05 | If *evseId* is set to 0 in GetChargingProfilesRequest | The Charging Station SHALL only report charging profiles installed on the Charging Station itself (the grid connection) that match all fields in *chargingProfile*. | **EVSE #0 can have a `ChargingStation MaxProfile`, `ChargingStation ExternalConstraints` or a `TxDefaultProfile`.**<br>**Note, that a `TxDefaultProfile` is not applied to EVSE #0 but to all individual EVSEs (see K01.FR.14).** |

# 13.6. Page 236 - (v2) Use case K01: *numberPhases*, *phaseToUse* does not match EVSE capabilities [577]

If the CSMS sets a charging profile with specific phase information on an EVSE that cannot support that, it is currently unclear how the Charging Station should react. Furthermore, it is stated that *numberPhases* must match the EVSE capabilities of an AC charger, but does not mention the case for DC chargers, nor the *phaseToUse* field.

Requirement K01.FR.18 does not cover the full scope and must be split into multiple requirements.

*Deleted requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.18 | | For AC charging, the CSMS SHALL NOT set numberPhases different from the EVSE capabilities in a SetChargingProfileRequest, otherwise the Charging Station SHOULD respond with *Rejected*. | When a ChargingProfile asks for 3 phases and the Charging Station is able to charge 3 phases, it is not guaranteed that the EV and/or cable are able to charge 3 phases. Based on *MeterValues* the CSMS can determine the phases used. |

The following requirements are added to cover everything related to the number of phases. Also requirements have been added for the current per phase calculation. These were inexplicitly defined by the enum value descriptions of chargingRateUnitEnumType.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.43 | When a SetChargingProfileRequest with a value for *numberPhases* is received<br>AND<br>the EVSE is of type AC AND the Charging Station cannot ensure that no more than the received numberPhases will be used | The Charging Station SHALL respond with status = *Rejected* | Note that even when for example the ChargingProfile defines 3 phases and the Charging Station is able to charge with 3 phases, it is not guaranteed that the EV or cable are able to charge with 3 phases.<br>Based on received MeterValues the CSMS can determine the used number of phases.<br>Please refer to requirement K01.FR.50 and K01.FR.51, for correctly calculating the limits per phase. |
| K01.FR.44 | When a SetChargingProfileRequest with a value for *numberPhases* or *phaseToUse* is received<br>AND<br>the EVSE is of type DC | The Charging Station MAY respond with status = *Accepted*, instead of *Rejected* and ignore the provided values for *numberPhases* and *phaseToUse*. | |
| K01.FR.45 | When a SetChargingProfileRequest with a value for *numberPhases* is received<br>AND<br>the EVSE is of type AC AND the received *numberPhases* is NOT supported by the Charging Station and higher than the *numberPhases* that are supported by the Charging Station | The Charging Station MAY respond with status = *Accepted*, instead of *Rejected* and impose the limits to a lower *numberPhases* | Please refer to requirement K01.FR.50 and K01.FR.51, for correctly calculating the limits per phase. |
| K01.FR.46 | When a SetChargingProfileRequest with *numberPhases = 1* and a value for *phaseToUse* is received AND<br>the EVSE is of type AC AND the EVSE is capable of switching the phase connected to the EV, which is indicated by ACPhaseSwitchingSupported defined as *true* OR<br>the EVSE is already going to use the received *phaseToUse* | The Charging Station SHALL use the phase indicated by the received *phaseToUse* to connect to the EV. | |
| K01.FR.47 | When a SetChargingProfileRequest with *numberPhases = 1* and *phaseToUse* is omitted is received AND<br>the EVSE is of type AC | The Charging Station SHALL select the phase on its own. | |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.48 | When a SetChargingProfileRequest with a value for *phaseToUse* is received AND the EVSE is NOT capable of switching the phase connected to the EV, which is indicated by ACPhaseSwitchingSupported not being implemented or defined as *false* AND the EVSE is NOT going to use the received *phaseToUse* | The Charging Station SHALL respond with status = *Rejected*. | |
| K01.FR.49 | When a SetChargingProfileRequest without a value for *numberPhases* is received AND the EVSE is of type AC | The Charging Station SHALL assume numberPhases = 3 as a default value. | |
| K01.FR.50 | When a SetChargingProfileRequest with a chargingRateUnit = *W* is received AND The ChargingSchedule is used for AC charging | The Charging Station SHOULD calculate the phase current limit via: Current per phase = Power / (Line Voltage * Number of Phases). | The "Line Voltage" used in the calculation is not the measured voltage, but the set voltage for the area (for example, 230 or 110 V). The "Number of Phases" is the numberPhases from the ChargingSchedulePeriod.<br>It is usually more convenient to use chargingRateUnit = *A* for AC charging. |
| K01.FR.51 | When a SetChargingProfileRequest with a chargingRateUnit = *A* is received | The Charging Station SHALL use the provided limits, to limit the amount of Ampere per phase, not the sum of all phases. | |

### 13.6.1. Page 374 - (v2) Updated description of *numberPhases*

The description of *numberPhases* mentions that the default value is 3. This is not correct when used for a DC charging station.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | numberPhases | integer | 0..1 | Optional. The number of phases that can be used for charging. If a number of phases is needed, numberPhases=3 will be assumed unless another number is given. |
| New text | numberPhases | integer | 0..1 | Optional. The number of phases that can be used for charging.<br>**For a DC EVSE this field should be omitted.**<br>**For an AC EVSE a default value of numberPhases = 3 will be assumed if the field is absent.** |

## 13.7. Page 236 - (v1) Use case K01: Missing requirement for *startSchedule* [510]

In the description of ChargingProfileKindType it is mentioned that a *startSchedule* is needed in a ChargingSchedule of a ChargingProfile with *chargingProfileKind* = `Absolute`, and that it should be absent for a `Relative` profile. A requirement for this is missing in K01.

In order to make this explicit, the following requirements are added:

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.40 | When *chargingProfileKind* of a ChargingProfile is `Absolute` or `Recurring` | A value for *startSchedule* SHALL exist in the ChargingSchedule of the ChargingProfile. | This determines start date-time of the schedule and of the recurrency sequence. |
| K01.FR.41 | When *chargingProfileKind* of a ChargingProfile is `Relative` | The field *startSchedule* SHALL be absent in the ChargingSchedule of the ChargingProfile. | A relative profile starts from when the profile is activated. |

## 13.8. Page 235 - (v2) Definition of start of a relative charging profile [638]

The point in time when a relative charging profile (chargingProfileKind = Relative) starts is not well defined. It can be interpreted as relative to start of transaction, start of charging or receiving of the SetChargingProfileRequest. The following requirement is added to remove this ambiguity.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.42 | K01.FR.41 | It is RECOMMENDED to make the ChargingSchedulePeriods relative to the moment the Charging Station is ready to deliver energy. i.e. when the EV driver is authorized and the EV is connected. | This is the point in a transaction where the charging station is ready to deliver energy. If PowerPathClosed is a TxStartPoint, then this will concur with the start of a transaction.<br>In the next OCPP version, this will become a more strict requirement. |

### 13.8.1. Page 373 - ChargingScheduleType

The description of the field *startSchedule* has been updated.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **startSchedule** | dateTime | 0..1 | Optional. Starting point of an absolute schedule. If absent the schedule will be relative to start of charging. |
| New text | **startSchedule** | dateTime | 0..1 | Optional. Starting point of an absolute or recurring schedule. |

### 13.8.2. Page 395 - ChargingProfileKindEnumType

The description of the value `Relative` has been updated.

| | Value | Description |
|---|---|---|
| Old text | **Relative** | Charging schedule periods start when ChargingProfile is activated. In most cases this will be at start of the power delivery. When a ChargingProfile is received for a transaction in progress, then it should activate immediately. No value for *startSchedule* should be supplied. |
| New text | **Relative** | **Charging schedule periods should start when the EVSE is ready to deliver energy. i.e. when the EV driver is authorized and the EV is connected. When a ChargingProfile is received for a transaction that is already charging, then the charging schedule periods should remain relative to the PowerPathClosed moment.**<br>No value for *startSchedule* should be supplied. |

## 13.9. Page 238 - (v2) Use case K02: Incorrect note of K02.FR.03 [597]

The note states that it is possible to replace a profile by *stackLevel* and *chargingProfilePurpose* combination. However, this is not allowed.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | K02.FR.03 | | In order to ensure that an updated ChargingProfile applies only to the current transaction, the CSMS SHALL set the chargingProfilePurpose of the ChargingProfile to *TxProfile*. | An updated charging profile can be sent by the CSMS by sending a ChargingProfile with the same chargingProfileId, or the same combination of stackLevel / ChargingProfilePurpose. |
| New text | K02.FR.03 | | In order to ensure that an updated ChargingProfile applies only to the current transaction, the CSMS SHALL set the chargingProfilePurpose of the ChargingProfile to *TxProfile*. | An updated charging profile can be sent by the CSMS by sending a ChargingProfile with the same chargingProfileId ~~, or the same combination of stackLevel / ChargingProfilePurpose.~~ |

]

## 13.10. Page 245 - (v1) Use case K05: Remote Start Transaction with Charging Profile [354]

When a ChargingProfile with a RequestStartTransactionRequest is provided, then it is not possible to provide a *transactionId* for the ChargingProfile, even though that would be required when a TxProfile is set via SetChargingProfileRequest.

Add the following text to the remark in the use case:

| | | |
|---|---|---|
| **8** | **Remark(s)** | [...]<br>When a ChargingProfile with purpose TxProfile is provided as part of a RequestStartTransactionRequest, then a *transactionId* cannot be provided in the ChargingProfile, because it is not known at the time. |

### 13.10.1. Page 373 - (v1) section 2.10 ChargingProfile

The description of *transactionId* needs a slight change to clarify that it is only mandatory in the context of a SetChargingProfileRequest and not when used in a RequestStartTransactionRequest.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **transactionId** | identifierString[0..36] | 0..1 | Optional. SHALL only be included if ChargingProfilePurpose is set to TxProfile. The transactionId is used to match the profile to a specific transaction. |
| New text | **transactionId** | identifierString[0..36] | 0..1 | Optional. SHALL only be included when ChargingProfilePurpose is set to TxProfile **in a SetChargingProfileRequest**. The transactionId is used to match the profile to a specific transaction. |

## 13.11. Page 245 - (v2) Use case K05: added requirements from F01/F02 [558]

Use cases F01 and F02 state that a charging station may ignore a charging profile from a RequestStartTransactionRequest if it does not support smart charging. Similarly, it must be possible for a smart charging-capable charging station to reject an invalid charging profile.

There requirements are missing in K05.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K05.FR.04 | If a Charging Station **without** support for Smart Charging receives a RequestStartTransactionRequest with a ChargingProfile. | The Charging Station SHALL ignore the specified ChargingProfile. | The device model variable SmartChargingCtrlr.Enabled tells CSMS whether smart charging is supported. |
| K05.FR.05 | If a Charging Station **with** support for Smart Charging receives a RequestStartTransactionRequest with an invalid ChargingProfile. | The Charging Station SHALL respond with RequestStartTransactionResponse with *status* = `Rejected` and optionally with *reasonCode* = "InvalidProfile" or "InvalidSchedule". | The device model variable SmartChargingCtrlr.Enabled tells CSMS whether smart charging is supported. |

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | K05.FR.03 | K05.FR.01 | The Charging Station SHALL use the given profile to calculate its composite schedule. |
| New text | K05.FR.03 | K05.FR.01 **AND NOT K05.FR.04** | The Charging Station SHALL use the given profile to calculate its composite schedule. |

# 13.12. Page 250 - (v1) Unclear precondition of K08.FR.06 [462]

Requirement K08.FR.06 specifies what should be calculated as a composite charging profile for an EVSE, when there is currently no transaction active. This is only needed when a request for a composite schedule is received, but that was not mentioned in the precondition.

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | K08.FR.06 | When there is no transaction active on an EVSE | The Charging Station SHALL calculate the CompositeSchedule as if there is a transaction ongoing on the EVSE that is using the TxDefaultProfile (if this profile purpose is set) |
| New text | K08.FR.06 | **K08.FR.02 AND**<br>When there is no transaction active on an EVSE | The Charging Station SHALL calculate the CompositeSchedule as if there is a transaction ongoing on the EVSE that is using the TxDefaultProfile (if this profile purpose is set) |

# 13.13. Page 251 - (v1) Requirements K09 GetChargingProfiles [406]

The requirement K09.FR.03 does not make clear that the fields *evseId* and *chargingProfile* are filters of equal importance and all present fields needs to apply for a charging profile to be reported.

There is an inconsistency between K09.FR.03 and requirements K09.FR.04-06, that is resolved by adding the fact that the *chargingProfile* filter also needs to be applied.

Requirement K09.FR.03 mentions *evseId* as one of the filters, but that is redundant, because the *evseId* field is already covered by requirements K09.FR.04-06. K03.FR.03 is rephrased to only focus on the *chargingProfile* criteria field.

*Changed requirements*

| | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old text | K09.FR.03 | | The CSMS SHALL either specify a (list of) chargingProfileId(s) OR include one or more of the fields *stackLevel*, *evseId*, *chargingLimitSource* and *chargingProfilePurpose* in the GetChargingProfilesRequest (that are matched as a logical AND) to specify which Charging Profiles need to be reported. | |

| | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| New text | K09.FR.03 | | The CSMS SHALL specify in *chargingProfile* criteria in GetChargingProfilesRequest either:<br>- a (list of) *chargingProfileId(s)* OR<br>- one or more of the fields *stackLevel*, *chargingLimitSource*, *chargingProfilePurpose*. | These fields are filter values of equal importance, but because a *chargingProfileId* uniquely identifies a charging profile, the other fields are not needed if *chargingProfileIds* are used. |
| Old text | K09.FR.04 | If *evseId* is set to a value greater than 0 in the GetChargingProfilesRequest | The Charging Station SHALL report the installed charging profiles for the specified EVSE | |
| New text | K09.FR.04 | If *evseId* is set to a value greater than 0 in the GetChargingProfilesRequest | The Charging Station SHALL report the installed charging profiles for the specified EVSE **that match all fields in *chargingProfile*.** | |
| Old text | K09.FR.05 | If *evseId* is set to 0 in GetChargingProfilesRequest | The Charging Station SHALL only report charging profiles installed on the Charging Station itself (the grid connection) | |
| New text | K09.FR.05 | If *evseId* is set to 0 in GetChargingProfilesRequest | The Charging Station SHALL only report charging profiles installed on the Charging Station itself (the grid connection) **that match all fields in *chargingProfile*.** | For *evseId* = 0, you should only have a `ChargingStationMaxProfile` purpose (or `ChargingStationExternalConstraints`), because a `TxProfile` is not allowed on *evseId* = 0 and a `TxDefaultProfile` for *evseId* = 0 is not applied to #0 but to all individual EVSEs (see K01.FR.14). |
| Old text | K09.FR.06 | If *evseId* is NOT set in the GetChargingProfilesRequest | The Charging Station SHALL report all installed charging profiles. | |
| New text | K09.FR.06 | If *evseId* is NOT set in the GetChargingProfilesRequest | The Charging Station SHALL report all installed charging profiles **that match all fields in *chargingProfile*.** | |

**NOTE** K06.FR.05 has been revised in Errata v2 at Page 234 - (v2) Use case K01: Improved explanation of TxDefaultProfile on evseId #0 [593]

## 13.13.1. Page 346 - (v1) GetChargingProfileRequest

The description of field *evseId* does not mention that *chargingProfile* criteria still need to match.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old description | evseId | integer | 0..1 | Optional. For which EVSE installed charging profiles SHALL be reported. If 0, only charging profiles installed on the Charging Station itself (the grid connection) SHALL be reported. If omitted, all installed charging profiles SHALL be reported. |
| New description | evseId | integer | 0..1 | Optional. For which EVSE installed charging profiles SHALL be reported. If 0, only charging profiles installed on the Charging Station itself (the grid connection) SHALL be reported. If omitted, all installed charging profiles SHALL be reported. **Reported charging profiles SHALL match the criteria in field *chargingProfile*.** |

## 13.14. Page 252 - (v1) Use case K10 requirements are incomplete [502]

A CSMS is not allowed to clear a charging profile with *chargingProfilePurpose* = `ChargingStationExternalConstraints`. This is intended by requirement K10.FR.06, but this does not cover the case where a charging profile for external constraints is cleared by providing the *chargingProfileid* or when it matches the *chargingProfileCriteria*.

This is made explicit in the following two requirements:

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | K10.FR.03 | Upon receipt of a ClearChargingProfileRequest with a specified id. | The Charging Station SHALL clear the Charging Profile with the matching id and respond with a ClearChargingProfileResponse message. | |
| New text | K10.FR.03 | Upon receipt of a ClearChargingProfileRequest with a specified *chargingProfileId* **AND the chargingProfilePurpose of the referenced ChargingProfile is NOT `ChargingStationExternalConstraints`** | The Charging Station SHALL clear the Charging Profile with the matching id and respond with a ClearChargingProfileResponse message with *status* = `Accepted`. | |
| Old text | K10.FR.04 | NOT K10.FR.03 AND<br>Upon receipt of a ClearChargingProfileRequest, with optional values for evseId, chargingProfilePurpose, stackLevel | The Charging Station SHALL clear the Charging Profiles that match (as logical AND) the values in the request and respond with a ClearChargingProfileResponse message. | |
| New text | K10.FR.04 | NOT K10.FR.03 AND<br>**NOT K10.FR.08 AND**<br>Upon receipt of a ClearChargingProfileRequest, with optional values for *evseId*, *chargingProfilePurpose*, *stackLevel* | The Charging Station SHALL clear the ChargingProfile(s) that match (as logical AND) the values in the request, **except those for that have ChargingProfile = `ChargingStationExternalConstraints`** and respond with a ClearChargingProfileResponse message with *status* = `Accepted`. | |

When the only charging profiles are of purpose `ChargingStationExternalConstraints`, then the response *status* is `Unknown`, as if no charging profiles were found.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K10.FR.08 | Upon receipt of a ClearChargingProfileRequest, with optional values for *evseId*, *chargingProfilePurpose*, *stackLevel* AND<br>the matched ChargingProfile(s) all have ChargingProfile = `ChargingStationExternalConstraints` | The Charging Station SHALL respond with a ClearChargingProfileResponse message with *status* = `Unknown`. | Charging profiles for external constraints are disregarded by ClearChargingProfile message. |
| K10.FR.09 | Upon receipt of a ClearChargingProfileRequest with a specified *chargingProfileId* AND<br>the chargingProfilePurpose of the referenced ChargingProfile = `ChargingStationExternalConstraints` | The Charging Station SHALL respond with a ClearChargingProfileResponse message with *status* = `Unknown`. | Charging profiles for external constraints are disregarded by ClearChargingProfile message. |

## 13.15. Page 253 - (v1) K01.FR.34 refers to ChargingSchedulePeriodType, but should be ChargingScheduleType [363]

The requirement K01.FR.34 means to say that only for ISO15118 there can be up to three ChargingSchedules. However, it now states that there can be only one ChargingSchedulePeriodType. That is wrong.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | K01.FR.34 | The CSMS has not received a NotifyEVChargingNeeds Request for the current transaction, i.e. charging session is not using ISO 15118 | The SetChargingProfileRequest SHALL contain at most one ChargingSchedulePeriodType and no SalesTariffType elements. | See use cases K15-K17 for ISO 15118 smart charging. |
| New text | K01.FR.34 | The CSMS has not received a NotifyEVChargingNeeds Request for the current transaction, i.e. charging session is not using ISO 15118 | **The ChargingProfile in** the SetChargingProfileRequest SHALL contain **only one ChargingScheduleType.** | See use cases K15-K17 for ISO 15118 smart charging. |

# 13.16. Page 254 - (v1) Charging profile id’s for external constraints profiles [365]

Use cases K11 and K12 explain (in K11.FR.06 an K12.FR.05) how an externally received constraint can be reported to CSMS in a ReportChargingProfilesRequest message as a charging profile with purpose `ChargingStationExternalConstraints`. Such a charging profile has an *id*, but since this *id* is not assigned by CSMS, there always is a chance, that this *id* clashes with an *id* that CSMS has already assigned to a profile or may use for a future profile.

Is is recommended to use negative integer values as *id’s* for charging profiles that are created by the Charging Station to report external constraints. This minimizes the chance of CSMS using the same value for a new charging profile, because these *id’s* are likely to be positive numbers.

The following recommendation is added to below requirements:

*Changed requirements*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K11.FR.06 | When an external charging limit/schedule is received | The Charging Station SHALL use purpose ChargingStationExternalConstraints when reporting about this limit (e.g. in a ReportChargingProfilesRequest). | **It is RECOMMENDED to use negative values for the *id* of a `ChargingStationExternalConstraints` profile, to minimize the risk of a clash with an *id* that CSMS might use for a (future) charging profile.** |

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K12.FR.05 | When an external charging limit/schedule is received | The Charging Station SHALL use purpose ChargingStationExternalConstraints when reporting about this limit (e.g. in a ReportChargingProfilesRequest). | **It is RECOMMENDED to use negative values for the *id* of a `ChargingStationExternalConstraints` profile, to minimize the risk of a clash with an *id* that CSMS might use for a (future) charging profile.** |

In the next OCPP release a standardized device model variable will be introduced to reserve a range of charging profile *id’s* that shall only be used for external constraints profiles. If CSMS sees that this variable is reported in the device model report of the charging station, then it will refrain from using *id’s* in that value range.

# 13.17. Page 259 - (v1) section K15: Recommendation for NotifyEVChargingScheduleRequest [349]

When smart charging during an ISO 15118 session, the CSMS may provide up to three schedules to the EV in a SetChargingProfileRequest. The EV selects one of these schedules and returns the ID of the selected schedule to the Charging Station. However, if the Charging Station does not send the message NotifyEVChargingScheduleRequest to CSMS, because it is optional, then CSMS will not know according to which charging schedule the EV will be charging.

This problem does not occur if CSMS provides only one schedule, because then it knows the schedule that will be used. This is added as a recommendation. Similarly, we recommend the Charging Station to always send a NotifyEVChargingSchedule to CSMS. If the EV does not send its own charging profile in the PowerDeliveryReq, then the Charging Station returns the schedule that the EV selected to use and which is designated by the SAScheduleTupleID field in the PowerDeliveryReq message.

This requires the following changes to the specification:

## 13.17.1. Usecase K15

### At Table 188.

Requirements are added for above-mentioned recommendations.

**WARNING** This has been superseded by erratum *Page 259 - Improved requirement definitions regarding NotifyEVChargingScheduleRequest [349]*.

*New requirements*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K15.FR.18 | K15.FR.03 OR K15.FR.05 | CSMS IS RECOMMENDED to use only one *chargingSchedule* in a SetChargingProfileRequest. | This ensures that there is no doubt about which schedule the EV will follow, even when no NotifyEVChargingScheduleRequest is received. |
| K15.FR.19 | K15.FR.07<br>AND<br>EV does not return a charging profile | Charging Station IS RECOMMENDED to return an EV charging profile in a NotifyEVChargingScheduleRequest message to CSMS that matches the schedule that was selected by the EV | In ISO 15118-2 the EV charging profile and the selected schedule are returned as *ChargingProfile* and *SAScheduleTupleId* in PowerDeliveryReq. |

## 13.17.2. Usecase K16

### At table of requirements

**WARNING** This has been superseded by erratum *Page 259 - Improved requirement definitions regarding NotifyEVChargingScheduleRequest [349]*.

*New requirement*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K16.FR.13 | EV does not return a charging profile | Charging Station IS RECOMMENDED to return an EV charging profile in a NotifyEVChargingScheduleRequest message to CSMS that matches the schedule that was selected by the EV | In ISO 15118-2 the EV charging profile and the selected schedule are returned as *ChargingProfile* and *SAScheduleTupleId* in PowerDeliveryReq. |

This following requirement mentioned the wrong message in the precondition by mistake:

*Changed requirement*

| Version | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | K16.FR.12 | K16.FR.09<br>AND<br>Charging Station sends a NotifyEVChargingScheduleRequest | The CSMS SHALL send a SetChargingProfileRequest. | [..] |
| New | K16.FR.12 | K16.FR.09<br>AND<br>Charging Station sends a **NotifyEVChargingNeedsRequest** | The CSMS SHALL send a SetChargingProfileRequest. | [..] |

### 13.17.3. Usecase K17

#### At Table 191.

**WARNING** This has been superseded by erratum *Page 259 - Improved requirement definitions regarding NotifyEVChargingScheduleRequest [349]*.

*New requirement*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K17.FR.16 | K17.FR.07<br>EV does not return a charging profile | Charging Station IS RECOMMENDED to return an EV charging profile in a NotifyEVChargingScheduleRequest message to CSMS that matches the schedule that was selected by the EV | In ISO 15118-2 the EV charging profile and the selected schedule are returned as *ChargingProfile* and *SAScheduleTupleId* in PowerDeliveryReq. |

## 13.18. Page 259 - Improved requirement definitions regarding NotifyEVChargingScheduleRequest [349]

The wording of the requirement definitions regarding NotifyEVChargingScheduleRequest needs to be improved.

*Changed requirements*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K15.FR.19 | K15.FR.07<br>AND<br>EV does not return a charging profile | Charging Station IS RECOMMENDED to return an EV charging profile **as a chargingSchedule** in a NotifyEVChargingScheduleRequest message to CSMS that matches the schedule that was selected by the EV **(i.e. chargingSchedule.id = SAScheduleTupleId)** | In ISO 15118-2 the EV charging profile and the selected schedule are returned as *ChargingProfile* and *SAScheduleTupleId* in PowerDeliveryReq. |
| K16.FR.13 | EV does not return a charging profile | Charging Station IS RECOMMENDED to return an EV charging profile **as a chargingSchedule** in a NotifyEVChargingScheduleRequest message to CSMS that matches the schedule that was selected by the EV **(i.e. chargingSchedule.id = SAScheduleTupleId)** | In ISO 15118-2 the EV charging profile and the selected schedule are returned as *ChargingProfile* and *SAScheduleTupleId* in PowerDeliveryReq. |
| K17.FR.16 | K17.FR.07<br>EV does not return a charging profile | Charging Station IS RECOMMENDED to return an EV charging profile **as a chargingSchedule** in a NotifyEVChargingScheduleRequest message to CSMS that matches the schedule that was selected by the EV **(i.e. chargingSchedule.id = SAScheduleTupleId)** | In ISO 15118-2 the EV charging profile and the selected schedule are returned as *ChargingProfile* and *SAScheduleTupleId* in PowerDeliveryReq. |

## 13.19. Page 260 - (v1) Use case K15: Improved error handling description

The error handling description for use case K15 has been improved, because it was not clear how this related to NotifyEVChargingNeeds and SetChargingProfile messages.

| | | | |
|---|---|---|---|
| Old | **9** | **Error handling** | A hard requirement from ISO 15118 is that the response should be sent within the timeout (thus OCPP messaging should have an even lower timeout). If the timeout has been reached, the EV will stop and does not do a retry according to ISO 15118. Therefore, if the SalesTariff cannot be handled fast enough, the Charging Station should start charging by delivering the mandatory PMaxSchedule parameter and in parallel it should handle the optional SalesTariff and start a ISO 15118 renegotiation according to K17 - Renegotiating a Charging Schedule. |

| | | | |
|---|---|---|---|
| New | **9** | **Error handling** | The Charging Station needs to use the information from the SetChargingProfileRequest message to create the response to the ISO 15118 ChargeParameterDiscoveryReq towards the EV. This message has a timeout of 60 seconds, which means the SetChargingProfileRequest has to be sent well within 60 seconds after receiving the NotifyEVChargingNeedsRequest. If the Charging Station does not receive the SetChargingProfileRequest in time or when the NotifyEVChargingNeedsResponse has *status* = `Processing`, then the Charging Station will return a schedule in ChargeParameterDiscoverRes that matches the capabilities of the EVSE. When CSMS sends the SetChargingProfileRequest at a later time, then this will trigger a renegotiation according to use case K16 - Renegotiation initiated by CSMS. |

### 13.19.1. Page 260 - (v1) Added note to K15.FR.05

A note has been added to clarify behavior when *status* is `Processing`.

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K15.FR.05 | K15.FR.02 | If the CSMS is able to provide a charging schedule; but needs processing time, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to 'Processing'. | **The Charging Station does not have to wait for the SetChargingProfileRequest. CSMS will send it later and trigger a renegotiation as per use case K16.** |

## 13.20. Page 260 - (v1) Note to requirement K15.FR.07 about composite schedule

A note has been added to clarify that the SASchedule that Charging Station sends to the EV is the **composite** schedule that applies to the EVSE. This may be a combination of the `TxProfile` with a `ChargingStationMaxProfile` and a `ChargingStationExternalConstraints` if they are present.

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K15.FR.07 | K15.FR.03 or K15.FR.05 | The CSMS SHALL send a SetChargingProfileRequest with *chargingProfilePurpose* = `TxProfile` and a *transactionId* and at most three *chargingSchedule* and optional *salesTariff* elements, that each contain no more periods than specified by *maxScheduleTuples* in NotifyEVChargingNeedsRequest and by device model variable `SmartChargingCtrlr.PeriodsPerSchedule`. | **The Charging Station will calculate the composite schedule(s) for the EVSE (taking into account a `ChargingStationMaxProfile` or `ChargingStationExternalConstraints` if present) and will convert that to the SAScheduleList format for ISO 15118.** |

## 13.21. Page 261 - (v1) Requirement K15.FR.17 has incomplete precondition [366]

Requirement K15.FR.17 refers to the situation during an ISO 15118 charging session in which CSMS sends a charging profile before it has received the NotifyEVChargingNeeds. It is important that CSMS sends a SetChargingProfileRequest after receiving NotifyEVChargingNeeds, because the Charging Station might wait for this information to be sent to the EV.

The precondition has been changed to make clear that this refers to the situation immediately after a transaction has started. The requirement definition now states that Charging Station SHOULD reject the charging profile to make clear to CSMS that the message was sent too early.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | K15.FR.17 | When Charging Station receives a SetChargingProfileRequest before EV has sent charging needs | The Charging Station SHALL respond with SetChargingProfileResponse with *status* = `Accepted` and ignore the information. | CSMS sent profile too early and will send a profile again in response to NotifyEVChargingNeedsRequest. |

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New text | K15.FR.17 | When Charging Station receives a SetChargingProfileRequest immediately after the transaction has started and before it has sent the NotifyEVChargingNeedsRequest to CSMS | The Charging Station SHOULD respond with SetChargingProfileResponse with *status* = `Rejected` and a *statusInfo* with *reasonCode*= `InvalidMessageSequence`. | CSMS sent profile too early. It does not harm if CS accepts the charging profile instead of rejecting it, as long as it sends a charging profile again when it receives the NotifyEVChargingNeedsRequest. |

# 14. Use case L Firmware Management

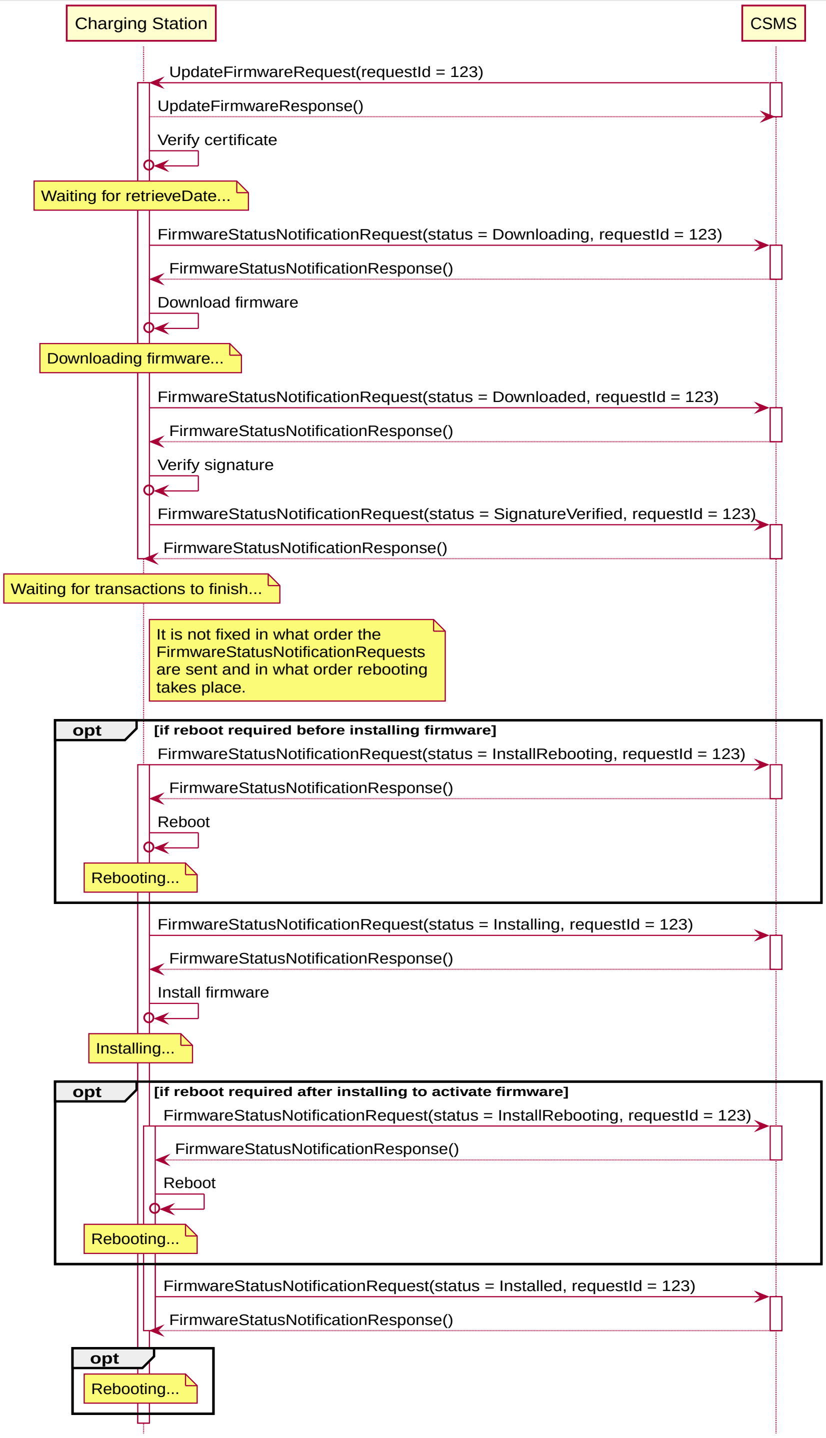
## 14.1. Page 271, 277 - (v2) Sequence diagrams allow reboot before status=Installed [639]

The sequence diagram has a note that the order of rebooting is not fixed. However, the sequence diagrams of Figure 116 and Figure 118 show an optional block for "Rebooting…" at the end. This may be confusing, because usually a reboot needs to be performed to activate the newly installed firmware, and normally the Charging Station will send the FirmwareStatusNotificationRequest(*status* = Installed) after the firmware has been successfully activated.

### 14.1.1. Page 271, Use case L01

*Change to sequence diagram Figure 116*

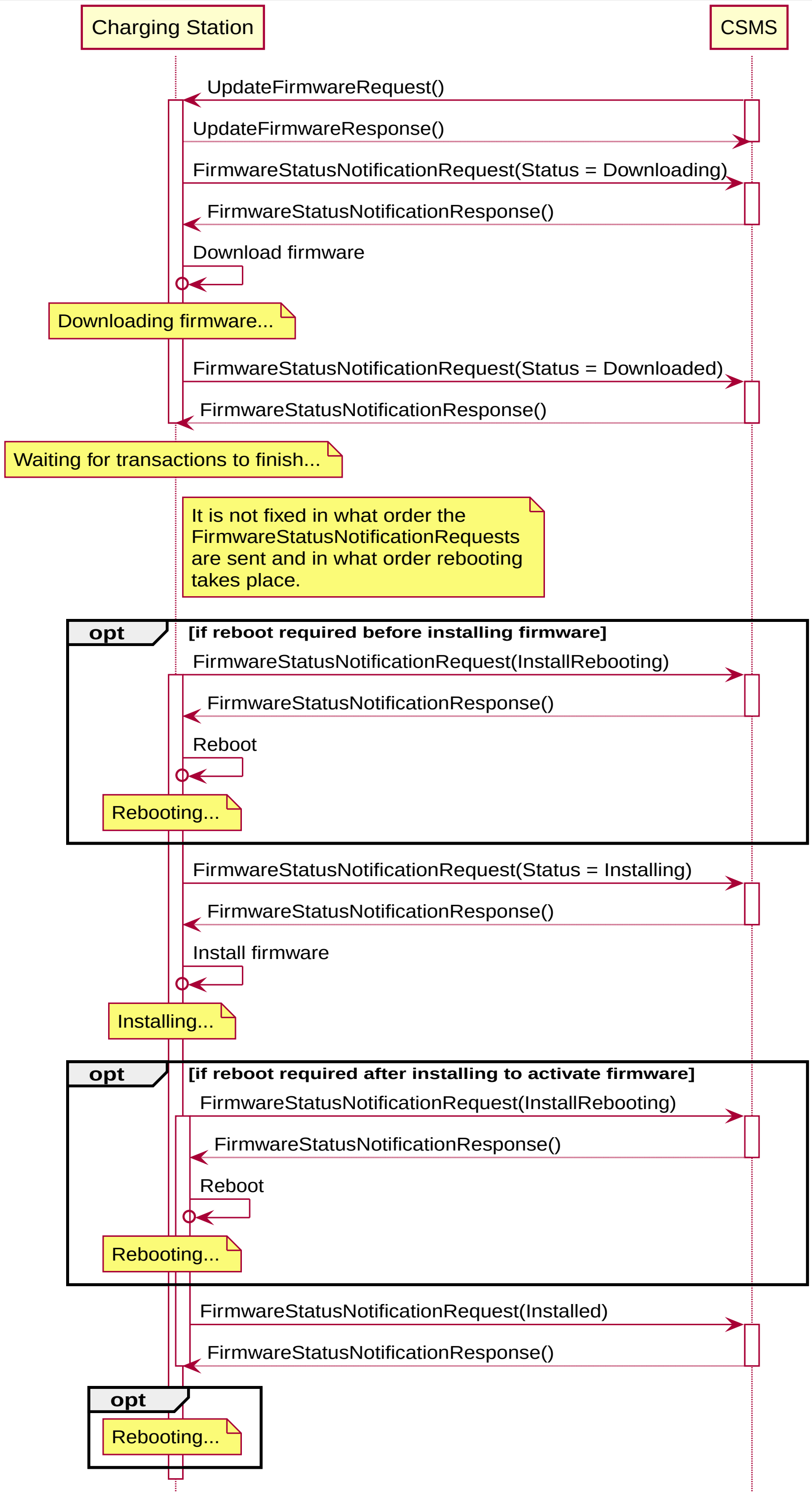
- Add an optional block "Rebooting…" above the message FirmwareStatusNotificationRequest(*status* = Installed).


Charging Station
CSMS
UpdateFirmwareRequest(requestId = 123)
UpdateFirmwareResponse()
Verify certificate
Waiting for retrieveDate...
FirmwareStatusNotificationRequest(status = Downloading, requestId = 123)
FirmwareStatusNotificationResponse()
Download firmware
Downloading firmware...
FirmwareStatusNotificationRequest(status = Downloaded, requestId = 123)
FirmwareStatusNotificationResponse()
Verify signature
FirmwareStatusNotificationRequest(status = SignatureVerified, requestId = 123)
FirmwareStatusNotificationResponse()
Waiting for transactions to finish...
It is not fixed in what order the FirmwareStatusNotificationRequests are sent and in what order rebooting takes place.
opt
[if reboot required before installing firmware]
FirmwareStatusNotificationRequest(status = InstallRebooting, requestId = 123)
FirmwareStatusNotificationResponse()
Reboot
Rebooting...
FirmwareStatusNotificationRequest(status = Installing, requestId = 123)
FirmwareStatusNotificationResponse()
Install firmware
Installing...
opt
[if reboot required after installing to activate firmware]
FirmwareStatusNotificationRequest(status = InstallRebooting, requestId = 123)
FirmwareStatusNotificationResponse()
Reboot
Rebooting...
FirmwareStatusNotificationRequest(status = Installed, requestId = 123)
FirmwareStatusNotificationResponse()
opt
Rebooting...

## 14.1.2. Page 277, Use case L02

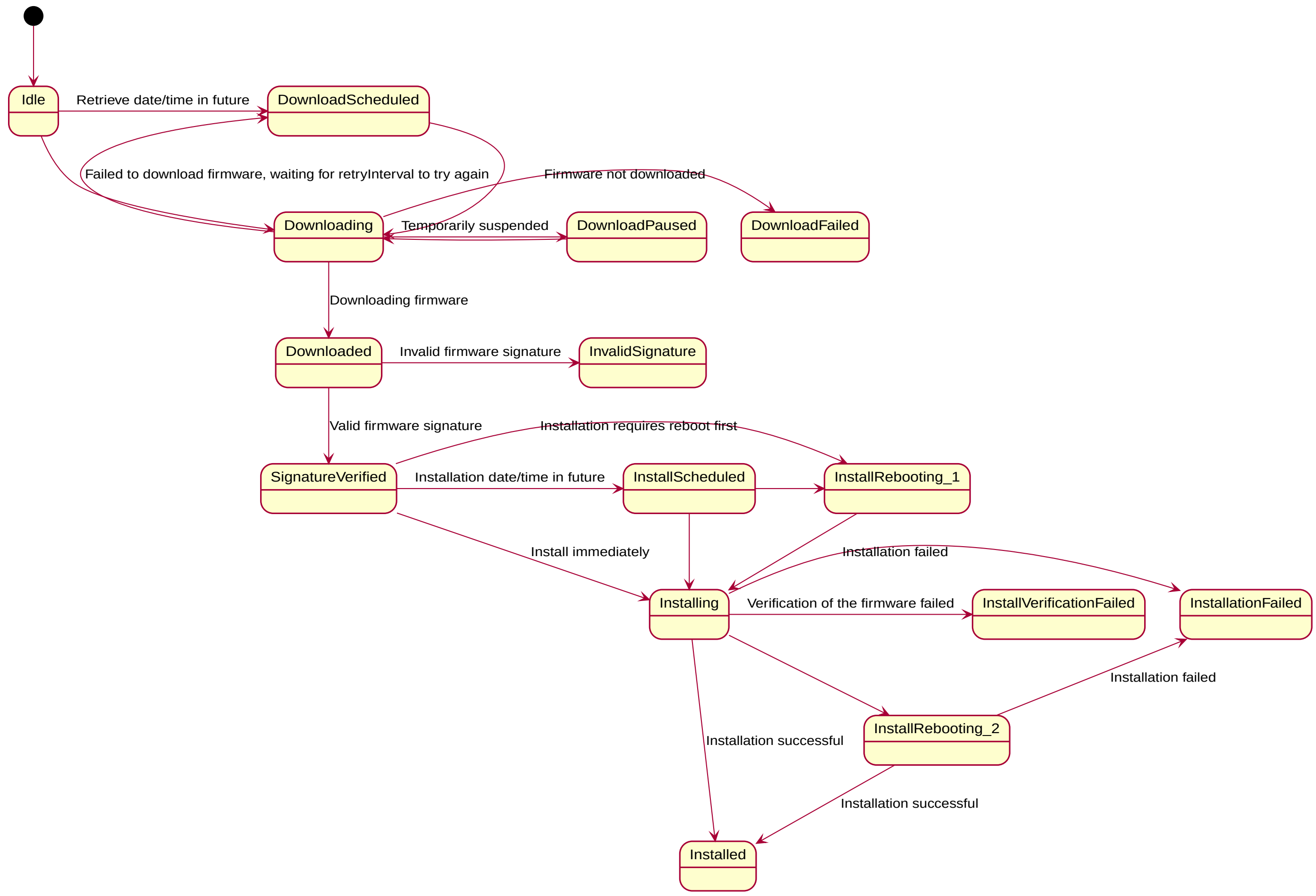
*Change to sequence diagram Figure 118*

- Add an optional block "Rebooting..." above the message FirmwareStatusNotificationRequest(*status* = Installed).


Charging Station
CSMS
UpdateFirmwareRequest()
UpdateFirmwareResponse()
FirmwareStatusNotificationRequest(Status = Downloading)
FirmwareStatusNotificationResponse()
Download firmware
Downloading firmware...
FirmwareStatusNotificationRequest(Status = Downloaded)
FirmwareStatusNotificationResponse()
Waiting for transactions to finish...
It is not fixed in what order the FirmwareStatusNotificationRequests are sent and in what order rebooting takes place.
opt
[if reboot required before installing firmware]
FirmwareStatusNotificationRequest(InstallRebooting)
FirmwareStatusNotificationResponse()
Reboot
Rebooting...
FirmwareStatusNotificationRequest(Status = Installing)
FirmwareStatusNotificationResponse()
Install firmware
Installing...
opt
[if reboot required after installing to activate firmware]
FirmwareStatusNotificationRequest(InstallRebooting)
FirmwareStatusNotificationResponse()
Reboot
Rebooting...
FirmwareStatusNotificationRequest(Installed)
FirmwareStatusNotificationResponse()
opt
Rebooting...

## 14.2. Page 273 - (v2) Firmware update process graph is incomplete [634]

The states for DownloadScheduled, InstallScheduled and InstallRebooting were missing.



## 14.3. Page 273 - (v1) Missing requirement if firmware verification fails [455]

A requirement for the status notification `InstallVerificationFailed` is missing.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| L01.FR.29 | If the verification of the new firmware (e.g. using a checksum or some other means) fails | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status `InstallVerificationFailed` | |

## 14.4. Page 274, 278 - (v1) Changed note of L01.FR.13 [456]

### 14.4.1. Page 274, Use case L01

The note for L01.FR.13 conflicts with requirement L01.FR.24 and is therefore changed.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | L01.FR.13 | When the Charging Station enters the Download Scheduled state. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | For example when it is busy with installing another firmware or it is busy Charging. |
| New text | L01.FR.13 | When the Charging Station enters the Download Scheduled state. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | **For example when it is busy charging.** |

### 14.4.2. Page 278, Use case L02

The note for L02.FR.07 conflicts with requirement L02.FR.15 and is therefore changed.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | L02.FR.07 | When the Charging Station enters the Download Scheduled state. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | For example when it is busy with installing another firmware or it is busy Charging. |
| New text | L02.FR.07 | When the Charging Station enters the Download Scheduled state. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | **For example when it is busy charging.** |

## 14.5. Page 274 - (v2) Changed preconditions for DownloadScheduled [614]

The precondition of L01.FR.13 is not clear, because a "Download Scheduled" state is not defined anywhere.

We therefore change it as follows in both L01 and L02:

### 14.5.1. Page 274 - L01

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | L01.FR.13 | When the Charging Station enters the Download Scheduled state. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | For example when it is busy charging. |
| New text | L01.FR.13 | When the Charging Station **does not start downloading firmware, because it is busy charging or because *retrieveDateTime* is in the future** | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | |

### 14.5.2. Page 278 - L02

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | L02.FR.07 | When the Charging Station enters the Download Scheduled state. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | For example when it is busy with installing another firmware or it is busy Charging. |
| New text | L02.FR.07 | When the Charging Station **does not start downloading firmware, because it is busy charging or because *retrieveDateTime* is in the future** | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadScheduled. | |

## 14.6. Page 275, 279 - (v1) Requirement for DownloadFailed missing [384]

A requirement for `DownloadFailed` is missing, even though it is shown in the figure 117 of the firmware update process.

### 14.6.1. Page 275, Use case L01

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| L01.FR.30 | When the Charging Station has failed all retry attempts to download the firmware. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadFailed. | A Charging Station MAY send a new Downloading status upon each retry attempt. |

### 14.6.2. Page 279, Use case L02

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| L02.FR.19 | When the Charging Station has failed all retry attempts to download the firmware. | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status DownloadFailed. | A Charging Station MAY send a new Downloading status upon each retry attempt. |

## 14.7. Page 275, 279 - (v2) Requirement for Installed is unclear [634]

The requirement to report `Installed` was not very clear for use case L01 and missing for use case L02.

### 14.7.1. Page 275, Use case L01

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | L01.FR.28 | After Charging Station has sent FirmwareStatusNotificationRequest with *status* = Installed | Charging Station SHOULD have activated the new firmware or do so immediately. This MAY involve an automatic reboot, but not necessarily so. | |
| New text | L01.FR.28 | When the Charging Station has successfully installed the new<br>firmware AND<br>has activated the new firmware<br>AND<br>has NOT already reported status Installed | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status Installed | It is recommended to report status Installed after the new firmware has been activated. |

### 14.7.2. Page 279, Use case L02

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| L02.FR.20 | When the Charging Station has successfully installed the new<br>firmware AND<br>has activated the new firmware AND has NOT already reported status Installed | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status Installed | It is recommended to report status Installed after the new firmware has been activated. |

## 14.8. Page 275 - (v2) Requirement for SecurityEvent FirmwareUpdated [648]

The security event "FirmwareUpdated" is marked as a critical event, but a requirement to send it is missing.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| L01.FR.31 | L01.FR.28 | The Charging Station SHALL send a SecurityEventNotificationRequest message with *type* = "FirmwareUpdated". | |

## 14.9. Page 275 - (v2) Allow the Charging Station to report InstallRebooting after installing, but before activating the new firmware

Requirement L01.FR.15 only covers one of the possible situations InstallRebooting can be sent. We need to allow the Charging Station to send InstallRebooting as described by the InstallRebooting FirmwareStatusEnumType description.

*New requirement*

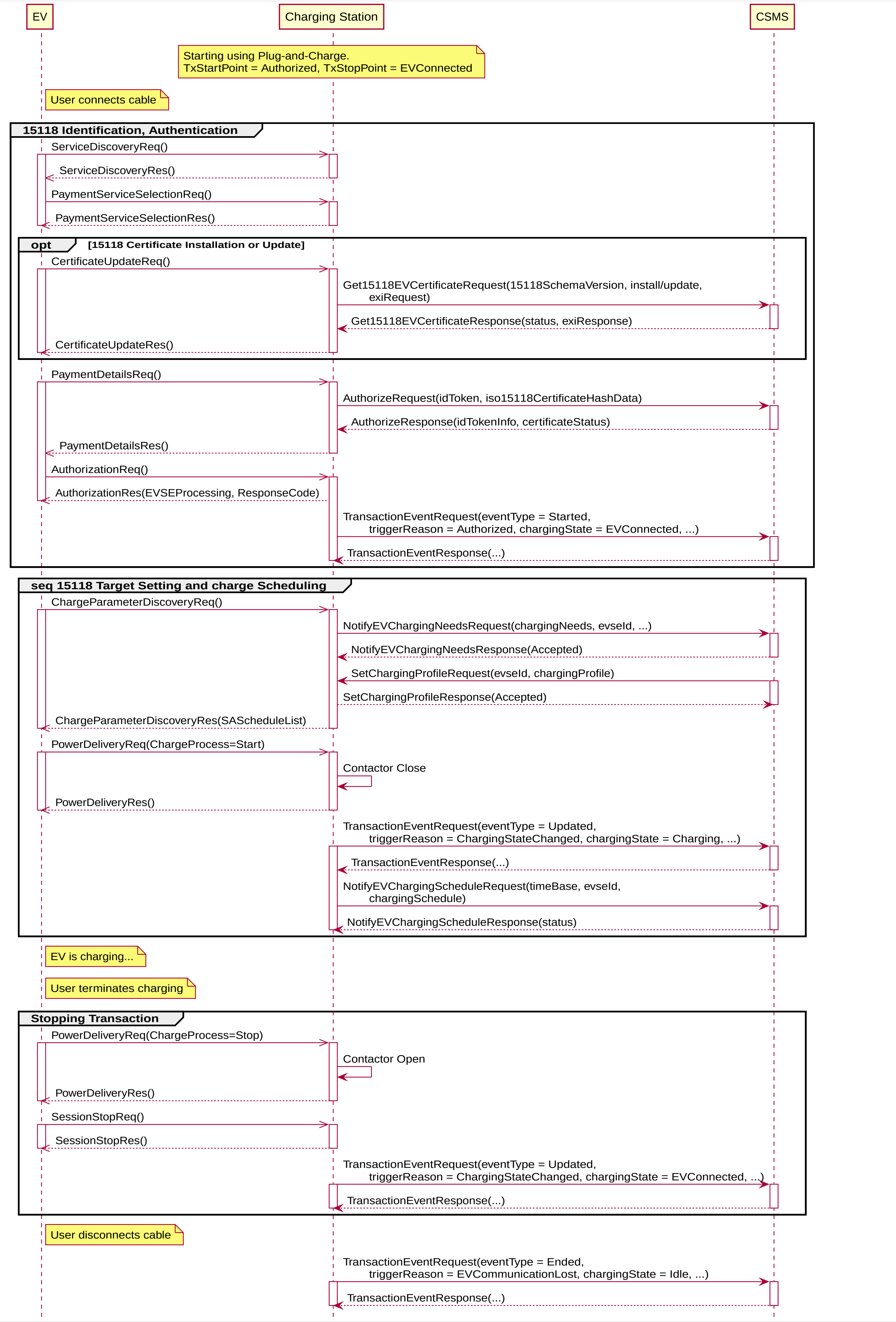
| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| L01.FR.32 | When the Charging Station has successfully installed the new firmware AND<br>the Charging Station needs to reboot before activating the new firmware | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status set to Installed or preferably to InstallRebooting and report another FirmwareStatusNotificationRequest with status Installed after the new firmware has been activated. | It is optional to report the FirmwareStatusNotificationRequest with status InstallRebooting, however if it is deemed necessary to report to the CSMS that the Charging Station succeeded in installing the new firmware, but needs to reboot before being able to activate the new firmware, it is recommended to use status InstallRebooting for this. |
| L02.FR.21 | When the Charging Station has successfully installed the new firmware AND<br>the Charging Station needs to reboot before activating the new firmware | The Charging Station SHALL send a FirmwareStatusNotificationRequest with status set to Installed or preferably to InstallRebooting and report another FirmwareStatusNotificationRequest with status Installed after the new firmware has been activated. | It is optional to report the FirmwareStatusNotificationRequest with status InstallRebooting, however if it is deemed necessary to report to the CSMS that the Charging Station succeeded in installing the new firmware, but needs to reboot before being able to activate the new firmware, it is recommended to use status InstallRebooting for this. |

# 15. Use case M ISO15118 Certificate Management

## 15.1. Page 285 - (v2) Chapter M: improved sequence diagram Figure 121

The sequence diagram for an ISO 15118 charging session had minor flaws in it.

Below is an improved version of the diagram.


EV
Charging Station
CSMS
Starting using Plug-and-Charge.
TxStartPoint = Authorized, TxStopPoint = EVConnected
User connects cable
15118 Identification, Authentication
ServiceDiscoveryReq()
ServiceDiscoveryRes()
PaymentServiceSelectionReq()
PaymentServiceSelectionRes()
opt
[15118 Certificate Installation or Update]
CertificateUpdateReq()
Get15118EVCertificateRequest(15118SchemaVersion, install/update,
exiRequest)
Get15118EVCertificateResponse(status, exiResponse)
CertificateUpdateRes()
PaymentDetailsReq()
AuthorizeRequest(idToken, iso15118CertificateHashData)
AuthorizeResponse(idTokenInfo, certificateStatus)
PaymentDetailsRes()
AuthorizationReq()
AuthorizationRes(EVSEProcessing, ResponseCode)
TransactionEventRequest(eventType = Started,
triggerReason = Authorized, chargingState = EVConnected, ...)
TransactionEventResponse(...)
seq 15118 Target Setting and charge Scheduling
ChargeParameterDiscoveryReq()
NotifyEVChargingNeedsRequest(chargingNeeds, evseId, ...)
NotifyEVChargingNeedsResponse(Accepted)
SetChargingProfileRequest(evseId, chargingProfile)
SetChargingProfileResponse(Accepted)
ChargeParameterDiscoveryRes(SAScheduleList)
PowerDeliveryReq(ChargeProcess=Start)
Contactor Close
PowerDeliveryRes()
TransactionEventRequest(eventType = Updated,
triggerReason = ChargingStateChanged, chargingState = Charging, ...)
TransactionEventResponse(...)
NotifyEVChargingScheduleRequest(timeBase, evseId,
chargingSchedule)
NotifyEVChargingScheduleResponse(status)
EV is charging...
User terminates charging
Stopping Transaction
PowerDeliveryReq(ChargeProcess=Stop)
Contactor Open
PowerDeliveryRes()
SessionStopReq()
SessionStopRes()
TransactionEventRequest(eventType = Updated,
triggerReason = ChargingStateChanged, chargingState = EVConnected, ...)
TransactionEventResponse(...)
User disconnects cable
TransactionEventRequest(eventType = Ended,
triggerReason = EVCommunicationLost, chargingState = Idle, ...)
TransactionEventResponse(...)

## 15.2. Page 288 - (v1) Improving definition of V2GRootCertificate [283]

| | | |
|---|---|---|
| Old text | V2GRootCertificate | Certificate of the V2G Root. The V2G Charging Station Certificate MUST BE derived from this root. |
| New text | V2GRootCertificate | Certificate of the **ISO15118** V2G Root. The V2G Charging Station Certificate MUST BE derived from this root. |

## 15.3. Page 292 - (v1) Use Cases M01 and M02, contract certificate pool [288]

The prerequisites for use case M01 should be identical to those for M02.
The prerequisite to use the ISO 15118 contract pool was too restrictive.

### 15.3.1. Page 292 - (v1) M01

| | |
|---|---|
| Old text | 1. See ISO15118-1, use case Prerequisites C2, page 22.<br>2. CSMS should be able to communicate with the contract certificate pool |
| New text | 1. Communication between EV and EVSE SHALL be established successfully.<br>2. Online connection between Charging Station and CSMS SHALL be possible.<br>3. CSMS should be able to communicate with a third party that can process the CertificateInstallationRequest, for example a contract certificate pool. |

### 15.3.2. Page 292 - (v1) Requirement M01.FR.01

The note for this requirement is changed to:

"The CSMS is responsible for forwarding it to the secondary actor which will process the CertificateUpdateRequest. This could be a contract certificate pool as outlined in application guide VDE-AR-2802-100-1."

### 15.3.3. Page 293 - (v1) M02

| | |
|---|---|
| Old text | 1. Communication between EV and EVSE SHALL be established successfully.<br>2. Online connection between Charging Station and CSMS SHALL be possible.<br>3. CSMS should be able to communicate with the contract certificate pool |
| New text | 1. Communication between EV and EVSE SHALL be established successfully.<br>2. Online connection between Charging Station and CSMS SHALL be possible.<br>3. CSMS should be able to communicate with **a third party that can process the CertificateInstallationRequest, for example a contract certificate pool.** |

### 15.3.4. Page 294 - (v1) Requirement M02.FR.01

The note for this requirement is changed to:

> "The CSMS is responsible for forwarding it to the secondary actor which will process the CertificateUpdateRequest. This could be a contract certificate pool as outlined in application guide VDE-AR-E 2802-100-1."

## 15.4. Page 294 - (v1) Some occurences of *typeOfCertificate* instead of *certificateType* [389]

At some locations the field *certificateType* is called *typeOfCertificate*.
Change all occurences of *typeOfCertificate* to *certificateType* at the following locations:

| Chapter | Item |
|---|---|
| M03 | Figure 126 |
| M03 | M03.FR.02 |
| M03 | M03.FR.03 |
| M03 | M03.FR.05 |
| Messages | 1.4.1 CertificateSignedRequest |
| Enumerations | 3.54 MessageTriggerEnumType |

## 15.5. Page 296 - (v2) Use case M05: reference to sub-CA certificate instead of root certificate [388]

The description of M05 was referring to an eMobility Operator Sub-CA certificate, but this must be a root certificate.

| | No. | Type | Description |
|---|---|---|---|
| Old text | **4** | **Description** | The CSMS requests the Charging Station to install a new CSMS root certificate, Sub-CA certificate for an eMobility Operator, Manufacturer root, or a V2G root certificate. |
| New text | **4** | **Description** | The CSMS requests the Charging Station to install a new CSMS root certificate, **an eMobility Operator root certificate**, Manufacturer root **certificate**, or a V2G root certificate. |

## 15.6. Page 296 - (v1) Use case M04, A Charging Station should be allowed to prevent the deletion of the last certificate from a defined certificate type

*Changed requirement*

| Version | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | M04.FR.02 | M04.FR.01 AND The requested certificate was found | The Charging Station SHALL delete it, and indicate success by setting 'status' to 'Accepted' in the DeleteCertificateResponse. | |
| New | M04.FR.02 | M04.FR.01 AND The requested certificate was found | The Charging Station SHALL **attempt to** delete it, and indicate success by setting *status* to `Accepted` in the DeleteCertificateResponse. | |
| Old | M04.FR.03 | M04.FR.01 AND The deletion fails | The Charging Station SHALL indicate failure by setting 'status' to 'Failed' in the DeleteCertificateResponse. | |
| New | M04.FR.03 | M04.FR.01 AND (The deletion fails<br>**OR**<br>**the Charging Station rejects the request to delete the specified certificate.)** | The Charging Station SHALL indicate failure by setting *status* to `Failed` in the DeleteCertificateResponse. | **A Charging Station may reject the request to prevent the deletion of a certificate, if it is the last one from its certificate type.** |

### 15.6.1. Page 399 - DeleteCertificateStatusEnumType

*Changed description*

| | Value | Description |
|---|---|---|
| Old text | Failed | Processing failure. |

|  | Value | Description |
|---|---|---|
| New text | Failed | **The Charging Station either failed to remove the certificate or rejected the request. A Charging Station may reject the request to prevent the deletion of a certificate, if it is the last one from its certificate type.** |

# 15.7. Page 296 - (v1) Requirement M04.FR.06 misses status code [471]

*Changed requirement*

|  | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | M04.FR.06 |  | Deletion of the *Charging Station Certificate* SHALL NOT be possible via a DeleteCertificateRequest. |  |
| New text | M04.FR.06 | **M04.FR.01 AND<br>When *certificateHashData* refers to the *Charging Station Certificate* (see use case A)** | **Charging Station SHALL respond with DeleteCertificateReponse with *status* = `Failed`.** | **Deletion of the *Charging Station Certificate* is not allowed via DeleteCertificateRequest.** |

# 15.8. Page 296 - (v2) Use case M04: Deletion of certificate also deletes its child certificates [309]

When DeleteCertificateRequest is used to delete a root certificate or a sub-CA certificate, then all child certificates should also be deleted. If these would not be removed, then these certificates remain as orphan certificates that can no longer be removed, because the hash values needed for the DeleteCertificateRequest can not be calculated anymore, since the issuer certificates have already been removed.

This leads to the following new requirement.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| M04.FR.08 | M04.FR.02 AND<br>Certificate to delete is a sub-CA or root certificate | Charging Station MAY also delete all child certificates. | Else these child certificates remain as unusable orhan certificates that can no longer be deleted. |

# 15.9. Page 298 - (v2) Use case M05: missing requirement about replacing a certificate [617]

Use case M05 mentions the installing of a new certificate, but the certificate does not necessarily have to be new. The message can also be used to install a certificate that is already present, in which case the certificate will be overwritten. This may not be a likely case in real life, but is very useful during testing.

A requirement is added to make explicit that this is allowed.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| M05.FR.17 | NOT M05.FR.10 AND<br>After receiving an InstallCertificateRequest for a certificate that is already present in the certificate trust store of the Charging Station | The Charging Station SHALL replace the certificate and respond with InstallCertificateResponse with *status* = `Accepted`. |

## 15.10. Page 299 - (v2) M06.FR.04 refers to wrong status [576]

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | M06.FR.04 | M06.FR.01<br>AND<br>The CSMS was not successful in retrieving the OCSP certificate status | The CSMS SHALL indicate it was not successful by setting 'status' to 'Rejected' in the GetCertificateStatusResponse. |
| New text | M06.FR.04 | M06.FR.01<br>AND<br>The CSMS was not successful in retrieving the OCSP certificate status | The CSMS SHALL indicate it was not successful by setting ***status* to `Failed`** in the GetCertificateStatusResponse. |

# 16. Use case N Diagnostics

## 16.1. Page 303 - (v1) Misspelled enumeration in N01.FR.10 [443]

The enumeration value *UploadFailed* should be *UploadFailure*.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | N01.FR.10 | When uploading a log document failed | The Charging Station SHALL send a LogStatusNotificationRequest with status *UploadFailed*, *BadMessage*, *PermissionDenied* OR *NotSupportedOperation*. | It is RECOMMENDED to send a status that describes the reason of failure as precise as possible. |
| New text | N01.FR.10 | When uploading a log document failed | The Charging Station SHALL send a LogStatusNotificationRequest with status ***UploadFailure***, *BadMessage*, *PermissionDenied* OR *NotSupportedOperation*. | It is RECOMMENDED to send a status that describes the reason of failure as precise as possible. |

## 16.2. Page 303 - (v1) Requirements for GetLogRequest are incomplete [497]

The specification on how to upload a log file to CSMS is not complete. The following implicit requirements are made explicit by the following additions.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| N01.FR.14 | | It is RECOMMENDED that Charging Station and CSMS support at least HTTP(s) as transport mechanism for the log file upload | HTTP transport is most likely to be supported, since it is also used for OCPP messaging. |
| N01.FR.15 | | Charging Station SHALL at least support the CSMS trust chain for secure transports | |
| N01.FR.16 | | It is RECOMMENDED that Charging Station supports the usual CAs provided by the operating system | The log file storage of CSMS may be a cloud service operated separately from the CSMS itself and not part of the CSMS trustchain. |
| N01.FR.17 | When CSMS requires basic authorization for the upload | CSMS is RECOMMENDED to require a different basic authorization password for the upload, then the one used for OCPP connectivity. | This is to avoid leaking the OCPP password to 3rd parties if the log file storage is a different system.<br>Basic authorization can be added to the URL as follows: http://*username:password*@csms.org/logs |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| N01.FR.18 | | Is is RECOMMENDED that CSMS accepts both PUT and POST requests for uploads from Charging Station. | |
| N01.FR.19 | When Charging Station uses a HTTP(s) POST request to upload the log file | Charging Station SHALL provide at least the following attributes: `Content-Type:` (e.g. application/octet-stream) and `Content-Disposition:` with a specification of the filename. | For example:<br>Content-Type: application/octet-stream<br>Content-Disposition: form-data; name="uploadedfile"; filename="logfile_20210420.zip" |

## 16.3. Page 303 - (v2) LogStatusNotification for AcceptedCanceled made explicit [594]

A charging station must send LogStatusNotificationRequest messages to report about the progress of the upload. In requirement N01.FR.12 it is stated that a *status* = `AcceptedCanceled` must be returned by the GetLogResponse message, but it is not made explicit that this, of course, also leads to a LogStatusNotificationRequest with an `AcceptedCanceled` status.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| N01.FR.20 | N01.FR.12 AND<br>Charging Station cancels the log file upload | The Charging Station SHALL send a LogStatusNotificationRequest with *status* = `AcceptedCanceled`. | N01.FR.12 is a "SHOULD" requirement. Only send status notification when requirement is executed. |

## 16.4. Page 305 - (v1) Use case N02 Get Monitoring Report: conflicting requirements [355]

The following two requirements of use case N02 are conflicting:

| ID | Precondition | Requirement definition |
|---|---|---|
| N02.FR.01 | When the Charging Station receives a getMonitoringReportRequest for supported *monitoringCriteria* OR without *monitoringCriteria* | The Charging Station SHALL send a getMonitoringReportResponse with Accepted. |
| N02.FR.10 | When the Charging Station receives a GetMonitoringReportRequest with a combination of criteria which results in an empty result set. | The Charging Station SHALL respond with a GetMonitoringReportResponse(*status*=`EmptyResultSet`). |

To fix this, the precodition of requirement N02.FR.01 needs to be changed as follows:

*Changed requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| N02.FR.01 | **NOT N02.FR.10 AND**<br>When the Charging Station receives a getMonitoringReportRequest for supported *monitoringCriteria* OR without *monitoringCriteria* | The Charging Station SHALL send a getMonitoringReportResponse with Accepted. |

## 16.5. Page 305 - (v2) Use case N02 GetMonitoringReport handling of empty variables or instances [565]

**NOTE** This is similar to Page 59 - (v2) Use case B08: GetReport handling of empty variables or instances [565]

The GetMonitoringReportRequest takes *componentVariable* as a filter that limits the scope of the report. This implies that if no *variable* is specified, all variables of the component are reported. Similarly, if no *instance* is specified, then all instances of component or variable are reported, and if no *evse* is provided, than components over all EVSEs are reported.

This is made explicit in the following requirements.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| N02.FR.15 | When Charging Station receives a GetMonitoringReportRequest with *componentVariable* elements in which *component.instance* and/or *component.evse* are missing | The Charging Station SHALL report for every instance and/or EVSE of the *component* in *componentVariable*. |
| N02.FR.16 | When Charging Station receives a GetMonitoringReportRequest with *componentVariable* elements in which *variable* is missing | The Charging Station SHALL report for every *variable* of the *component* in *componentVariable*. |
| N02.FR.17 | When Charging Station receives a GetMonitoringReportRequest with *componentVariable* elements in which *variable* is present, but *instance* is missing | The Charging Station SHALL report for every instance of the *variable* of the *component* in *componentVariable*. |

## 16.6. Page 306 - (v2) Use case N03: SetMonitoringBaseRequest clarification [616]

The following provides a better description of the SetMonitoringBaseRequest message, because the difference between the various settings was not well-described.

*Improved description*

| | | | |
|---|---|---|---|
| Old text | **8** | **Remark(s)** | Note, that upon receipt of a SetMonitoringBaseRequest the Charging Station will discard of any previously configured monitoring settings and will activate the monitoring settings that are related to the MonitoringBase. |
| New text | **8** | **Remark(s)** | 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.<br>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.<br>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. |

*Changed requirements*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | N03.FR.03 | N03.FR.01 AND<br>When the Charging Station received a setMonitoringBaseRequest with *monitoringBase* All | Then the Charging Station SHALL activate all preconfigured monitoring. |
| New text | N03.FR.03 | N03.FR.01 AND<br>When the Charging Station received a setMonitoringBaseRequest with *monitoringBase* All | Then the Charging Station SHALL activate all preconfigured monitoring **whilst leaving all installed custom monitors (including changed preconfigured monitors) intact.** |
| Old text | N03.FR.04 | N03.FR.01 AND<br>When the Charging Station received a setMonitoringBaseRequest with *monitoringBase* FactoryDefault | Then the Charging Station SHALL activate the default monitoring settings as recommended by the manufacturer. |
| New text | N03.FR.04 | N03.FR.01 AND<br>When the Charging Station received a setMonitoringBaseRequest with *monitoringBase* FactoryDefault | 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. |

## 16.7. Page 307 - (v1) Misspelled type in requirement N04.FR.05 [372]

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | N04.FR.05 | When the Charging Station receives a SetVariableMonitoringRequest with an MonitorType which is not supported by the specific Variable | The Charging Station SHALL set the *attributeStatus* field in the corresponding SetMonitoringResult to: NotSupportedMonitorType. | |
| New text | N04.FR.05 | When the Charging Station receives a SetVariableMonitoringRequest with an MonitorType which is not supported by the specific Variable | The Charging Station SHALL set the *attributeStatus* field in the corresponding SetMonitoringResult to: **UnsupportedMonitorType**. | |

## 16.8. Page 308 - (v1) Requirement N04.FR.10 is too restrictive [358]

Requirement N04.FR.10 forbids two monitors to have the same type and severity, but it fails to mention that this only applies when the monitors are for the same component/variable combination. For example, it should be possible to set the same `EVSE.Power` monitor on all EVSEs with the same severity.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | N04.FR.10 | When the Charging Station receives a SetVariableMonitoringRequest with a type/severity combination for which a monitor already exists. | The Charging Station SHALL set the *attributeStatus* field in the corresponding SetMonitoringResult to: Duplicate. | There cannot be two monitors of the same type with the same severity. E.g. with an UpperThreshold at value "67" and severity "4-Error" there cannot be another Upperthreshold at value "78" with same severity "4-Error" defined. Also it is only possible to replace a monitor on Id. |
| New text | N04.FR.10 | When the Charging Station receives a SetVariableMonitoringRequest **for a *component/variable* combination for which a monitor with the same *type* and *severity* already exists with a different *id*.** | The Charging Station SHALL set the *attributeStatus* field in the corresponding SetMonitoringResult to: Duplicate. | There cannot be two monitors of the same type with the same severity **on the same variable**. E.g. when a component/variable has a monitor with an UpperThreshold at value "67" and severity "4-Error", then there cannot be another UpperThreshold at value "78" with same severity "4-Error" defined. |

## 16.9. Page 308 - (v1) Precondition incomplete in N04.FR.11 [373]

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | N04.FR.11 | When the Charging Station receives a SetVariableMonitoringRequest without an Id | The Charging Station will generate an Id and return it in the SetVariableMonitoringResponse. | |
| New text | N04.FR.11 | When the Charging Station receives a SetVariableMonitoringRequest<br>without an Id **AND N04.FR.08** | The Charging Station will generate an Id and return it in the SetVariableMonitoringResponse. | |

## 16.10. Page 308 - (v1) Precondition unclear in N04.FR.16 [374]

The precondition of N04.FR.16 did not make clear, that this requirement refers to the situation where CSMS wants to change a monitor with the given Id.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old text | N04.FR.16 | When the Charging Station receives a SetVariableMonitoringRequest<br>AND<br>the given Component/Variable combination does NOT correspond with the existing VariableMonitor. | The Charging Station SHALL respond with *Rejected* AND NOT replace the VariableMonitor. | It is not allowed to change Variable or Component of a monitor. |
| New text | N04.FR.16 | **When the Charging Station receives a SetVariableMonitoringRequest**<br>**with an Id AND**<br>**a monitor exists matching the**<br>**given Id AND**<br>the given Component/Variable combination does NOT correspond with the existing VariableMonitor. | The Charging Station SHALL respond with *Rejected* AND NOT replace the VariableMonitor. | It is not allowed to change Variable or Component of a monitor. |

## 16.11. Page 308 - (v2) Use case N04: Requirement missing that hardwired monitors cannot be changed [628]

Requirement N04.FR.15 states that a `PreconfiguredMonitor` becomes a `CustomMonitor` when it is changed. A requirement is missing that a `HardWiredMonitor` is not allowed to be changed.

This is made explicit as a new requirement.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| N04.FR.18 | N04.FR.12 AND<br>The *id* in the SetVariableMonitoringRequest refers to a `HardWiredMonitor` | The Charging Station SHALL respond with *Rejected* AND NOT replace the VariableMonitor. | It is not possible to change a hardwired monitor. |

## 16.12. Page 308 - (v2) Use case N04: Recommend GetMonitoringReport after reboot [643]

The custom variable monitors are persistent after a reboot. However, there is no guarantee that these monitors still have the same ID. Some hardwired monitors or pre-configured monitors may have been changed before the reboot, either as a result of a local configuration change in the charging station or as result of a firmware update. When that happens, the IDs of the persistent custom variable monitors may have changed in order to avoid clashes with IDs of the new pre-configured monitors. It is therefore strongly recommended to always request a GetMonitoringReport after a charging station has rebooted.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| N04.FR.19 | The Charging Station has rebooted | The CSMS IS RECOMMENDED to send a GetMonitoringReportRequest message to get a new list of monitors. | Custom monitors are persistent after reboot or firmware update, but IDs may have changed. |

## 16.13. Page 310 - (v2) Use case N06: Requirement missing when exceeding ItemsPerMessage/BytePerMessage [620]

Requirement N06.FR.04 states that CSMS shall not send more *id* elements in a ClearVariableMonitoringRequest then allowed by configuration variable MonitoringCtrlr.ItemsPerMessage[ClearVariableMonitoring].

Similarly, the size of the message cannot be larger than MonitoringCtrlr.BytesPerMessage[ClearVariableMonitoring].

Requirements for the charging station on how to respond to such a situation are missing. Since a memory-constrained charging station may not be able to properly process such a message, the requirement allows a charging station in such a situation to return a CALLERROR.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| N06.FR.06 | Charging Station receives a ClearVariableMonitoringRequest with more *id* elements than allowed by `ItemsPerMessageClearVariableMonitoring` | The Charging Station MAY respond with a CALLERROR(OccurenceConstraintViolation) |
| N06.FR.07 | Charging Station receives a ClearVariableMonitoringRequest with a length of more bytes than allowed by `BytesPerMessageClearVariableMonitoring` | The Charging Station MAY respond with a CALLERROR(FormatViolation) |

## 16.14. Page 310 - (v1) Error in requirement N06.FR.05 [369]

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | N06.FR.05 | | For every *id* in a ClearVariableMonitoringRequest the CSMS SHALL add a *clearMonitoringResult* element to the ClearVariableMonitoringResponse send to the CSMS. |
| New text | N06.FR.05 | | For every *id* in a ClearVariableMonitoringRequest the **Charging Station** SHALL add a *clearMonitoringResult* element to the ClearVariableMonitoringResponse **sent** to the CSMS. |

## 16.15. Page 311 - (v2) Use case N07: remark about OfflineMonitoringEventQueuingSeverity [633]

The following suggestion has been added as a remark to use case N07.

| | | |
|---|---|---|
| **8** | **Remark(s)** | Requirement N07.FR.04 states that events with a severity equal or less than OfflineMonitoringEventQueuingSeverity shall be queued while the charging station is offline, and delivered once online. This implies that events with a severity greater than OfflineMonitoringEventQueuingSeverity will not be sent to CSMS. The result is, that the logical chain of events may be broken when the charging station is back online.<br>For example, a monitoring event for a variable exceeding a threshold occurred while offline and was not sent. Once back online, at some point in time the monitoring event is reported with the variable *cleared* set to true, but CSMS did not even know that the threshold had been exceeded. CSMS will have to be able to deal with that.<br>This problem can be prevented, while still adhering to the specification, by not simply discarding these monitoring events, but by delaying the evaluation of those monitors that exceed OfflineMonitoringEventQueuingSeverity, until the charging station comes back online. The result is, that when the charging station is back online, CSMS will get the monitoring events that apply to the current situation, and it is fully up-to-date regarding the monitors. Only those monitoring events that were triggered & cleared during the offline period will remain invisible to CSMS. |

## 16.16. Page 311 - (v2) Typo in N07.FR.04 [633]

Requirement N07.FR.04 has misspelled OfflineMonitoringEventQueuingSeverity with an "e" too many in Queuing.

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | N07.FR.04 | When a monitor is triggered AND<br>The severity number of the monitor is equal to or lower than the severity number set in the Configuration Variable `OfflineMonitoringEventQueueingSeverity`<br>AND<br>The Charging Station is *offline* | The Charging Station SHALL queue this NotifyEventRequest and deliver it when it is back online. |
| New text | N07.FR.04 | When a monitor is triggered AND<br>The severity number of the monitor is equal to or lower than the severity number set in the Configuration Variable **`OfflineMonitoringEventQueuingSeverity`**<br>AND<br>The Charging Station is *offline* | The Charging Station SHALL queue this NotifyEventRequest and deliver it when it is back online. |

## 16.17. Page 311 - (v1) N07.FR.06 uses variableMonitoringId [477]

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | N07.FR.06 | When a monitor is triggered | An eventData element in a NotifyEventRequest SHALL contain the Component, Variable and variableMonitoring data that caused the event. |
| New text | N07.FR.06 | When a monitor is triggered | An eventData element in a NotifyEventRequest SHALL contain the Component, Variable and ***variableMonitoringId*** that caused the event. |

## 16.18. Page 312 - (v1) Wrong precondition in requirement N07.FR.14 [356]

Requirement N07.FR.14 refers to N07.FR.08, but this requirement does not exist.

The text can be fixed as follows:

*Changed requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old text | N07.FR.14 | N07.FR.08 AND<br>A reboot occurred AND<br>After the reboot the *UpperThreshold* or *LowerThreshold* is cleared. | The Charging Station SHALL send a NotifyEventRequest with an eventData with the attribute *cleared* is true. |
| New text | N07.FR.14 | **When a variableMonitoring setting of type *UpperThreshold* or *LowerThreshold* has been triggered AND after a reboot occurred the monitored value returned within the configured threshold.** | The Charging Station SHALL send a NotifyEventRequest with an eventData with the attribute *cleared* is true. |

## 16.19. Page 312 - (v1) Added note to requirements N07.FR.16 and NR.FR.17 [359]

The requirement N07.FR.16 states that upon *exceeding the threshold* a notification shall be sent. When dealing with integer values this may seem counter-intuitive. For example, when one wants to be notified when a value reaches 10, then one needs to set the threshold at 9.

Similar reasoning applies to N07.FR.17.

A note is added to these requirements, as follows:

*Changed requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| N07.FR.16 | When there is a monitor with type UpperThreshold on a Component/Variable combination<br>AND<br>the Actual value (attributeType Actual) of the Variable exceeds *monitorValue* | The Charging Station SHALL send a NotifyEventRequest with *trigger* Alerting for the triggered monitor. | Notification is sent when exceeding the threshold, not on the threshold. |
| N07.FR.17 | When there is a monitor with type LowerThreshold on a Component/Variable combination<br>AND<br>the Actual value (attributeType Actual) of the Variable drops below *monitorValue* | The Charging Station SHALL send a NotifyEventRequest with *trigger* Alerting for the triggered monitor. | Notification is sent when dropping below the threshold, not on the threshold. |

## 16.20. Page 312 - (v1) Moved requirements about periodic monitors from N07 to N08 [367]

This errata moves two requirements from N07 to N08. There are no changes in functionality or behavior.

The requirements N07.FR.20 and N07.FR.21 are about periodic monitors and do not belong in N07 Alert Event.

*Changed requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| N07.FR.20 | When there is a monitor with type Periodic on a<br>Component/Variable combination AND the number of seconds specified in *monitorValue* have passed (starting from the time that this monitor was set or triggered) | The Charging Station SHALL send a NotifyEventRequest with *trigger* Periodic for the triggered monitor. |

| ID | Precondition | Requirement definition |
|---|---|---|
| N07.FR.21 | When there is a monitor with type PeriodicClockAligned on a Component/Variable combination AND<br>the number of seconds specified by *monitorValue*, starting from the nearest clock-aligned interval after this monitor was set, have passed (For example, a *monitorValue* of 900 will trigger event notices at 0, 15, 30 and 45 minutes after the hour, every hour) | The Charging Station SHALL send a NotifyEventRequest with *trigger* Periodic for the triggered monitor. |

There requirements are moved to N08 Periodic Event:

1. N07.FR.20 becomes **N08.FR.06**
2. N07.FR.21 becomes **N08.FR.07**

## 16.20.1. Page 313 - (v1) Requirement N08.FR.01 is replaced by N08.FR.06 and N08.FR.07

The precondition of requirement N08.FR.01 is not worded correctly, because a periodic monitor does not "reach" a *monitorValue*. Instead, the new requirements N08.FR.06 and N08.FR.07 take its place and N08.FR.01 is removed.

*Updated requirements for N08:*

| ID | Precondition | Requirement definition |
|---|---|---|
| N08.FR.01 | <deleted> | |
| N08.FR.02 | When the CSMS receives an NotifyEventRequest | The CSMS SHALL respond with an empty NotifyEventResponse. |
| N08.FR.03 | **N08.FR.06 OR N08.FR.07**<br>AND<br>The severity number of the monitor is equal to or lower than the severity number set in the Configuration Variable `OfflineMonitoringEventQueueingSeverity`<br>AND<br>The Charging Station is *offline* | The Charging Station SHALL queue this NotifyEventRequest and deliver it when it is back online. |
| N08.FR.04 | **N08.FR.06 OR N08.FR.07** AND<br>This NotifyEventRequest is the first or only report part. | The Charging Station SHALL set *seqNo* to 0. |
| N08.FR.05 | **N08.FR.06 OR N08.FR.07** AND<br>When the variableMonitoring setting which triggered the event is either of type Periodic or PeriodicClockAligned | The Charging Station SHALL set *trigger* to Periodic. |
| **N08.FR.06** | When there is a monitor with type Periodic on a Component/Variable combination AND<br>the number of seconds specified in *monitorValue* have passed (starting from the time that this monitor was set or triggered) | The Charging Station SHALL send a NotifyEventRequest with *trigger* Periodic for the triggered monitor. |
| **N08.FR.07** | When there is a monitor with type PeriodicClockAligned on a Component/Variable combination AND<br>the number of seconds specified by *monitorValue*, starting from the nearest clock-aligned interval after this monitor was set, have passed (For example, a *monitorValue* of 900 will trigger event notices at 0, 15, 30 and 45 minutes after the hour, every hour) | The Charging Station SHALL send a NotifyEventRequest with *trigger* Periodic for the triggered monitor. |

# 17. Use case P Data Transfer

## 17.1. Page 331 - (v2) Added example for using JSON in DataTransfer data [574]

The DataTransferRequest/Response contains a field without a length or type specification. It can be convenient to use this field as structured JSON content.

*Example of embedded JSON*

```
[2,
"<unique msg id>",
"DataTransfer",
{
  "vendorId": "com.mycompany.ice",
  "messageId": "iceParkedAtCs"
  "data": { "start_time": "2020-04-01T11:01:02" }
}
]
```

# 18. Messages

## 18.1. Page 339 - (v2) Error in description of CertificateSignedRequest [425]

Change the description of the field *certificateChain* as follows:

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | certificateChain | string[0..10000] | 1..1 | Required. The signed PEM encoded X.509 certificate. This can also contain the necessary sub CA certificates. In that case, the order of the bundle should follow the certificate chain, starting from the leaf certificate.<br>The Configuration Variable MaxCertificateChainSize can be used to limit the maximum size of this field. |
| New text | certificateChain | string[0..10000] | 1..1 | Required. The signed PEM encoded X.509 certificate. This **SHALL** also contain the necessary sub CA certificates, **when applicable**. ~~In that case,~~ The order of the bundle ~~should~~ follows the certificate chain, starting from the leaf certificate.<br>The Configuration Variable MaxCertificateChainSize can be used to limit the maximum size of this field. |

## 18.2. Page 344 - (v1) Typographical error in FirmwareNotificationRequest [480]

The following sentence in section 1.15.1 FirmwareStatusNotificationRequest in the chapter Messages, Datatypes & Enumerations contains a spelling errorl

| | |
|---|---|
| Old text | This contains the field definition of the FirmwareStatusNotifitacionRequest PDU sent by the Charging Station to the CSMS. |
| New text | This contains the field definition of the **FirmwareStatusNotificationRequest** PDU sent by the Charging Station to the CSMS. |

# 18.3. Page 345 - (v2) Enlarging *exiResponse* in Get15118EVCertificate [637]

**IMPORTANT** This errata entry only applies to installations that support ISO 15118 Plug-and-Charge.
It can only be implemented by modifying the original OCPP schema definition.

The message Get15118EVCertificateRequest passes a request for a new contract certificate from EV on towards CSMS. The ISO 15118 EXI-encoded message is passed on as-is in the field *exiRequest*. This field contains an OEM certificate and a list of installed V2G root certificates. CSMS forwards the *exiRequest* to a contract certificate pool and gets an *exiResponse* back that contains a contract certificate for the EV.

In OCPP the size of the *exiRequest* and *exiResponse* is limited to 5600 characters. Unfortunately, when the first Plug-and-charge contract certificates came to market, the *exiResponse* message turned out to be slightly larger, namely up to 6100 characters, due to the large amount of metadata in the certificates.

It is advised to accept strings for *exiResponse* being a maximum of 8000 characters long. The field length is defined in the OCPP schema definition for Get15118EVCertificateRequest and Get15118EVCertificateResponse. If a system wishes to support these larger fields, then there is no other option but to change the length of *exiResponse* in GET15118EVCertificateResponse.

A charging station can announce support for this bigger field size by setting the FieldLength variable of the OCPPCommCtrlr component in its device model. "FieldLength" is new **read-only** variable that has been added to OCPPCommCtrlr. The instance of this field determines the message and field that it applies to. For example:

- OCPPCommCtrlr.FieldLength["Get15118EVCertificateResponse.exiResponse"] = 8000

It is highly recommended to report this variable, because this lets CSMS know which charging stations support the longer fields.

## 18.3.1. Device model variable

Here is a description of the device model variable.

### `MessageFieldLength`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | OCPPCommCtrlr | |
| **Variable** | **variableName** | FieldLength | |
| | **variableInstance** | <message>.<field> | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | integer |
| **Description** | This variable is used to report the length of <field> in <message> when it is larger than the length that is defined in the standard OCPP message schema. | | |

## 18.3.2. Get15118EVCertificateResponse

Below is the definition of Get15118EVCertificateResponse with a larger *exiResponse*.

**NOTE** This message is based on CertificateInstallationRes from ISO 15118-2.

| Field Name | Field Type | Card. | Description |
|---|---|---|---|
| **status** | Iso15118EVCertificateStatusEnumType | 1..1 | Required. Indicates whether the message was processed properly. |
| **exiResponse** | string[0..5600] | 1..1 | Required. Raw CertificateInstallationRes response for the EV, Base64 encoded. The Charging Station can let the CSMS know it supports a higher field size by reporting this using the device model as OCPPCommCtrlr.FieldLength["Get15118EVCertificateResponse.exiResponse"] = <New max length> |
| **statusInfo** | StatusInfoType | 0..1 | Optional. Detailed status information. |

## 18.4. Page 349 - (v2) Better definition of *retries* [586]

The description of the field *retries* in GetLogRequest, PublishFirmwareRequest and UpdateFirmwareRequest is confusing, because it is not clear whether this field refers to the amount of "tries" or "retries".

### 18.4.1. Page 349 - GetLogRequest

| Field Name | Field Type | Card. | New description |
|---|---|---|---|
| retries | integer | 0..1 | Optional. This specifies how many times the Charging Station must **retry** to upload the log before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry. **If the value is 0, it means: no retries.** |

### 18.4.2. Page 456 - PublishFirmwareRequest

| Field Name | Field Type | Card. | New description |
|---|---|---|---|
| retries | integer | 0..1 | Optional. This specifies how many times the Charging Station must **retry** to download the firmware before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry. **If the value is 0, it means: no retries.** |

### 18.4.3. Page 468 - UpdateFirmwareRequest

| Field Name | Field Type | Card. | New description |
|---|---|---|---|
| retries | integer | 0..1 | Optional. This specifies how many times the Charging Station must **retry** to download the firmware before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry. **If the value is 0, it means: no retries.** |

## 18.5. Page 351 - (v1) Remark at InstallCertificateRequest [283]

Extend the description of the InstallCertificateRequest message with the following:

> "Note: This message is not for installing a TLS client certificate in a charging station. The CertificateSignedRequest mechanism is used for that."

## 18.6. Page 358 - (v2) Description of groupIdToken in RequestStartTransactionRequest [645]

The RequestStartTransactionRequest has a field *groupIdToken*, for which the purpose is not obvious. An explanation is added to the description.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **groupIdToken** | IdTokenType | 0..1 | Optional. The group identifier that the Charging Station must use to start a transaction. |
| New text | **groupIdToken** | IdTokenType | 0..1 | Optional. The *groupIdToken* is only relevant when the transaction is to be started on an EVSE for which a reservation for *groupIdToken* is active, and the configuration variable AuthorizeRemoteStart = false (otherwise the AuthorizeResponse could return the *groupIdToken*). |

## 18.7. Page 365 - (v1) Wrong description of timestamp in StatusNotificationRequest [379]

The description of *timestamp* reads: "Required. The time for which the status is reported. If absent time of receipt of the message will be assumed." Since *timestamp* is required it cannot be absent. Change description to:

> Required. The time for which the status is reported.

## 18.8. Page 366 - (v2) Description of IdToken in TransactionEventRequest [583]

From the description of *idToken* in TransactionEventRequest it may not be clear that an authorization is started and ended with an *idToken*. The sentence "once in a TransactionEventRequest for every authorization done for this transaction." refers to both the starting of authorization and the stopping. This has been made explicit in the new description.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **idToken** | IdTokenType | 0..1 | Optional. This contains the identifier for which a transaction has to be/was started. Is required when the EV Driver becomes authorized for this transaction. The IdToken should only be send once in a TransactionEventRequest for every authorization done for this transaction. |
| New text | **idToken** | IdTokenType | 0..1 | Optional. This contains the identifier for which a transaction **is (or will be)** started **or stopped**. Is required when the EV Driver becomes authorized for this transaction **and when the EV Driver ends authorization.** The IdToken should only be sent once in a TransactionEventRequest for every authorization **(for starting or for stopping)** done for this transaction. |

# 19. Datatypes

## 19.1. Page 371 - (v2) Description of CertificateHashDataType [542]

The descriptions of the fields in CertificateHashDataType have been improved to better match their definition in IETF RFC 6960).

*Changed descriptions*

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | issuerNameHash | identifierString[0..128] | 1..1 | Required. Hashed value of the Issuer DN (Distinguished Name). |
| New text | issuerNameHash | identifierString[0..128] | 1..1 | Required. **The hash of the issuer's distinguished name (DN), that must be calculated over the DER encoding of the issuer's name field in the certificate being checked.** |
| Old text | issuerKeyHash | string[0..128] | 1..1 | Required. Hashed value of the issuers public key |
| New text | issuerKeyHash | string[0..128] | 1..1 | Required. **The hash of the DER encoded public key: the value (excluding tag and length) of the subject public key field in the issuer's certificate.** |
| Old text | serialNumber | identifierString[0..40] | 1..1 | Required. The serial number of the certificate. |
| New text | serialNumber | identifierString[0..40] | 1..1 | Required. **The string representation of the hexadecimal value of the serial number without the prefix "0x" and without leading zeroes.** |

## 19.2. Page 380 - (v1) Description of MessageInfoType [481]

The name "Master resource identifier" is confusing. It is just an 'id'.

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **id** | integer | 1..1 | Required. Master resource identifier, unique within an exchange context. It is defined within the OCPP context as a positive Integer value (greater or equal to zero). |
| New text | **id** | integer | 1..1 | Required. **Unique id** within an exchange context. It is defined within the OCPP context as a positive Integer value (greater or equal to zero). |

## 19.3. Page 382 - (v2) Description of OCSPRequestDataType [542]

The descriptions of the fields in OCSPRequestDataType have been improved to better match their definition in IETF RFC 6960).

*Changed descriptions*

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | issuerNameHash | identifierString[0..128] | 1..1 | Required. Hashed value of the Issuer DN (Distinguished Name). |
| New text | issuerNameHash | identifierString[0..128] | 1..1 | Required. **The hash of the issuer’s distinguished name (DN), that must be calculated over the DER encoding of the issuer’s name field in the certificate being checked.** |
| Old text | issuerKeyHash | string[0..128] | 1..1 | Required. Hashed value of the issuers public key |
| New text | issuerKeyHash | string[0..128] | 1..1 | Required. **The hash of the DER encoded public key: the value (excluding tag and length) of the subject public key field in the issuer’s certificate.** |
| Old text | serialNumber | identifierString[0..40] | 1..1 | Required. The serial number of the certificate. |
| New text | serialNumber | identifierString[0..40] | 1..1 | Required. **The string representation of the hexadecimal value of the serial number without the prefix "0x" and without leading zeroes.** |

## 19.4. Page 386 - (v1) SetVariablesRequest can have empty string [488]

It is allowed to set the value of an attribute to an empty string. This is made explicit in the description of SetVariableDataType

| | Field Name | Field Type | Card. | Description |
|---|---|---|---|---|
| Old text | **attributeValue** | string[0..1000] | 1..1 | Required. Value to be assigned to attribute of variable. The Configuration Variable ConfigurationValueSize can be used to limit SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal. |
| New text | **attributeValue** | string[0..1000] | 1..1 | Required. Value to be assigned to attribute of variable.<br>**The value is allowed to be an empty string ("").**<br>The Configuration Variable ConfigurationValueSize can be used to limit SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal. |

# 20. Enumerations

## 20.1. Page 400 - (v1) Wrong description of EventTriggerEnumType Alerting [351]

| | | |
|---|---|---|
| Old text | **Alerting** | Monitored variable has passed an Alert or Critical threshold |
| New text | **Alerting** | Monitored variable has passed a **Lower or Upper Threshold** |

## 20.2. Page 404 - (v1) LocationEnumType Inlet description [494]

The description of **Inlet** has been improved.

| | Value | Description |
|---|---|---|
| Old text | **Inlet** | Measurement at network ("grid") inlet connection. |
| New text | **Inlet** | For the Charging Station (*evseId* = 0): measurement at network ("grid") inlet connection. For measurements with *evseId* > 0, these are measurements taken at the EVSE inlet (This can be useful for a DC charger). |

## 20.3. Page 406 - (v2) Voltage: improved description [544]

The description in the enumeration MeasurandEnumType of the measurand "Voltage" has been enhanced.

| | Value | Description |
|---|---|---|
| Old text | **Voltage** | Instantaneous DC or AC RMS supply voltage |
| New text | **Voltage** | Instantaneous DC or AC RMS supply voltage.<br>For *location* = `Inlet` and *evseId* = 0: voltage at charging station grid connection.<br>For *location* = `Outlet` and *evseId* > 0: voltage at EVSE outlet towards the EV. |

## 20.4. Page 409 - (v1) Enumeration value for OCPP 2.0.1 missing [501]

OCPPVersionEnumType that is used in setNetworkProfile:SetNetworkProfileRequest.NetworkConnectionProfileType does not have a value for OCPP 2.0.1, so `OCPP20` must now be used for OCPP 2.0.1:

| Value | Description |
|---|---|
| **OCPP12** | OCPP version 1.2 |
| **OCPP15** | OCPP version 1.5 |
| **OCPP16** | OCPP version 1.6 |
| **OCPP20** | OCPP version 2.0<br>**The OCPP 2.0 release of OCPP has been deprecated, so this value `OCPP20` must now be used for OCPP 2.0.1 implementations in the NetworkConnectionProfile. Note that OCPP 2.0.1 does have its own Websocket subprotocol name: `ocpp2.0.1`.** |

# 21. Referenced Components and Variables

## 21.1. Page 420 - (v1) List of Components and Variables in XLS format [145]

An Excel sheet with a list of referenced components and their typical or required variables and instance names has been created and will be released together with this errata document.

## 21.2. Page 420 - (v1) Section 2 Referenced Components and Variables: Added reference to part 1

For a proper understanding of this section it is essential that chapter 4 in "Part 1 - Architecture & Topology" about the addressing of Components and Variables has been read.

Therefore, the following text is added, as follows:

| | |
|---|---|
| After this text | A required Configuration Variable mentioned under a particular function block only has to be supported by the Charging Station if it supports that functional block. |
| Add new text | Please see chapter 4 in "Part 1 - Architecture & Topology" about the addressing of Components and Variables in the Device Model. |

## 21.2.1. (Page 446) - Charging Infrastructure Related Variables

Some clarification has been requested about the addressing of EVSEs and Connectors in the device model. The following text is added for that purpose.

### At 2.13

Add new section "Example Reporting of EVSEs and Connectors via device model"

The following example illustrates how the device model reports EVSEs and Connectors for an example charging station that has two EVSEs, of which EVSE #1 has one Type2 connector and EVSE #2 has two connectors: CCS and CHAdeMO.

| **Component** | | | | **Variable** | | **VariableAttribute** | | **VariableCharacteristics** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **name** | **evse id** | **evse connectorId** | **instance** | **name** | **instance** | **type** | **value** | **dataType** | **maxLimit** | **supports Monitoring** |
| ChargingStation | | | | Available | | Actual | true | boolean | | false |
| ChargingStation | | | | AvailabilityState | | Actual | Available | boolean | | false |
| ChargingStation | | | | SupplyPhases | | Actual | integer | 3 | | false |
| ChargingStation | | | | ACCurrent | "L1" | Actual | decimal | 45.0 | | true |
| ChargingStation | | | | ACCurrent | "L2" | Actual | decimal | 44.9 | | true |
| ChargingStation | | | | ACCurrent | "L3" | Actual | decimal | 44.9 | | true |
| EVSE | 1 | | "left" | Available | | Actual | true | boolean | | false |
| EVSE | 1 | | "left" | AvailabilityState | | Actual | Available | optionList | | false |
| EVSE | 1 | | "left" | SupplyPhases | | Actual | 3 | integer | | false |
| EVSE | 1 | | "left" | Power | | Actual | 0.0 | decimal | 22000.0 | true |
| Connector | 1 | 1 | | Available | | Actual | true | boolean | | false |
| Connector | 1 | 1 | | ConnectorType | | Actual | sType2 | string | | false |
| Connector | 1 | 1 | | SupplyPhases | | Actual | 3 | integer | | false |
| EVSE | 2 | | "right" | Available | | Actual | true | boolean | | false |
| EVSE | 2 | | "right" | AvailabilityState | | Actual | Occupied | optionList | | false |
| EVSE | 2 | | "right" | SupplyPhases | | Actual | 0 | integer | | false |
| EVSE | 2 | | "right" | Power | | Actual | 41000.0 | decimal | 50000.0 | true |
| Connector | 2 | 1 | | Available | | Actual | true | boolean | | false |
| Connector | 2 | 1 | | AvailabilityState | | Actual | Occupied | optionList | | false |
| Connector | 2 | 1 | | ConnectorType | | Actual | cCCS2 | string | | false |
| Connector | 2 | 1 | | SupplyPhases | | Actual | 0 | integer | | false |
| Connector | 2 | 2 | | Available | | Actual | true | boolean | | false |
| Connector | 2 | 2 | | AvailabilityState | | Actual | Unavailable | optionList | | false |
| Connector | 2 | 2 | | ConnectorType | | Actual | cG105 | string | | false |
| Connector | 2 | 2 | | SupplyPhases | | Actual | 0 | integer | | false |

**NOTE**

An instance name has been given to the EVSEs in this example. This is to illustrate that it is allowed to provide an instance name even if only one instance of the component exists. It is not required to do so.
The variable Voltage of ChargingStation has been added to show an example of a multi-instance variable.
Not all VariableAttributes and VariableCharacteristics are shown in the table.

### At 2.13.1 `Available`

Add to Note:

"EVSE and Connector components are addressed on their respective tier. So, EVSE #1 is addressed as component EVSE on tier "*evse* = 1" and connector #1 on this EVSE is addressed as component Connector on tier "*evse* = 1, *connector* = 1."

### At 2.13.2 `AvailabilityState`

Add to Note:

"An EVSE component is addressed on its own tier. So, EVSE #1 is addressed as component EVSE on tier "*evse* = 1."

# 21.3. Page 422 - (v1) Improved description of OfflineThreshold [487]

| | | |
|---|---|---|
| Old text | **Description** | When the offline period of a Charging Station exceeds the `OfflineThreshold` it is recommended to send a StatusNotificationRequest for all its Connectors. |
| New text | **Description** | When the offline period of a Charging Station exceeds the `OfflineThreshold` it is recommended to send a StatusNotificationRequest for all its Connectors **when the Charging Station is back online.** |

# 21.4. Page 423 - (v2) Variable UnlockOnEVSideDisconnect does not have an evse attribute

The configuration variable UnlockOnEVSideDisconnect suggests that it can be related to an EVSE. That is wrong. Only components can be associated with an EVSE. The OCPPCommCtrlr, however, cannot be specific for an EVSE, because there is one OCPPCommCtrlr for the entire charging station.

The row "**evse**" has been removed from the table below and the description has been updated.

## 21.4.1. `UnlockOnEVSideDisconnect`

| | | | |
|---|---|---|---|
| **Required** | yes | | |
| **Component** | **componentName** | OCPPCommCtrlr | |
| **Variable** | **variableName** | UnlockOnEVSideDisconnect | |
| | **~~evse~~** | | |
| | **variableAttributes** | **mutability** | ReadWrite/ReadOnly |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | When set to true, the Charging Station SHALL unlock the cable on the Charging Station side when the cable is unplugged at the EV. For an EVSE with **only** fixed cables, the mutability SHALL be ReadOnly and the actual value SHALL be false. **For a charging station with fixed cables and sockets, the variable is only applicable to the sockets.** | | |

# 21.5. Page 432 - (v2) DeviceDataCtrlr.Items/BytesPerMessage also applies to GetMonitoringReport [649]

The component DeviceDataCtrlr has two variable *ItemsPerMessage* and *BytesPerMessage* to limit the maximum size. These variables have instances for "GetReport" and "GetVariables". Requirement N02.FR.07 states that this also applies to GetMonitoringReport, but that is not clear from the description of the configuration variable.

The **Description** field has been updated with the text "or GetMonitoringReportRequest" (marked in boldface).

## 21.5.1. `ItemsPerMessageGetReport`

| | | |
|---|---|---|
| **Required** | yes | |
| **Component** | **componentName** | DeviceDataCtrlr |

| | | | |
|---|---|---|---|
| **Variable** | **variableName** | ItemsPerMessage | |
| | **variableInstance** | GetReport | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | integer |
| **Description** | Maximum number of ComponentVariable entries that can be sent in one GetReportRequest **or** **GetMonitoringReportRequest** message. | | |

### 21.5.2. `BytesPerMessageGetReport`

| | | | |
|---|---|---|---|
| **Required** | yes | | |
| **Component** | **componentName** | DeviceDataCtrlr | |
| **Variable** | **variableName** | BytesPerMessage | |
| | **variableInstance** | GetReport | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | integer |
| **Description** | Message Size (in bytes) - puts constraint on GetReportRequest **or** **GetMonitoringReportRequest** message size. | | |

## 21.6. Page 426 - (v1) Variable ClockCtrlr.TimeAdjustmentReportingThreshold [492]

The optional variable ClockCtrlr.TimeAdjustmentReportingThreshold needs to be mentioned here, because it occurs as a SecurityEvent "SettingSystemTime".

### 21.6.1. `TimeAdjustmentReportingThreshold`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ClockCtrlr | |
| **Variable** | **variableName** | TimeAdjustmentReportingThreshold | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | integer |
| **Description** | When the clock time is adjusted forwards or backwards for more then TimeAdjustmentReportingThreshold number of seconds, a SecurityEventNotification( "SettingSystemTime" ) is sent by the charging station. A reasonable value is 20 seconds. | | |

## 21.7. Page 427 - (v1) Variable SecurityCtrlr.BasicAuthPassword [489]

The type of a `BasicAuthPassword` is a case-sensitive version of identifierString. This new type is now called "passwordString". The change of "identifierString" to "passwordString" is marked in bold in the text below.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | componentName | SecurityCtrlr | |
| **Variable** | variableName | BasicAuthPassword | |
| | variableAttributes | mutability | WriteOnly |
| | variableCharacteristics | dataType | **passwordString** |
| | | maxLimit | 40 (Max length of the BasicAuthPassword) |
| **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 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). 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.<br>This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented. | | |

## 21.8. Page 428 - (v2) Improved description of OrganizationName [393]

| | | |
|---|---|---|
| **Component** | **componentName** | SecurityCtrlr |
| **Variable** | **variableName** | OrganizationName |

The description of SecurityCtrlr.OrganizationName was not clear on what is meant by "specify the subject field". The description is improved as follows.

| | **Description** |
|---|---|
| Old text | This configuration variable is used to set the organization name of the CSO or an organization trusted by the CSO. This organization name is used to specify the subject field in the client certificate. |
| New text | This configuration variable is used to set the organization name of the CSO or an organization trusted by the CSO. **It is used to set the O (*organizationName*) RDN in the subject field of the client certificate. See also A00.FR.509.** |

## 21.9. Page 428 - (v1) Note with AdditionalRootCertificateCheck variable [479]

A second note is added to the description of `AdditionalRootCertificateCheck` to clarify, that although the variable is required to be present for security profiles 2 and 3, the feature is not required to be implemented. The variable can be fixed at a value of *false*.

*Note 2* is added to the description. See text in **bold**.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | SecurityCtrlr | |
| **Variable** | **variableName** | AdditionalRootCertificateCheck | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | […]<br>***Note 2: The statement that the variable is required, means that the configuration variable must be present, but does NOT indicate that the feature must be implemented. This is an optional feature. By setting the value to false, the Charging Station indicates that it does not support this feature, whereas true means that it does support the feature.*** | | |

## 21.10. Page 429 - (v1) Improved description of AuthEnabled [485]

The description of `Enabled` might suggest that the token reader is disabled, but that does not have to be the case. If a token is read, it can still be provided in the transaction event message.

| | | |
|---|---|---|
| Old text | **Description** | If set to FALSE, then authorization is switched off. Transactions are still possible, but no authorization will take place. This implies, that the value of idToken in transaction events SHALL be `NoAuthorization` |
| New text | **Description** | If set to *false*, then no authorization is done before starting a transaction or when reading an idToken. If an idToken was provided, then it will be put in the *idToken* field of the TransactionEventRequest. If no idToken was provided, then *idToken* in TransactionEventRequest will be left empty and *type* is set to `NoAuthorization`. |

## 21.11. Page 430 - (v1) New optional variable DisableRemoteAuthorization [486]

A new optional variable `DisableRemoteAuthorization` tells the Charging Station to not issue any AuthorizationRequests, but only use Authorization Cache and Local Authorization List to determine validity of idTokens.

### 21.11.1. `DisableRemoteAuthorization`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | AuthCtrlr | |
| **Variable** | **variableName** | DisableRemoteAuthorization | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | When set to *true* this instructs the Charging Station to not issue any AuthorizationRequests, but only use Authorization Cache and Local Authorization List to determine validity of idTokens.<br>Note: The difference with `DisablePostAuthorize` is that this variable disables all authorization with CSMS, whereas `DisablePostAuthorize` only disables re-authorization of tokens that are as not-Accepted in the Authorization Cache or Local Authorization List. | | |

## 21.12. Page 431 - (v1) New optional variable DisablePostAuthorize [484]

Both the Authorization Cache and the Local Authorization List have requirements (C10.FR.03, C12.FR.05 and C14.FR.03) that state that the Charging Station shall send an AuthorizeRequest for an idToken that is not valid in the cache or local authorization list.

A new optional configuration variable `DisablePostAuthorize` can be set to *true* to disable this behavior. The variable can be part of AuthCacheCtrlr and LocalAuthListCtrlr. If the variable does not exist, it defaults to *false*, thus leaving the behavior unchanged for implementations that do not have it.

*New configuration variables*

### 21.12.1. Page 431

`AuthCacheDisablePostAuthorize`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | AuthCacheCtrlr | |
| **Variable** | **variableName** | DisablePostAuthorize | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | When set to *true* this variable disables the behavior to request authorization for an idToken that is stored in the cache with a status other than `Accepted`, as stated in C10.FR.03 and C12.FR.05. | | |

### 21.12.2. Page 432

`LocalAuthListDisablePostAuthorize`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | LocalAuthListCtrlr | |
| **Variable** | **variableName** | DisablePostAuthorize | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | When set to *true* this variable disables the behavior to request authorization for an idToken that is stored in the local authorization list with a status other than `Accepted`, as stated in C14.FR.03. | | |

## 21.13. Page 434 - (v1) Better description of TxStartPoint/TxStopPoint [348]

The names for TxStartPoint and TxStopPoint values are the same, but the meaning is exactly the opposite: in TxStartPoint it is about the start of a situation and in TxStopPoint it refers to the end of it. In order to better explain this we provide distinct explanations of TxStartPoint values and TxStopPoint values.

Replace the text of section **2.6.4.1 TxStartStopPoint** values by the following:

### 2.6.4.1 TxStartPoint values

The following table lists the values allowed for the `TxStartPoint` variable. These values represent logical steps or events that (may) occur during a charging session. When such an event occurs, and it is listed in in the `TxStartPoint` variable, then this marks the start of a transaction.

| Value | Description |
|---|---|
| **ParkingBayOccupancy** | An object (probably an EV) is detected in the parking/charging bay. |
| **EVConnected** | Both ends of the Charging Cable have been connected (if this can be detected, else detection of a cable being plugged into the socket), or for wireless charging: initial communication between EVSE and EV is established. |
| **Authorized** | Driver or EV has been authorized, this can also be some form of anonymous authorization like a start button. |
| **PowerPathClosed** | All preconditions for charging have been met, power can flow. This event is the logical AND of `EVConnected` and `Authorized` and should be used if a transaction is supposed to start when EV is connected and authorized. Despite its name, this event is not related to the state of the power relay.<br>Note: There may be situations where `PowerPathClosed` does not imply that charging starts at that moment, e.g. because of delayed charging or a battery that is too hot. |
| **EnergyTransfer** | Energy is being transferred between EV and EVSE. |
| **DataSigned** | The moment when the signed meter value is received from the fiscal meter, that is used in the TransactionEventRequest with *context* = `Transaction.Begin` and *triggerReason* = `SignedDataReceived`. This TxStartPoint might be applicable when legislation exists that only allows a billable transaction to start when the first signed meter value has been received. |

Add the following text after the description of **TxStopPoint** in 2.6.5:

#### 2.6.5.1 TxStopPoint values

The following table lists the values allowed for the `TxStopPoint` variable. These values represent logical steps or events that (may) occur during a charging session. When such an event occurs, and it is listed in in the `TxStopPoint` variable, then this marks the end of a transaction.

The values are the same as for `TxStartPoint`, but in this case the meaning is different, since it refers to the ending of the event, rather than the start. For use with `TxStopPoint` each value should be interpreted as if it had "Not" prefixed to it. See the following table:

| Value | Description |
|---|---|
| **ParkingBayOccupancy** | An object (probably an EV) is **no longer** detected in the parking/charging bay. |
| **EVConnected** | One or both ends of the Charging Cable have been **disconnected** (if this can be detected, else detection of a cable being **unplugged** from the socket), or for wireless charging: communication between EVSE and EV is **lost**. |
| **Authorized** | Driver or EV **is no longer** authorized, this can also be some form of anonymous authorization like a start button. |
| **PowerPathClosed** | All preconditions for charging **are no longer met**, power **cannot** flow. This event is the logical OR of `EVConnected` and `Authorized` and should be used if a transaction is supposed to end when EV is **disconnected** and/or **deauthorized**. It is exactly the same as having the values `EVConnected`, `Authorized` in `TxStopPoint`.<br>Despite its name, this event is not related to the state of the power relay. |
| **EnergyTransfer** | Energy is **not** being transferred between EV and EVSE.<br>This is not recommended to use as a `TxStopPoint`, because it will stop the transaction as soon as EV or EVSE (temporarily) suspend the charging. |
| **DataSigned** | This condition has no meaning as a `TxStopPoint` and should not be used as such. |

## 21.14. Page 435 - (v1) Note to SampledDataSignReadings [526]

The description of the variable SignReadings for SampledDataCtrlr needs extra text to clarify, that some Charging Stations might only be able to sign `Transaction.Begin` and `Transaction.End` meter values.

| | |
|---|---|
| Old text | If set to *true*, the Charging Station SHALL include signed meter values in the TransactionEventRequest to the CSMS. When a Charging Station does not support signed meter values it SHALL NOT report this variable. |
| New text | If set to *true*, the Charging Station SHALL include signed meter values in the TransactionEventRequest to the CSMS. **Some Charging Stations might only be able to sign `Transaction.Begin` and `Transaction.End` meter values.** When a Charging Station does not support signed meter values it SHALL NOT report this variable. |

## 21.15. Page 438 - (v2) Variable AlignedDataSendDuringIdle does not have evse attribute

The description of AlignedDataSendDuringIdle suggests that the variable can be associated with a EVSE. This is wrong. Only its component, AlignedDataCtrlr, can be specific for an EVSE.

In the table below the row "**evse**" has been moved to the Component.

### 21.15.1. `AlignedDataSendDuringIdle`

| | |
|---|---|
| **Required** | no |

|  |  |  |  |
|---|---|---|---|
| **Component** | **componentName** | AlignedDataCtrlr | |
| | **evse**<br>*(moved to component)* | * | |
| **Variable** | **variableName** | SendDuringIdle | |
| | ~~**evse**~~ | | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | If set to *true*, the Charging Station SHALL NOT send clock aligned meter values when a transaction is ongoing. When an EVSE is specified, it SHALL stop sending the clock aligned meter values for this EVSE when it has an ongoing transaction. When no EVSE is specified, it SHALL stop sending the clock aligned meter values when any transaction is ongoing on this Charging Station. | | |

# 21.16. Page 438 - (v2) AlignedDataSignReadings only applies to meter values in TransactionEvent [625]

The signing of meter values is only relevant in the context of a transaction. The signed meter value may even contain transaction-specific information, such as the IdToken. Signed clock-aligned meter values therefore only make sense in a TransactionEventRequest. These measurands are defined by configuration variable `AlignedDataTxEndedMeasurands`.

## 21.16.1. `AlignedDataSignReadings`

*Changed description*

|  |  |  |
|---|---|---|
| Old text | **Description** | If set to *true*, the Charging Station SHALL include signed meter values in the SampledValueType in the MeterValuesRequest to the CSMS. When a Charging Station does not support signed meter values it SHALL NOT report this variable. |
| New text | **Description** | If set to *true*, the Charging Station SHALL include signed meter values in the **SampledValueType in the TransactionEventRequest to the CSMS for those measurands defined in `AlignedDataTxEndedMeasurands`.**<br>When a Charging Station does not support signed meter values it SHALL NOT report this variable. |

# 21.17. Page 439 - (v1) Note to PublicKeyWithSignedMeterValue [460]

The description of the variable PublicKeyWithSignedMeterValue for OCPPCommCtrlr needs an extra note to clarify that the field cannot be omitted, but only left empty.

|  |  |
|---|---|
| Old text | This Configuration Variable can be used to configure whether a public key needs to be sent with a signed meter value. |
| New text | This Configuration Variable can be used to configure whether a public key needs to be sent with a signed meter value. **Note, that the field is required, so it needs to be present as an empty string when the public key is not sent.** |

# 21.18. Page 440 - (v2) Improved description of ChargingProfileMaxStackLevel [551]

## 21.18.1. `ChargingProfileMaxStackLevel`

|  |  |  |  |
|---|---|---|---|
| **Required** | yes | | |
| **Component** | **componentName** | SmartChargingCtrlr | |
| **Variable** | **variableName** | ProfileStackLevel | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | integer |
| **Old Description** | ~~Max StackLevel of a ChargingProfile. The number defined also indicates the max allowed number of installed charging schedules per Charging Profile Purposes.~~ | | |

| | |
|---|---|
| **New Description** | Maximum acceptable value for *stackLevel* in a ChargingProfile. Since the lowest *stackLevel* is 0, this means that if SmartChargingCtrlr.ProfileStackLevel = 1, there can be at most 2 valid charging profiles per Charging Profile Purpose per EVSE. |

## 21.19. Page 442 - (v2) Typo in description of QueueAllMessages [546]

In the description of OCPPCommCtrlr.QueueAllMessage the sentence:

> “…SHALL drop intermediate values first (1st value, 3th value, 5th etc), not start dropping values from the beginning **or and** of the measurements/meter data.”

should be read as:

> “…SHALL drop intermediate values first (1st value, 3th value, 5th etc), not start dropping values from the beginning **or end** of the measurements/meter data.”

## 21.20. Page 445 - (v1) New optional MonitoringCtrlr variables [357]

There is no message to get the currently set MonitoringBase or MonitoringLevel values. The following optional variables can be used to provide this information.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | MonitoringCtrlr | |
| **Variable** | **variableName** | ActiveMonitoringBase | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | OptionList |
| **Description** | Shows the currently used MonitoringBase. Valid values according MonitoringBaseEnumType: `All`, `FactoryDefault`, `HardwiredOnly`. | | |

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | MonitoringCtrlr | |
| **Variable** | **variableName** | ActiveMonitoringLevel | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | integer |
| **Description** | Shows the currently used MonitoringLevel. Valid values are severity levels of SetMonitoringLevelRequest: 0-9. | | |

## 21.21. Page 446 - (v1) CustomImplementationEnabled is located in wrong section [387]

The configuration variable `CustomImplementationEnabled`, that is part of the CustomizationCtrlr, is located in section 2.12 Display Message related.
This is wrong. It should be part of the section 2.1 General.

## 21.22. Page 447 - (v1) Reporting new connector types [478]

When new connector types appear on the market, we cannot immediately update the ConnectorEnumType, because that is defined in the OCPP schema files. However, it is possible to report new connector types in the variable `ConnectorType` of the Connector component, because that is a string value, as shown below:

| | | |
|---|---|---|
| **Required** | yes | |
| **Component** | **componentName** | Connector |
| | **evse** | * |

| | | | |
|---|---|---|---|
| **Variable** | **variableName** | ConnectorType | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | Value of the type of connector as defined by ConnectorEnumType in "Part 2 - Specification" | | |

Since ConnectorEnumType is only used by ReserveNowRequest, it is in most cases not a problem that this enumeration has not yet been extended.

In the description of the variable `ConnectorType` we add (in addition to the connector type enumeration values) three new connector types with the OCPP code that they will get when the ConnectorEnumType is extended in the next OCPP release. This ensures that the relation between the values in the variable `ConnectorType` and the enumeration ConnectorEnumType remains intact.

- GB/T: Chinese DC charging connector. OCPP code: `cGBT`
- ChaoJi: New CHAdeMO connector harmonised with GB/T. OCPP code: `cChaoJi`
- OppCharge: Reverse pantograph. OCPP code: `OppCharge`

*New description for ConnectorType*

| | |
|---|---|
| **Description** | Value of the type of connector as defined by ConnectorEnumType in "Part 2 - Specification" **plus additionally: `cGBT`, `cChaoJi`, `OppCharge`.** |

## 21.23. Page 449 - (v1) New variables for ISO15118Ctrlr about protocol version in use [378]

During the handshake between EV and EVSE as part of setting up an ISO 15118 connection, the EV sends a list of protocol versions that it supports with a priority to tell which versions are preferred. (Priority "1" being the most preferred version.) EV and EVSE agree on a protocol version with the lowest priority number that they both support. The list of supported versions per EV may be interesting information for a CSO to capture. For that purpose two new optional variables are introduced: one that lists the supported versions and one that shows the agreed version.

**WARNING** This has been superseded by Page 449 - (v2) Variables about ISO15118Ctrlr protocol versions moved to ConnectedEV [598]

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ~~ISO15118Ctrlr~~ | |
| **Variable** | **variableName** | ProtocolSupportedByEV | |
| | **variableInstance** | <Priority> | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | A string with the following comma-separated items:<br>“<uri>,<major>,<minor>”.<br>This is information from the supportedAppProtocolReq message from ISO 15118. Each priority is given its own variable instance.<br>Example: "urn:iso:15118:2:2013:MsgDef,2,0" | | |

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ~~ISO15118Ctrlr~~ | |
| **Variable** | **variableName** | ProtocolAgreed | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | A string with the following comma-separated items:<br>“<uri>,<major>,<minor>”.<br>This is the protocol uri and version information that was agreed upon between EV and EVSE in the supportedAppProtocolReq handshake from ISO 15118.<br>Example: "urn:iso:15118:2:2013:MsgDef,2,0" | | |

## 21.24. Page 449 - (v2) Variables about ISO15118Ctrlr protocol versions moved to ConnectedEV [598]

The method to capture ISO 15118 protocol versions as described in Page 449 - (v1) New variables for ISO15118Ctrlr about protocol version in use [378] above is not correct. There can be implementations that have a single ISO 15118 controller to handle multiple EVSEs. The agreed and supported protocol versions can therefore no longer be stored as part of ISO15118Ctrlr component.

The solution is to move these variables to the ConnectedEV component.

The following is added to chapter 2.14 "ISO 15118 Related".

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ConnectedEV | |
| | **evse** | * | |
| **Variable** | **variableName** | ProtocolSupportedByEV | |
| | **variableInstance** | <Priority> | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | A string with the following comma-separated items:<br>“<uri>,<major>,<minor>”.<br>This is information from the supportedAppProtocolReq message from ISO 15118. Each priority is given its own variable instance.<br>Example: "urn:iso:15118:2:2013:MsgDef,2,0" | | |

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ConnectedEV | |
| | **evse** | * | |
| **Variable** | **variableName** | ProtocolAgreed | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | A string with the following comma-separated items:<br>“<uri>,<major>,<minor>”.<br>This is the protocol uri and version information that was agreed upon between EV and EVSE in the supportedAppProtocolReq handshake from ISO 15118.<br>Example: "urn:iso:15118:2:2013:MsgDef,2,0" | | |

## 21.25. Page 449 - (v1) New configuration variables for ISO15118Ctrlr [180, 443]

The following is added to chapter 2.14 "ISO 15118 Related".

### 21.25.1. `ISO15118PnCEnabled`

**WARNING** This has been superseded by erratum *Page 449 - (v2) Variable ISO15118Ctrlr.PnCEnabled description improved and additional variables added for certificate management*.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ISO15118Ctrlr | |
| **Variable** | **variableName** | PnCEnabled | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | If this variable is true, then ISO 15118 plug and charge is enabled. If this variable is false, then the Charging Station won’t initiate ISO 15118 CSRs. | | |

### 21.25.2. `ISO15118CtrlrEvseId`

In ISO 15118 it is possible to define value added services. The VAS uses the TLS connection between the CS and the EV to access services in the internet. For these, it can be necessary to know at which EVSE the vehicle is loading. In OCPP the EVSE ID is an integer whereas in ISO 15118 it is a string. We add the possibility to associate a new variable `EvseId` with the `ISO15118Ctrlr` component that holds the EVSE ID in string format required by ISO 15118.

**WARNING** This has been superseded by Page 449 - (v2) Setting Plug-and-Charge variables [564, 571] below.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ~~ISO15118Ctrlr~~ | |
| | **evse** | * | |
| **Variable** | **variableName** | EvseId | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2. | | |

### 21.25.3. `ISO15118RequestMeteringReceipt`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ISO15118Ctrlr | |
| **Variable** | **variableName** | RequestMeteringReceipt | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | If this variable is *true*, then Charging Station shall request a metering receipt from EV before sending a fiscal meter value to CSMS. | | |

## 21.26. Page 449 - (v2) Variable ISO15118Ctrlr.PnCEnabled description improved and additional variables added for certificate installation

The following is changed/added to chapter 2.14 "ISO 15118 Related".

The description of *ISO15118PnCEnabled* is unclear. The description has been improved and new variables have been added to clearly separate enabling/disabling ISO15118 plug & charge and certificate installation.

### 21.26.1. `ISO15118PnCEnabled`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ISO15118Ctrlr | |
| **Variable** | **variableName** | PnCEnabled | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | **If this variable is *true*, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is enabled.**<br>**If this variable is *false*, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is disabled.** | | |

### 21.26.2. `ISO15118V2GCertificateInstallationEnabled`

| | | |
|---|---|---|
| **Required** | no | |
| **Component** | **componentName** | ISO15118Ctrlr |

| | | | |
|---|---|---|---|
| **Variable** | **variableName** | V2GCertificateInstallationEnabled | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | If this variable is *true*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is enabled.<br>If this variable is *false*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is disabled. | | |

### 21.26.3. `ISO15118ContractCertificateInstallationEnabled`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ISO15118Ctrlr | |
| **Variable** | **variableName** | ContractCertificateInstallationEnabled | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | If this variable is *true*, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is enabled.<br>If this variable is *false*, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is disabled. | | |

## 21.27. Page 449 - (v2) Setting Plug-and-Charge variables [564, 571]

In order to support plug-and-charge the charging station needs the following configuration variables. This is added to chapter 2.14 "ISO 15118 Related".

### 21.27.1. `SeccId`

An ISO 15118 controller (SECC) is assigned an SECC ID. This SECC ID is used as the SECC certificate's commonName. CSMS can set or read the SECC ID as a variable that is part of the `ISO15118Ctrlr` component. This component can be part of the EVSE tier or the toplevel tier of the device model structure.

**WARNING** Since an SECC has its SECC ID for commonName in its certificate it is not possible to have a combined certificate for V2GCertificate and ChargingStationCertificate, as is suggested in a note in M.2.2. "Using ISO 15118 Certificates in OCPP".

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ISO15118Ctrlr | |
| | **evse** | * (optional) | |
| **Variable** | **variableName** | SeccId | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | The name of the SECC in the string format as required by ISO 15118.<br>It is used as the **commonName** (CN) of the SECC leaf certificate.<br>Example: "DE-ICE-S-0003C4D5578786756453309675436-2" | | |

### 21.27.2. `CountryName`

The country name of an SECC is used as the SECC certificate's countryName. It is the ISO 3166-1 (Alpha-2) code of the country in which the charging station resides. CSMS can set or read the countryName as a variable that is part of the `ISO15118Ctrlr` component.

| | | |
|---|---|---|
| **Required** | no | |
| **Component** | **componentName** | ISO15118Ctrlr |
| | **evse** | * (optional) |

| | | | |
|---|---|---|---|
| **Variable** | **variableName** | CountryName | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | The countryName of the SECC in the ISO 3166-1 format.<br>It is used as the **countryName** (C) of the SECC leaf certificate.<br>Example: "DE" | | |

## 21.27.3. `OrganizationName`

The organization name of the CSO that operates the charging station. It is used in an SECC as the SECC certificate's organizationName.

CSMS can set or read the organizationName as a variable that is part of the `ISO15118Ctrlr` component.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | ISO15118Ctrlr | |
| | **evse** | * (optional) | |
| **Variable** | **variableName** | OrganizationName | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | The organizationName of the CSO operating the charging station.<br>It is used as the **organizationName** (O) of the SECC leaf certificate.<br>Example: "John Doe Charging Services Ltd"<br>Note: This value will usually be identical to SecurityCtrlr.OrganizationName, but it does not have to be. | | |

## 21.27.4. `EvseId`

Plug-and-charge requires that the EVSE ID of the EVSE in the format specified by ISO 15118 is known. This value can also be needed for VAS services via ISO 15118. (The VAS uses the TLS connection between the CS and the EV to access services in the internet.)

In OCPP the EVSE ID is an integer whereas in ISO 15118 it is a string. A new variable *ISO15118EvseId* of the EVSE component holds the EVSE ID in the string format required by ISO 15118.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | EVSE | |
| | **evse** | * | |
| **Variable** | **variableName** | ISO15118EvseId | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | string |
| **Description** | The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2.<br>Example: "DE*ICE*E*1234567890*1" | | |

# 22. Appendix 1 Security Events

## 22.1. Page 3 - (v2) Certain security events must be marked as Critical [585]

Requirement A04.FR.01 states that critical security events must be sent to CSMS, but for some security events that are not marked as critical, requirements exist that they must be sent as a SecurityEventNotification. They must therefore be marked as Critical in the list below.

| Security Event | Description | Critical |
|---|---|---|
| InvalidCsmsCertificate | The certificate that the CSMS uses was not valid or could not be verified | **Yes** |

| Security Event | Description | Critical |
|---|---|---|
| InvalidChargingStationCertificate | The certificate sent to the Charging Station using the CertificateSignedRequest message is not a valid certificate | **Yes** |
| InvalidTLSVersion | The TLS version used by the CSMS is lower than 1.2 and is not allowed by the security specification | **Yes** |
| InvalidTLSCipherSuite | The CSMS did only allow connections using TLS cipher suites that are not allowed by the security specification | **Yes** |

## 22.2. Page 3 - (v2) Unclear which security events are mandatory to be implemented and how this relates to Critical events

For now a note will be added that refers to the specification. In the next version of the appendices a column can be added that marks which security events are mandatory.

| | Description |
|---|---|
| Old text | The table below provides a list of security events. Security events that are marked critical should be pushed to the CSMS.<br>This is a non-exhaustive list of security events, when a security event matches the *description* of one of the Security Events in this section, for interoperability reasons, the Security Event from this section SHALL be used, instead of adding a new (proprietary) Security Event. |
| New text | The table below provides a list of security events. **Security events that are implemented SHALL be stored at the security log and** security events that are **implemented and marked as critical SHALL also** be pushed to the CSMS.<br>This is a non-exhaustive list of security events, when a security event matches the *description* of one of the Security Events in this section, for interoperability reasons, the Security Event from this section SHALL be used, instead of adding a new (proprietary) Security Event. **Some security events like; *InvalidCsmsCertificate*, *InvalidChargingStationCertificate*, etc. are mandatory to be implemented. Please refer to Part 2 - Specification for which security events are mandatory to be implemented.** |

# 23. Appendix 3 Standardized Components

## 23.1. Page 8 - (v1) DefaultMessageTimeout missing for OCPPCommCtrlr [482]

The variable `MessageTimeout` is required for OCPPCommCtrlr, but the variable is missing from the list of typical variable for the component.

The following entry is added to the table in 3.1.11 in Appendix 3.

| Variables | Type | Description |
|---|---|---|
| MessageTimeout | integer | Message timeout in seconds. The message timeout setting in a Charging Station can be configured in the messageTimeout field in the NetworkConnectionProfile. |

## 23.2. Page 8 - (v1) Description of NetworkConfigurationPriority for OCPPCommCtrlr [483]

Then description of `NetworkConfigurationPriority` contains a sentence about NTP servers that does not belong there.

| | Variables | Type | Description |
|---|---|---|---|
| Old text | NetworkConfigurationPriority | string | A comma separated ordered list of the priority of the possible Network Connection Profiles. Multiple NTP servers can be configured as backups, etc. If the NTP client supports it, it can also connect to multiple NTP servers simultaneous to get a more reliable time source. Variable instance value is single digit NTP priority (1=highest). |
| New text | NetworkConfigurationPriority | string | A comma separated ordered list of the priority of the possible Network Connection Profiles. |

## 23.3. Page 9 - (v1) BasicAuthPassword of SecurityCtrlr [490]

The description of BasicAuthPassword in SecurityCtrlr is not correct.

| | | | |
|---|---|---|---|
| Old text | BasicAuthPassword | string | Hexadecimal representation of the password that the Charging Station uses to authenticate itself if HTTP Basic authentication is used (20 bytes maximum, represented as a string of up to 40 hexadecimal digits). If certificates are used, this option does not have to be present. |
| New text | BasicAuthPassword | string | The basic authentication password that is used for HTTP Basic Authentication. The datatype is passwordString 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). 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.<br>This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented. |

## 23.4. Page 14 - (v1) New component ConnectedEV

`ConnectedEV` is a component that represents a connected vehicle for which data is received via an ISO 15118 or CHAdeMO interface. The generic information that is received, is represented as variables of `ConnectedEV`. Any protocol-specific information is represented in variables of the `ISO15118Ctrlr` or `CHAdeMOCtrlr` component.

| Description |
|---|
| ConnectedEV is a component that represents a connected vehicle for which data is received via an ISO 15118 or CHAdeMO interface. The generic information that is received, is represented as variables of ConnectedEV. Any protocol-specific information is represented in variables of the ISO15118Ctrlr or CHAdeMOCtrlr component. |

| Variable | Unit | ISO 15118-2 value | CHAdeMO value |
|---|---|---|---|
| Available | boolean | Is true when an EV is connected | |
| ***Vehicle ID:*** | | | |
| VehicleId | string | EVCCID (from SessionSetupReq)<br>*Six bytes, represented as hexbinary encoded string, e.g. "010203040A0B"* | Vehicle ID (H'710 + H'711 + H'712)<br>*Three times 8 bytes, represented as hexbinary encoded string, e.g. "**010203040A0B0C0D**111213141A1B1C1D**212223242A2B2C2D**". A concatenation of **H'710** + H'711 + **H'712**.* |
| ***Voltage and current values:*** | | | |
| ACCurrent.minSet | A | EVMinCurrent | - |
| ACCurrent.maxSet | A | EVMaxCurrent | - |
| ACVoltage.maxSet | V | EVMaxVoltage | - |
| DCCurrent.minSet | A | - | Minimum charge current (H'100.0) |
| DCCurrent.maxSet | A | EVMaximumCurrentLimit | - |
| DCCurrent.target | A | EVTargetCurrent | Charging current request (H'102.3)<br>*If HighCurrentControl is true, use the value from* Charging current request (extended) (H'110.1,2). |
| DCVoltage.minSet | V | - | Minimum battery voltage (H'100.2,3) |
| DCVoltage.maxSet | V | EVMaximumVoltageLimit | Maximum battery voltage (H'100.4,5) |
| DCVoltage.target | V | EVTargetVoltage | Target battery voltage (H'102.1,2) |
| ***Power, energy and time values:*** | | | |
| Power.maxSet | W | EVMaximumPowerLimit | - |
| EnergyImport.maxSet | Wh | EVEnergyCapacity | Total capacity of traction battery * 100 (H'101.5,6) |
| EnergyImport.target | Wh | EVEnergyRequest (DC)<br>EAmount (AC) | - |

| **Variable** | **Unit** | **ISO 15118-2 value** | **CHAdeMO value** |
|---|---|---|---|
| DepartureTime | dateTime | DepartureTime<br>*Provided as seconds since message receipt. Convert to absolute time.* | - |
| RemainingTimeBulk | s | RemainingTimeToBulkSoC | - |
| RemainingTimeFull.maxSet | s | - | Maximum charging time * 60 (H'101.2) |
| RemainingTimeFull.actual | s | RemainingTimeToFullSoc | Estimated charging time * 60 (H'101.3) |
| StateOfChargeBulk | % | BulkSoC | - |
| StateOfCharge.maxSet | % | FullSoC | Charged rate reference constant (H'100.6) |
| StateOfCharge.actual | % | DC_EVStatus.EVRESSSOC | State of charge (H'102.6) |
| ChargingCompleteBulk | boolean | BulkChargingComplete | - |
| ChargingCompleteFull | boolean | ChargingComplete | - |
| ***Status values:*** | | | |
| ChargingState<br>*with a memberlist consisting of the following values:* | | | |
| * *BatteryOvervoltage* | | - | Battery overvoltage (H'102.4.0) |
| * *BatteryUndervoltage* | | - | Battery undervoltage (H'102.4.1) |
| * *ChargingCurrentDeviation* | | FAILED_ChargingCurrentDifferential | Battery current deviation (H'102.4.2) |
| * *BatteryTemperature* | | FAILED_RESSTemperatureInhibit | High battery temperature (H'102.4.3) |
| * *VoltageDeviation* | | FAILED_ChargingVoltageOutOfRange | Battery voltage deviation (H'102.4.4) |
| * *ChargingSystemError* | | FAILED_EVRESSMalfunction | Charging system error (H'102.5.2) |
| * *VehicleShiftPosition* | | FAILED_EVShiftPosition | Vehicle shift position (H'102.5.1) |
| * *VehicleChargingEnabled* | | - | Vehicle charging enabled (H'102.5.0) |
| * *ChargingSystemIncompatibility* | | FAILED_ChargingSystemIncompatibility | - |
| * *ChargerConnectorLockFault* | | FAILED_ChargerConnectorLockFault | - |

## 23.5. Page 14 - (v2) ConnectedEV: new EnergyCapacity variable [553]

In Errata v2 the variable for energy capacity of the EV has been changed from "EnergyImport(maxSet)" to "EnergyCapacity", because the value of EnergyImport may exceed the "maxSet" value (which would be set to the total energy capacity of the EV), when both charging and discharging is taking place during the charging session.

| **Variable** | **Unit** | **ISO 15118-2 value** | **CHAdeMO value** |
|---|---|---|---|
| ***Power, energy and time values:*** | | | |
| ~~EnergyImport.maxSet~~<br>EnergyCapacity | Wh | EVEnergyCapacity | Total capacity of traction battery * 100 (H'101.5,6) |

# 24. Appendix 5 Reason Codes

## 24.1. (v1) MissingDeviceModelInfo [383]

A generic reason code has been added to report that an operation failed, because some information from the Device Model is missing.

This can be used, for example, when Charging Station tries to check the certificate properties of a CertificateSignedRequest, but it does not have a value for `OrganizationName` in its SecurityCtrlr component.

| | | |
|---|---|---|
| MissingDeviceModelInfo | Information needed for operation is missing from Device Model | (generic) |

# 25. List of Referenced Components and Typical Variables

This table provides an overview of components and typical values that are referenced in the specification. A machine-readable form is available in the files dm_components_vars.csv and dm_components_vars.xls, that are part of the OCPP 2.0.1 download set at openchargealliance.org.

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| <generic> | ACCurrent | | no | decimal | A | RMS AC Current (in amperes). For 3-phase circuits, each phase (and optional neutral) is represented by a Variable instance equal to a value of the PhaseEnumType (e.g. L1,N). Unkeyed values reported for a Component declared to be multi-phase are assumed to be an average of all per-phase readings and written values are common per-phase settings. Example(s): ChargingStation: Total AC current consumption (all EVSE's, ancillaries), EVSE: Total current consumed by EVSE: includes losses (AC→DC) and EVSE specific ancillaries (e.g. fans), ElectricalFeed: Inflow AC current on feed |
| <generic> | Active | | no | boolean | | Component is in its non-resting / active state: e.g: On, Engaged, Locked. Some Components may have secondary functions that have corresponding Active Variables with an explicit Variable instance., Note: Monitoring of changes in the Active state of any Component can be specified by setting Delta monitoring on the boolean value with a delta values of 1. Setting/clearing an Active Variable activates/stops the associated functionality, where remotely controllable. Only components that are Available and Enabled can be in the Active state. |
| <generic> | ACVoltage | | no | decimal | V | RMS AC Voltage (in volts). For 3-phase circuits, each phase (and optional neutral) is represented by a Variable instance equal to a value of the PhaseEnumType (e.g. L1,N). Unkeyed values reported for a Component declared to be multi-phase are assumed to be an average ofall per-phase readings and written values are common per-phase settings. Example(s): ElectricalFeed: Input Voltage |
| <generic> | AllowReset | | no | boolean | | Component can be reset. |
| <generic> | Angle | | no | decimal | Deg | Angle(s) relative to normal/design idle position. Multiple Variable instance values may be used to indicate angular position in multiple axes (e.g. Left-Right, Forward-Back). |
| <generic> | Attempts | | no | integer | | Number of attempts (INCLUDING the original attempt) in the last successful or attempted, cycle of operation. Applies typically to self-monitoring motorized electro-mechanical equipment, etc. {Null}: Unknown, 0: Not Attempted/Not allowed, 1: Single attempt/No retries [allowed], 2-N: [up to] N tries [allowed] |
| <generic> | Available | | no | boolean | | The Component exists and is locally configured/wired for use, but might not be (remotely) Enabled. |
| <generic> | Certificate | | no | string | | Digital Certificate (in Base64 encoding) |
| <generic> | Color | | no | string | | Standard 24 bit hexadecimal RGB values. Reg Green Blue color intensity, expressed as standard 24 bit hexadecimal RGB values: 3 00-FF (0-255), in order RRGGBB). E.g. 000000: Black, FF0000: Red, 00FF00: Green, 0000FF: Blue, FFFF00:Yellow, FFFFFF: White, 008000: Medium intensity green. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| <generic> | Complete | | no | boolean | | Component operation cycle has completed. Used only in event notifications, where it is always true. |
| <generic> | ConnectedTime | | no | decimal | s | Time since logical connection established |
| <generic> | Count | | no | integer | | General purpose integer count variable for Component state reporting |
| <generic> | CurrentImbalance | | no | decimal | Percent | Percentage current imbalance in an AC three phase supply. |
| <generic> | DataText | | no | string | | Text associated with a Component, e.g. a Display. |
| <generic> | DateTime | | no | dateTime | | Point in time value, in [RFC3339] datetime format. Time zone optional. |
| <generic> | DCCurrent | | no | decimal | A | DC Current (in amperes). May be an instantaneous measurement, or a period average, depending on context/equipment. |
| <generic> | DCVoltage | | no | decimal | V | DC Voltage (volts). May be an instantaneous measurement, or a period average, depending on context/equipment. |
| <generic> | ECVariant | | no | string | | Production series variants reflecting internal design changes or sub-component substitutions not affecting external functionality. |
| <generic> | Enabled | | no | boolean | | The Component is Enabled for operation. For Available components that cannot be selectively (remotely) enabled / disabled, this value is always true. Note: Available cannot be false of Enabled is true, so during inventory reporting, Enabled=1 also logically states Available=true |
| <generic> | Energy | | no | decimal | Wh, kWh | Energy quantity (in Wh) for reporting/configuring values related to stored energy (i.e. not transferred energy). |
| <generic> | Entries | | no | integer | | General purpose variable for reporting/managing numbers of entries in repetitive data structures. maxLimit characteristic reports maximum possible entries. |
| <generic> | Fallback | | no | boolean | | Component is operating in a fallback, or backup mode. In inventory reports, a Value of 1 for the maxLimit characteristic indicates that the component can enter a fallback state (i.e. a fallback mode is present). |
| <generic> | FanSpeed | | no | decimal | RPM | Fan Speed (in RPM). A value of 0 represents stopped/stalled. An empty value indicates that fan speed cannot be read. |
| <generic> | FirmwareVersion | | no | string | | Version number of firmware. |
| <generic> | Force | | no | decimal | N | Reports (impact) force/ acceleration values (estimates) in one or more directions, in units of Newtons or “g”. Multiple force readings in different (orthogonal) dimensions may be reported using Variable instance values, such as Down, Right, Forward. |
| <generic> | Formats | | no | MemberList | | List of message formats supported by this Charging Station. Possible values: ASCII, HTML, URI, UTF-8. |
| <generic> | Frequency | | no | decimal | Hz | Frequency of AC power, signal, or component operation. |
| <generic> | FuseRating | | no | decimal | A | Current rating of a fuse/breaker. Variable instances keyed by phase identifier (L1/L2/L3/N). |
| <generic> | Height | | no | decimal | m | Height above(+)/below(-) reference level (ground level unless context demands otherwise). |
| <generic> | Humidity | | no | decimal | RH | The relative humidity in %. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| <generic> | Hysteresis | | no | decimal | Percent | Specifies the width of a 'dead band' (as a percentage of the threshold) around the central value of a threshold setting (e.g. MinSet, MaxSet, monitor thresholds) to avoid repeated triggering when the measured quantity lies close to the threshold and is subject to small variations. |
| <generic> | ICCID | | no | string | | ICCID (Integrated Circuit Card IDentifier) of mobile data SIM card. |
| <generic> | Impedance | | no | decimal | Ohm | Impedance: Primary value is real (resistive only) impedance. Where a complex impedance is to be reported, the imaginary part (reactance) must be represented with a separate Variable instance value of 'reactance'. Reactance values are expressed at the (nominal) relevant operating frequency of the Component (e.g. 50/60Hz for mains electricity feed). |
| <generic> | IMSI | | no | string | | IMSI (International Mobile Subscriber Identity) number of mobile data SIM card |
| <generic> | Interval | | no | integer | s | Minimum Interval (in seconds) between (attempted) operations. |
| <generic> | Length | | no | decimal | m | General Purpose linear distance measure. |
| <generic> | Light | | no | decimal | lx | (Ambient) light level. The value is in Lux. |
| <generic> | Manufacturer | | no | string | | Component Manufacturer name |
| <generic> | Message | | no | string | | Specific stored message for display. |
| <generic> | MinimumStatusDuration | | no | integer | s | Minimum duration that a Charging Station or EVSE status is stable before StatusNotificationRequest is sent to the CSMS. |
| <generic> | Mode | | no | string | | Operating mode string from among valid options (communicated by OptionList, etc. during capability/configuration discovery). |
| <generic> | Model | | no | string | | Manufacturer’s Model code/number of Component, including suffixes etc. to identify functional, regional or linguistic variation, but NOT engineering change level internal variation not affecting external behaviour, etc. |
| <generic> | NetworkAddress | | no | string | | Current network address of a Component. |
| <generic> | Operated | | no | boolean | | The Component operated in an instantaneous, transient, or immediately self-resetting pattern. Used only in event notifications, where it is always true. |
| <generic> | OperatingTimes | | no | string | | Recurring operating times in iCalendar RRULE format. |
| <generic> | Overload | | no | boolean | | Component is in Overload state. |
| <generic> | Percent | | no | decimal | Percent | Generic dimensionless value reporting/setting value. |
| <generic> | PhaseRotation | | no | OptionList | | The phase wiring of Component, relative to it’s upstream feed Component/device. This variable describes the phase rotation of a Component relative to its parent Component, using a three letter string consisting of the letters: R, S, T and x. The letter 'R' can be identified as phase 1 (L1), 'S' as phase 2 (L2), 'T' as phase 3 (L3). The lower case 'x' is used to designate a phase that is not connected. An empty string means that phase rotation is not applicable or not known. |
| <generic> | PostChargingTime | | no | decimal | s | Elapsed time in seconds since last substantive energy transfer |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| <generic> | Power | | no | decimal | W, kW | Instantaneous (real) Power (measured/calculated, including power factor for AC). Where a component (e.g. AC to DC Power Converter) has multiple power measurements, the default (unkeyed) instance is “input” power. |
| <generic> | Problem | | no | boolean | | Component itself has a 'Problem' condition that impacts in any significant way on its normal operation. By definition, 'Problem' state includes (logical OR) 'Fault' state. 'Problem' specifically INCLUDES inability to operate that is propagated (up/down/sideways) from any other associated/connected/containing/contained Component. |
| <generic> | Protecting | | no | boolean | | Applies to 'sensor' type Components that have an associated protection capability, whereby they can disconnect power (e.g. using the main PowerContactor) if the sensed quantity is outside preset/configured limits. If Protecting is true, the Component is actively preventing/interrupting charging. |
| <generic> | SerialNumber | | no | string | | Serial number of Component. |
| <generic> | SignalStrength | | no | decimal | dBm | (Radio/Wired/Optical) data signal strength, in ASU (typically 0-31 or 99 for unknown). Or dbmW (typically -140 to -50). |
| <generic> | State | | no | string | | A state code or name identifier string, to allow the internal state of components to be reported and/or controlled |
| <generic> | StateOfCharge | | no | decimal | Percent | Energy Storage Device (e.g. battery) state of charge, expressed as a percentage of nominal design 0-100% operating range. Note: Values below or above 0-100% are possible and represent over discharged/charged states. |
| <generic> | Storage | | no | integer | B | In bytes. Amount of storage occupied. Storage(maxLimit) specifies absolute limit Storage(MaxSet) restricts usage to specified Max, if supported. |
| <generic> | SupplyPhases | | no | integer | | Number of alternating current phases connected/available. 1 or 3 for AC, 0 means DC (no alternating phases). Null value indicates that the number of phases (e.g. in use) is unknown. |
| <generic> | Suspending | | no | boolean | | If Suspending is true, the Component can is currently suspending charging. |
| <generic> | Suspension | | no | boolean | | Applies to 'sensor' type Components that have a charging suspension capability, typically for safety or equipment protection reasons. If Suspension is true, the component can suspend charging when the sensed quantity is outside preset/configured limits. |
| <generic> | Temperature | | no | decimal | Celsius, Fahrenheit | Temperature(s) of component (in Celsius, by default). Components may have multiple indexed temperature sensors. |
| <generic> | Time | | no | dateTime | | Point in time value, in ISO 8601 datetime format. Time zone optional. |
| <generic> | Timeout | | no | decimal | s | Generic timeout value for Component operation (in seconds). |
| <generic> | Tries | | no | integer | | Number of attempts done by a Component. |
| <generic> | Tripped | | no | boolean | | Single-shot device requires explicit intervention to re-prime/activate to normal. |
| <generic> | VendorName | | no | string | | Vendor or manufacturer of component. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| <generic> | VersionDate | | no | dateTime | | Version date of component in [RFC3339] format. |
| <generic> | VersionNumber | | no | string | | Version number of hardware |
| <generic> | VoltageImbalance | | no | decimal | Percent | Percentage voltage imbalance in three phase supply. |
| AlignedDataCtrlr | Available | | no | boolean | | If this variable reports a value of true, Clock-Aligned Data is supported. |
| AlignedDataCtrlr | Enabled | | no | boolean | | If this variable reports a value of true, Clock-Aligned Data is enabled |
| AlignedDataCtrlr | Interval | | yes | integer | s | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the MeterValuesRequest message. |
| AlignedDataCtrlr | Measurands | | yes | MemberList | | Clock-aligned measurand(s) to be included in MeterValuesRequest, every AlignedDataInterval seconds. |
| AlignedDataCtrlr | SendDuringIdle | | no | boolean | | If set to true, the Charging Station SHALL NOT send clock aligned meter values when a transaction is ongoing. |
| AlignedDataCtrlr | SignReadings | | no | boolean | | If set to true, the Charging Station SHALL include signed meter values in the SampledValueType in the MeterValuesRequest to the CSMS. |
| AlignedDataCtrlr | TxEndedInterval | | yes | integer | s | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. |
| AlignedDataCtrlr | TxEndedMeasurands | | yes | MemberList | | Clock-aligned measurands to be included in the meterValues element of TransactionEventRequest (eventType = Ended), every SampledDataTxEndedInterval seconds from the start of the transaction. |
| AuthCacheCtrlr | Available | | no | boolean | | Authorization caching is available, but not necessarily enabled. |
| AuthCacheCtrlr | Enabled | | no | boolean | | If set to true, Authorizaiton caching is enabled. |
| AuthCacheCtrlr | LifeTime | | no | integer | | Indicates how long it takes until a token expires in the authorization cache since it is last used |
| AuthCacheCtrlr | Policy | | no | OptionList | | Cache Entry Replacement Policy: least recently used, least frequently used, first in first out, other custom mechanism. |
| AuthCacheCtrlr | Storage | | no | integer | B | Indicates the number of bytes currently used by the Authorization Cache. MaxLimit indicates the maximum number of bytes that can be used by the Authorization Cache. |
| AuthCacheCtrlr | DisablePostAuthorize | | no | boolean | | When set to true this variable disables the behavior to request authorization for an idToken that is stored in the cache with a status other than Accepted, as stated in C10.FR.03 and C12.FR.05. |
| AuthCtrlr | AdditionalInfoItemsPerMessage | | no | integer | | Maximum number of AdditionalInfo items that can be sent in one message. |
| AuthCtrlr | AuthorizeRemoteStart | | yes | boolean | | Whether a remote request to start a transaction in the form of RequestStartTransactionRequest message should be authorized beforehand like a local action to start a transaction. |
| AuthCtrlr | Enabled | | no | boolean | | If set to false, then no authorization is done before starting a transaction or when reading an idToken. If an idToken was provided, then it will be put in the idToken field of the TransactionEventRequest. If no idToken was provided, then idToken in TransactionEventRequest will be left empty and type is set to NoAuthorization. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| AuthCtrlr | LocalAuthorizeOffline | | yes | boolean | | Whether the Charging Station, when Offline, will start a transaction for locally-authorized identifiers |
| AuthCtrlr | LocalPreAuthorize | | yes | boolean | | Whether the Charging Station, when online, will start a transaction for locally-authorized identifiers without waiting for or requesting an AuthorizeResponse from the CSMS. |
| AuthCtrlr | MasterPassGroupId | | no | string | | IdTokens that have this id as groupId belong to the Master Pass Group. Meaning they can stop any ongoing transaction, but cannot start transactions. |
| AuthCtrlr | OfflineTxForUnknownIdEnabled | | no | boolean | | Support for unknown offline transactions. |
| AuthCtrlr | DisableRemoteAuthorization | | no | boolean | | When set to true this instructs the Charging Station to not issue any AuthorizationRequests, but only use Authorization Cache and Local Authorization List to determine validity of idTokens. |
| CHAdeMOCtrlr | SelftestActive | | no | boolean | | Self-test is active or self-test is started by setting to true. |
| CHAdeMOCtrlr | CHAdeMOProtocolNumber | | no | integer | | CHAdeMO protocol number (H'102.0) |
| CHAdeMOCtrlr | VehicleStatus | | no | boolean | | Vehicle status (H'102.5.3) |
| CHAdeMOCtrlr | DynamicControl | | no | boolean | | Vehicle is compatible with dynamic control (H'110.0.0) |
| CHAdeMOCtrlr | HighCurrentControl | | no | boolean | | Vehicle is compatible with high current control (H'110.0.1) |
| CHAdeMOCtrlr | HighVoltageControl | | no | boolean | | Vehicle is compatible with high voltage control (H'110.1.2) |
| CHAdeMOCtrlr | AutoManufacturerCode | | no | integer | | Auto manufacturer code (H'700.0) |
| ChargingStation | AllowNewSessionsPendingFirmwareUpdate | | no | boolean | | Indicates whether new sessions can be started on EVSEs, while Charging Station is waiting for all EVSEs to become Available in order to start a pending firmware update |
| ChargingStation | AvailabilityState | | yes | OptionList | | This variable reports current availability state for the ChargingStation |
| ChargingStation | Available | | yes | boolean | | Component exists |
| ChargingStation | Model | | no | string | | Charging station model as reported in BootNotification. |
| ChargingStation | SupplyPhases | | yes | integer | | Number of alternating current phases connected/available. |
| ChargingStation | VendorName | | no | string | | Charging station vendor name as reported in BootNotification. |
| ClockCtrlr | DateTime | | yes | dateTime | | Contains the current date and time |
| ClockCtrlr | NextTimeOffsetTransitionDateTime | | no | dateTime | | Date time of the next time offset transition. |
| ClockCtrlr | NtpServerUri | | no | string | | This contains the address of the NTP server. |
| ClockCtrlr | NtpSource | | no | OptionList | | When an NTP client is implemented, this variable can be used to configure the client |
| ClockCtrlr | TimeAdjustmentReportingThreshold | | no | integer | | If set, then time adjustments with an absolute value in seconds larger than this need to be reported as a security event SettingSystemTime |
| ClockCtrlr | TimeOffset | | no | string | | A Time Offset with respect to Coordinated Universal Time (aka UTC or Greenwich Mean Time) in the form of an [RFC3339] time (zone) offset suffix, including the mandatory “+” or “-“ prefix. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| ClockCtrlr | TimeSource |  | yes | SequenceList |  | Via this variable, the Charging Station provides the CSMS with the option to configure multiple clock sources |
| ClockCtrlr | TimeZone |  | no | string |  | Configured current local time zone in the format: "Europe/Oslo", "Asia/Singapore" etc. For display purposes. |
| ConnectedEV | ProtocolAgreed |  | no | string |  | Information about uri and version that was agreed upon between EV and EVSE in the supportedAppProtocolReq message from ISO 15118. Example: urn:iso:15118:2:2013:MsgDef,2,0 |
| ConnectedEV | ProtocolSupportedByEV | <Priority> | no | string |  | Information from the supportedAppProtocolReq message from ISO 15118. Each priority is given its own variable instance. Example: urn:iso:15118:2:2013:MsgDef,2,0 |
| ConnectedEV | VehicleID |  | no | string |  | EVCCID (from ISO 15118 SessionSetupReq) |
| Connector | AvailabilityState |  | no | OptionList |  | This variable reports current availability state for the Connector. Optional, because already reported in StatusNotification. |
| Connector | Available |  | yes | boolean |  | Component exists |
| Connector | ChargeProtocol |  | no | string |  | The Charging Control Protocol applicable to a Connector. CHAdeMO: CHAdeMO protocol, ISO15118: ISO15118 V2G protocol (wired or wireless) as used with CCS, CPPWM: IEC61851-1 / SAE J1772 protocol (ELV DC & PWM signalling via Control Pilot wire), Uncontrolled: No charging power management applies (e.g. Schuko socket), Undetermined: Yet to be determined (e.g. before plugged in), Unknown: Not determinable, NOTE: ChargeProtocol is distinct from and orthogonal to connectorType. |
| Connector | ConnectorType |  | yes | string |  | A value of ConnectorEnumType (See part 2) plus additionally: cGBT, cChaoJi, OppCharge. Specific type of connector, including sub-variant information. Note: Distinct and orthogonal to Charging Protocol, Power Type, Phases. |
| Connector | SupplyPhases |  | yes | integer |  | Number of alternating current phases connected/available. |
| Controller | MaxMsgElements |  | no | integer |  | Array of implementation-defined limits to the number of elements of specific type that the Charging Station can accept in one message. |
| CPPWMController | State |  | no | string |  | IEC 61851-1 states ("A" to "E") |
| CustomizationCtrlr | CustomImplementationEnabled | <vendorId> | no | boolean |  | Custom implementation <vendorId> has been enabled. |
| DeviceDataCtrlr | BytesPerMessage | GetReport | yes | integer |  | Maximum number of bytes in a message related to instance name: GetReport, GetVariables, SetVariables |
| DeviceDataCtrlr | BytesPerMessage | GetVariables | yes | integer |  | Maximum number of bytes in a message related to instance name: GetReport, GetVariables, SetVariables |
| DeviceDataCtrlr | BytesPerMessage | SetVariables | yes | integer |  | Maximum number of bytes in a message related to instance name: GetReport, GetVariables, SetVariables |
| DeviceDataCtrlr | ConfigurationValueSize |  | no | integer |  | The limit to the following fields: SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal. |
| DeviceDataCtrlr | ItemsPerMessage | GetReport | yes | integer |  | Maximum number of ComponentVariable entries in message related to the instance name: GetReport, GetVariables, SetVariables |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| DeviceDataCtrlr | ItemsPerMessage | GetVariables | yes | integer | | Maximum number of ComponentVariable entries in message related to the instance name: GetReport, GetVariables, SetVariables |
| DeviceDataCtrlr | ItemsPerMessage | SetVariables | yes | integer | | Maximum number of ComponentVariable entries in message related to the instance name: GetReport, GetVariables, SetVariables |
| DeviceDataCtrlr | ReportingValueSize | | no | integer | | The limit to the following fields: GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal. |
| DeviceDataCtrlr | ValueSize | | no | integer | | Can be used to limit the following fields: SetVariableData.attributeValue, GetVariableResult.attributeValue, VariableAttribute.value, VariableCharacteristics.valueList and EventData.actualValue. |
| DisplayMessageCtrlr | Available | | no | boolean | | Whether display messages are supported. |
| DisplayMessageCtrlr | DisplayMessages | | yes | integer | | Amount of different messages that are currently configured in this Charging Station, via SetDisplayMessageRequest |
| DisplayMessageCtrlr | Enabled | | no | boolean | | Whether display messages are enabled. |
| DisplayMessageCtrlr | PersonalMessageSize | | no | integer | | Max size (in characters) of the personal message element of the IdTokenInfo data (0 specifies no personal data may be stored). |
| DisplayMessageCtrlr | SupportedFormats | | yes | MemberList | | List of message formats supported by this Charging Station. |
| DisplayMessageCtrlr | SupportedPriorities | | yes | MemberList | | List of the priorities supported by this Charging Station. |
| DistributionPanel | ChargingStation | | no | string | | Identity of charging station connected to the distribution panel. |
| DistributionPanel | DistributionPanel | | no | string | | List of Distribution Panels InstanceNames connected to this LocalController. |
| DistributionPanel | Fuse | <n> | no | integer | A | Fuse (index n) is the fuse for phase Ln in Ampere |
| EVSE | AllowReset | | no | boolean | | Can be used to announce that an EVSE can be reset individually |
| EVSE | AvailabilityState | | yes | OptionList | | This variable reports current availability state for the EVSE |
| EVSE | Available | | yes | boolean | | Component exists |
| EVSE | EvseId | | no | string | | The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2. |
| EVSE | Power | | yes | decimal | W, kW | The variableCharacteristic maxLimit, that holds the maximum power that this EVSE can provide, is required. The Actual value of the instantaneous (real) power is desired, but not required. |
| EVSE | SupplyPhases | | yes | integer | | Number of alternating current phases connected/available. |
| EVSE | ISO15118EvseId | | no | string | | The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2. Example: "DE*ICE*E*1234567890*1" |
| FiscalMetering | EnergyExport | | no | decimal | Wh, kWh | Total energy transferred: e.g. from EV during (ongoing or terminated) charging session (in wH by default) |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| FiscalMetering | EnergyExportRegister | | no | decimal | Wh, kWh | Cumulative export kWh register value, such as from a (certified) fiscal energy meter. |
| FiscalMetering | EnergyImport | | no | decimal | Wh, kWh | Total energy transferred. |
| FiscalMetering | EnergyImportRegister | | no | decimal | Wh, kWh | Cumulative export kWh register value, such as from a (certified) fiscal energy meter. |
| ISO15118Ctrlr | CentralContractValidationAllowed | | no | boolean | | If this variable exists and has the value true, then Charging Station can provide a contract certificate that it cannot validate, to the CSMS for validation as part of the AuthorizeRequest. |
| ISO15118Ctrlr | ContractValidationOffline | | yes | boolean | | If this variable is true, then Charging Station will try to validate a contract certificate when it is offline |
| ISO15118Ctrlr | SeccId | | no | string | | The ID of the SECC in string format as defined by ISO15118. |
| ISO15118Ctrlr | MaxScheduleEntries | | no | integer | | Maximum number of allowed schedule periods. |
| ISO15118Ctrlr | RequestedEnergyTransferMode | | no | OptionList | | The requested energy transfer mode. |
| ISO15118Ctrlr | RequestMeteringReceipt | | no | boolean | | If true, then Charging Station shall request a metering receipt from EV. |
| ISO15118Ctrlr | CountryName | | no | string | | The countryName of the SECC in the ISO 3166-1 format. It is used as the countryName © of the SECC leaf certificate. Example: "DE" |
| ISO15118Ctrlr | OrganizationName | | no | string | | The organizationName of the CSO operating the charging station. It is used as the organizationName (O) of the SECC leaf certificate. Example: "John Doe Charging Services Ltd" Note: This value will usually be identical to SecurityCtrlr.OrganizationName, but it does not have to be. |
| ISO15118Ctrlr | PnCEnabled | | no | boolean | | If this variable is true, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is enabled. If this variable is false, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is disabled. |
| ISO15118Ctrlr | V2GCertificateInstallationEnabled | | no | boolean | | If this variable is true, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is enabled. If this variable is false, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is disabled. |
| ISO15118Ctrlr | ContractCertificateInstallationEnabled | | no | boolean | | If this variable is true, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is enabled. If this variable is false, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is disabled. |
| LocalAuthListCtrlr | Available | | no | boolean | | Local Authorization List is available. |
| LocalAuthListCtrlr | BytesPerMessage | | yes | integer | | Maximum number of bytes in a SendLocalList message. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| LocalAuthListCtrlr | Enabled | | no | boolean | | If this variable exists and reports a value of true, Local Authorization List is enabled. |
| LocalAuthListCtrlr | Entries | | yes | integer | | Amount of IdTokens currently in the Local Authorization List |
| LocalAuthListCtrlr | ItemsPerMessage | | yes | integer | | Maximum number of records in SendLocalList |
| LocalAuthListCtrlr | Storage | | no | integer | B | Indicates the number of bytes currently used by the Local Authorization List. MaxLimit indicates the maximum number of bytes that can be used by the Local Authorization List. |
| LocalAuthListCtrlr | DisablePostAuthorize | | no | boolean | | When set to true this variable disables the behavior to request authorization for an idToken that is stored in the local authorization list with a status other than Accepted, as stated in C14.FR.03. |
| LocalEnergyStorage | Capacity | | no | decimal | Wh | Maximum storage capacity |
| MonitoringCtrlr | Available | | no | boolean | | Whether monitoring is available |
| MonitoringCtrlr | BytesPerMessage | ClearVariableMonitoring | no | integer | | Maximum number of bytes in a ClearVariableMonitoring message. |
| MonitoringCtrlr | BytesPerMessage | SetVariableMonitoring | yes | integer | | Maximum number of bytes in a SetVariableMonitoring message |
| MonitoringCtrlr | Enabled | | no | boolean | | Whether monitoring is enabled. |
| MonitoringCtrlr | ItemsPerMessage | ClearVariableMonitoring | no | integer | | Maximum number of IDs in a ClearVariableMonitoringRequest. |
| MonitoringCtrlr | ItemsPerMessage | SetVariableMonitoring | yes | integer | | Maximum number of setMonitoringData elements that can be sent in one setVariableMonitoringRequest message. |
| MonitoringCtrlr | OfflineQueuingSeverity | | no | integer | | When set and the Charging Station is offline, the Charging Station shall queue any notifyEventRequest messages triggered by a monitor with a severity number equal to or lower than the severity configured here. |
| MonitoringCtrlr | MonitoringBase | | no | OptionList | | Currently used monitoring base (readonly) |
| MonitoringCtrlr | MonitoringLevel | | no | integer | | Currently used monitoring level (readonly) |
| MonitoringCtrlr | ActiveMonitoringBase | | no | OptionList | | Shows the currently used MonitoringBase. Valid values according MonitoringBaseEnumType: All, FactoryDefault, HardwiredOnly. |
| MonitoringCtrlr | ActiveMonitoringLevel | | no | integer | | Shows the currently used MonitoringLevel. Valid values are severity levels of SetMonitoringLevelRequest: 0-9. |
| OCPPCommCtrlr | ActiveNetworkProfile | | no | string | | Indicates the configuration profile the station uses at that moment to connect to the network. |
| OCPPCommCtrlr | FileTransferProtocols | | yes | MemberList | | List of supported file transfer protocols |
| OCPPCommCtrlr | HeartbeatInterval | | no | integer | s | Interval of inactivity (no OCPP exchanges) with CSMS after which the Charging Station should send HeartbeatRequest. |
| OCPPCommCtrlr | MessageTimeout | Default | yes | integer | s | MessageTimeout(Default) specifies after which time a message times out. It is configured in the network connection profile. |
| OCPPCommCtrlr | MessageAttemptInterval | TransactionEvent | yes | integer | | MessageAttemptInterval(TransactionEvent) specifies long the Charging Station should wait before resubmitting a TransactionEventRequest message that the CSMS failed to process. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| OCPPCommCtrlr | MessageAttempts | TransactionEvent | yes | integer | | MessageAttempts(TransactionEvent) specifies how often the Charging Station should try to submit a TransactionEventRequest message when the CSMS fails to process it. |
| OCPPCommCtrlr | NetworkConfigurationPriority | | yes | string | | A comma separated ordered list of the priority of the possible Network Connection Profiles. |
| OCPPCommCtrlr | NetworkProfileConnectionAttempts | | yes | integer | | Specifies the number of connection attempts the Charging Station executes before switching to a different profile. |
| OCPPCommCtrlr | OfflineThreshold | | yes | integer | s | When the offline period of a Charging Station exceeds the OfflineThreshold it is recommended to send a StatusNotificationRequest for all its Connectors. |
| OCPPCommCtrlr | PublicKeyWithSignedMeterValue | | no | OptionList | | This Configuration Variable can be used to configure whether a public key needs to be sent with a signed meter value. Note, that the field is required, so it needs to be present as an empty string when the public key is not sent. |
| OCPPCommCtrlr | QueueAllMessages | | no | boolean | | When this variable is set to true, the Charging Station will queue all message until they are delivered to the CSMS. |
| OCPPCommCtrlr | ResetRetries | | yes | integer | | Number of times to retry a reset of the Charging Station when a reset was unsuccessful |
| OCPPCommCtrlr | RetryBackOffRandomRange | | no | integer | | When the Charging Station is reconnecting, after a connection loss, it will use this variable as the maximum value for the random part of the back-off time |
| OCPPCommCtrlr | RetryBackOffRepeatTimes | | no | integer | | When the Charging Station is reconnecting, after a connection loss, it will use this variable for the amount of times it will double the previous back-off time. |
| OCPPCommCtrlr | RetryBackOffWaitMinimum | | no | integer | | When the Charging Station is reconnecting, after a connection loss, it will use this variable as the minimum back-off time, the first time it tries to reconnect. |
| OCPPCommCtrlr | UnlockOnEVSideDisconnect | | yes | boolean | | When set to true, the Charging Station SHALL unlock the cable on the Charging Station side when the cable is unplugged at the EV. For an EVSE with only fixed cables, the mutability SHALL be ReadOnly and the actual value SHALL be false. For a charging station with fixed cables and sockets, the variable is only applicable to the sockets. |
| OCPPCommCtrlr | WebSocketPingInterval | | no | integer | s | 0 disables client side websocket Ping/Pong. In this case there is either no ping/pong or the server initiates the ping and client responds with Pong. Positive values are interpreted as number of seconds between pings. Negative values are not allowed. |
| OCPPCommCtrlr | FieldLength | | no | integer | | This variable is used to report the length of <field> in <message> when it is larger than the length that is defined in the standard OCPP message schema. |
| ReservationCtrlr | Available | | no | boolean | | Whether reservation is supported. |
| ReservationCtrlr | Enabled | | no | boolean | | Whether reservation is enabled. |
| ReservationCtrlr | NonEvseSpecific | | no | boolean | | If this configuration variable is present and set to true: Charging Station supports Reservation where EVSE id is not specified. |
| SampledDataCtrlr | Available | | no | boolean | | If this variable reports a value of true, Sampled Data is supported |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| SampledDataCtrlr | Enabled | | no | boolean | | If this variable reports a value of true, Sampled Data is enabled. |
| SampledDataCtrlr | SignReadings | | no | boolean | | If set to true, the Charging Station SHALL include signed meter values in the TransactionEventRequest to the CSMS |
| SampledDataCtrlr | TxEndedInterval | | yes | integer | s | Interval between sampling of metering (or other) data, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. |
| SampledDataCtrlr | TxEndedMeasurands | | yes | MemberList | | Sampled measurands to be included in the meterValues element of TransactionEventRequest (eventType = Ended), every SampledDataTxEndedInterval seconds from the start of the transaction. |
| SampledDataCtrlr | TxStartedMeasurands | | yes | MemberList | | Sampled measurand(s) to be taken at the start of any transaction to be included in the meterValues field of the first TransactionEventRequest message send at the start of a transaction (eventType = Started) |
| SampledDataCtrlr | TxUpdatedInterval | | yes | integer | s | Interval between sampling of metering (or other) data, intended to be transmitted via TransactionEventRequest (eventType = Updated) messages |
| SampledDataCtrlr | TxUpdatedMeasurands | | yes | MemberList | | Sampled measurands to be included in the meterValues element of TransactionEventRequest (eventType = Ended) |
| SampledDataCtrlr | RegisterValuesWithoutPhases | | no | boolean | | If this variable reports a value of true, then meter values of measurand Energy.Active.Import.Register will only report the total energy over all phases without reporting the individual phase values. If this variable is absent or false, then the value for each phase is reported, possibly also with a total value (depending on the meter). |
| SecurityCtrlr | AdditionalRootCertificateCheck | | no | boolean | | Required for all security profiles except profile 1. |
| SecurityCtrlr | BasicAuthPassword | | no | passwordString | | The basic authentication password is used for HTTP Basic Authentication. |
| SecurityCtrlr | CertificateEntries | | yes | integer | | Amount of Certificates currently installed on the Charging Station |
| SecurityCtrlr | CertSigningRepeatTimes | | no | integer | | Number of times to resend a SignCertificateRequest when CSMS does nor return a signed certificate. |
| SecurityCtrlr | CertSigningWaitMinimum | | no | integer | s | Seconds to wait before generating another CSR in case CSMS does not return a signed certificate. |
| SecurityCtrlr | Identity | | no | identifierString | | The Charging Station identity. |
| SecurityCtrlr | MaxCertificateChainSize | | no | integer | | Limit of the size of the 'certificateChain' field from the CertificateSignedRequest |
| SecurityCtrlr | OrganizationName | | yes | string | | The organization name of the CSO or an organization trusted by the CSO. This organization name is used to specify the subject field in the client certificate. |
| SecurityCtrlr | SecurityProfile | | yes | integer | | The security profile used by the Charging Station. |
| SmartChargingCtrlr | ACPhaseSwitchingSupported | | no | boolean | | This variable can be used to indicate an on-load/in-transaction capability. If defined and true, this EVSE supports the selection of which phase to use for 1 phase AC charging. |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| SmartChargingCtrlr | Available | | no | boolean | | Whether smart charging is supported. |
| SmartChargingCtrlr | Enabled | | no | boolean | | Whether smart charging is enabled. |
| SmartChargingCtrlr | Entries | ChargingProfiles | yes | integer | | Entries(ChargingProfiles) is the amount of Charging profiles currently installed on the Charging Station |
| SmartChargingCtrlr | ExternalControlSignalsEnabled | | no | boolean | | Indicates whether a Charging Station should respond to external control signals that influence charging. |
| SmartChargingCtrlr | LimitChangeSignificance | | yes | decimal | Percent | If at the Charging Station side a change in the limit in a ChargingProfile is lower than this percentage, the Charging Station MAY skip sending a NotifyChargingLimitRequest or a TransactionEventRequest message to the CSMS. |
| SmartChargingCtrlr | NotifyChargingLimitWithSchedules | | no | boolean | | Indicates if the Charging Station should include the externally set charging limit/schedule in the message when it sends a NotifyChargingLimitRequest message. |
| SmartChargingCtrlr | PeriodsPerSchedule | | yes | integer | | Maximum number of periods that may be defined per ChargingSchedule. |
| SmartChargingCtrlr | Phases3to1 | | no | boolean | | If defined and true, this Charging Station supports switching from 3 to 1 phase during a transaction |
| SmartChargingCtrlr | ProfileStackLevel | | yes | integer | | Maximum acceptable value for stackLevel in a ChargingProfile. Since the lowest stackLevel is 0, this means that if SmartChargingCtrlr.ProfileStackLevel = 1, there can be at most 2 valid charging profiles per Charging Profile Purpose per EVSE. |
| SmartChargingCtrlr | RateUnit | | yes | MemberList | | A list of supported quantities for use in a ChargingSchedule. Allowed values: 'A' and 'W |
| TariffCostCtrlr | Available | Tariff | no | boolean | | Instance Tariff: Whether tariffs are supported. |
| TariffCostCtrlr | Available | Cost | no | boolean | | Instance Cost: Wheter costs are supported. |
| TariffCostCtrlr | Currency | | yes | string | | Currency used by this Charging Station in a ISO 4217 [ISO4217] formatted currency code. |
| TariffCostCtrlr | Enabled | Tariff | no | boolean | | Instance Tariff: Whether tariffs are enabled. |
| TariffCostCtrlr | Enabled | Cost | no | boolean | | Instance Cost: Wheter costs are enabled. |
| TariffCostCtrlr | TariffFallbackMessage | | yes | string | | Message (and/or tariff information) to be shown to an EV Driver when there is no driver specific tariff information available. |
| TariffCostCtrlr | TotalCostFallbackMessage | | yes | string | | Message to be shown to an EV Driver when the Charging Station cannot retrieve the cost for a transaction at the end of the transaction. |
| TokenReader | Token | | no | string | | String of bytes representing an ID token. |
| TokenReader | TokenType | | no | OptionList | | Type of Token. Value is one of IdTokenEnumType. |
| TxCtrlr | ChargingTime | | no | decimal | s | Time from earliest to latest substantive energy transfer |
| TxCtrlr | EVConnectionTimeOut | | yes | integer | s | Interval from between "starting" of a transaction until incipient transaction is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). |

| Specific Component | Variable | Instance | Required? | Data Type | Unit | Description |
|---|---|---|---|---|---|---|
| TxCtrlr | MaxEnergyOnInvalidId | | no | integer | | Maximum amount of energy in Wh delivered when an identifier is deauthorized by the CSMS after start of a transaction. |
| TxCtrlr | StopTxOnEVSideDisconnect | | yes | boolean | | When set to true, the Charging Station SHALL deauthorize the transaction when the cable is unplugged from the EV. |
| TxCtrlr | StopTxOnInvalidId | | yes | boolean | | Whether the Charging Station will deauthorize an ongoing transaction when it receives a non-Accepted authorization status in TransactionEventResponse for this transaction. |
| TxCtrlr | TxBeforeAcceptedEnabled | | no | boolean | | Allow charging before having received a BootNotificationResponse with RegistrationStatus: Accepted. |
| TxCtrlr | TxStartPoint | | yes | MemberList | | Defines when the Charging Station starts a new transaction |
| TxCtrlr | TxStopPoint | | yes | MemberList | | Defines when the Charging Station ends a transaction |