OCPP 2.0.1

Part 2 Edition 2 - Errata

v1.0, 2023-06-30

Table of Contents

Version History

Version	Date	Description
1.0	2023-06-30	Errata OCPP 2.0.1 edition 2

1. Scope

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

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.

1.1. 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 is version 1 of the errata. The errata of this version are marked with "(v1)" in the section title. Whenever a second version of the errata is released, then its section titles will be 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]".

2. General

3. Use case A Security

3.1. Page 23 - (v1) Requirement A00.FR.316: Make clear that InvalidTLSVersion must be queued [689]

Requirement A00.FR.316 states that a security event InvalidTLSVersion is triggered and connection must be closed. It is not clear from this requirement that this must also be sent as a security event notification when a connection to CSMS is made. This is stated in use case A04. Obviously, once CSMS accepts the connection, the InvalidTLSVersion condition no longer applies at this time, but the event must still be reported.

Changed requirement

	ID	Precondition	Requirement definition
Old text	A00.FR.316	A00.FR.314 AND The Charging Station detects that the CSMS only allows connections using an older version of TLS, or only allows SSL	The Charging Station SHALL trigger an InvalidTLSVersion security event AND terminate the connection (See part 2 appendices for the full list of security events).
New text	A00.FR.316	A00.FR.314 AND The Charging Station detects that the CSMS only allows connections using an older version of TLS, or only allows SSL	The Charging Station SHALL trigger an InvalidTLSVersion security event AND terminate the connection (See part 2 appendices for the full list of security events). **NOTE: This is a critical security event that will need to be queued and sent to CSMS once a successful connection has been made, as described in use case A04.** **A security event only needs to be sent once for repeated failed connection attempts, in order to avoid overflow to the offline queue.**

3.1.1. Page 25 - Requirement A00.FR.419

Changed requirement

	ID	Precondition	Requirement definition
Old text	A00.FR.419	A00.FR.417 AND The Charging Station detects that the CSMS only allows connections using an older version of TLS, or only allows SSL	The Charging Station SHALL trigger an InvalidTLSVersion security event AND terminate the connection (See part 2 appendices for the full list of security events).
New text	A00.FR.419	A00.FR.417 AND The Charging Station detects that the CSMS only allows connections using an older version of TLS, or only allows SSL	The Charging Station SHALL trigger an InvalidTLSVersion security event AND terminate the connection (See part 2 appendices for the full list of security events). **NOTE: This is a critical security event that will need to be queued and sent to CSMS once a connection has been made, as described in use case A04.** **A security event only needs to be sent once for repeated failed connection attempts, in order to avoid overflow to the offline queue.**

4. Use case B Provisioning

4.1. Page 61 - (v1) Requirement B08.FR.19 and N02.FR.15 are ambiguous w.r.t. evse and instance wildcards [676]

Requirement B08.FR.19 tries to catch multiple situations in one requirement, but as a result it is no longer unambiguous. The requirement has therefore been split into multiple requirements, but with the same intention as B08.FR.19.

Delete requirement

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*.

The following new requirements replace B08.FR.19:

New requirements

ID	Precondition	Requirement definition
B08.FR.22	B08.FR.11 AND When Charging Station receives a GetReportRequest with a *component* in a *componentVariable* element that has a *component.evse.id*, but *component.evse.connector* is missing	The Charging Station SHALL report the component(s) with this *component.name*, *component.instance* and *component.evse.id* for every *component.evse.connector*, whilst taking into account B08.FR.24.
B08.FR.23	B08.FR.11 AND When Charging Station receives a GetReportRequest with a *component* in a *componentVariable* element that has no *component.evse.id*	The Charging Station SHALL report the component(s) with this *component.name*, *component.instance* for every *component.evse* field (including top level component without *component.evse*), whilst taking into account B08.FR.24.
B08.FR.24	B08.FR.11 AND When Charging Station receives a GetReportRequest with a *component* in a *componentVariable* element that has a value for *component.instance*	The Charging Station SHALL report the component(s) with this *component.name* for every *component.instance* field, whilst taking into account B08.FR.22, B08.FR.23.
B08.FR.25	B08.FR.11 AND When Charging Station receives a GetReportRequest with a *component* in a *componentVariable* element that has no *component.instance* field	The Charging Station SHALL report the component(s) with this *component.name* for every *component.instance* field or the component(s) without *component.instance* field, whichever is the case, whilst taking into account B08.FR.22, B08.FR.23.

4.1.1. Page 319 - N02.FR.15

Exactly the same applies, mutatis mutandis, for requirement N02.FR.15.

Delete requirement

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*.

New requirements

ID	Precondition	Requirement definition
N02.FR.18	N02.FR.11 AND When Charging Station receives a GetMonitoringReportRequest with a *component* in a *componentVariable* element that has a *component.evse.id*, but *component.evse.connector* is missing	The Charging Station SHALL report the component(s) with this *component.name*, *component.instance* and *component.evse.id* for every *component.evse.connector*, whilst taking into account N02.FR.20.
N02.FR.19	N02.FR.11 AND When Charging Station receives a GetMonitoringReportRequest with a *component* in a *componentVariable* element that has no *component.evse.id*	The Charging Station SHALL report the component(s) with this *component.name*, *component.instance* for every *component.evse* field (including top level component without *component.evse*), whilst taking into account N02.FR.20.
N02.FR.20	N02.FR.11 AND When Charging Station receives a GetMonitoringReportRequest with a *component* in a *componentVariable* element that has a value for *component.instance*	The Charging Station SHALL report the component(s) with this *component.name* for every *component.instance* field, whilst taking into account N02.FR.18, N02.FR.19.
N02.FR.21	N02.FR.11 AND When Charging Station receives a GetMonitoringReportRequest with a *component* in a *componentVariable* element that has no *component.instance* field	The Charging Station SHALL report the component(s) with this *component.name* for every *component.instance* field or the component(s) without *component.instance* field, whichever is the case, whilst taking into account N02.FR.18, N02.FR.19.

4.2. Page 62 - (v1) Use case B09/B10: Extended scenario description [683]

Use case B09 describes the setting of a NetworkConnectionProfile and use case B10 describes how to use NetworkConnectionProfiles to migrate a Charging Station to a new CSMS. The relationship with the variable OCPPCommCtrlr.NetworkConfigurationPriority was not made explicit. The use case descriptions have been updated for that.

4.2.1. Use case B09

The text marked in bold has been added to the scenario description.

No.	Type	Description
1	Name	Setting a new NetworkConnectionProfile.
2	ID	B09
	Functional block	B. Provisioning
3	Objectives	To enable the CSMS to update the connection details on the Charging Station.
4	Description	The CSMS updates the connection details on the Charging Station. For instance in preparation of a migration to a new CSMS. After completion of this use case, the Charging Station to CSMS connection data has been updated.
	Actors	Charging Station, CSMS
	Scenario description	**A Charging Station supports at least two network configuration slots, that are identified by a number. The available slot numbers are reported by the Charging Station in the *valuesList* of variable NetworkConfigurationPriority.** **For example: *valuesList* = "0,1,2" in the base report tells CSMS that three configuration slots, numbered 0, 1 and 2, are available (but not necessarily set).** **The configuration slot number that is used for the active connection is reported by variable OCPPCommCtrlr.ActiveNetworkProfile.** 1. The CSMS sends a SetNetworkProfileRequest PDU containing an updated connection profile **and a *configurationSlot* out of the *valuesList* of NetworkConfigurationPriority.** 2. The Charging Station receives the PDU, validates the content and stores the new data 3. The Charging Station responds by sending a SetNetworkProfileResponse PDU, with status *Accepted*
5	Prerequisites	The data supplied by the CSMS matches the Charging Station’s capabilities
6	Postcondition(s)	The Charging Station was able to store the new connection data

4.2.2. Requirement for configuration slots

A Charging Station must support at least two configuration slots for network profiles in order to support a migration. The number of the configuration slot must match an entry in the *valuesList* of the NetworkConfigurationPriority.

This was already implicitly required, or else the use case B09 and B10 would not work. It is now made explicit in the following new requirements.

New requirements

ID	Precondition	Requirement definition
B09.FR.05	When the value of *configurationSlot* in SetNetworkProfileRequest does not match an entry in *valuesList* of NetworkConfigurationPriority	The Charging Station SHALL respond by sending a SetNetworkProfileResponse message with status *Rejected*
B09.FR.06		A Charging Station SHALL support at least two configuration slots for network connection profiles.

4.2.3. Use case B10

The text marked in bold has been added to the scenario description.

No.	Type	Description
1	Name	Migrate to new CSMS, using a different NetworkConnectionProfile.
2	ID	B10
	Functional block	B. Provisioning
3	Objectives	After completion of this use case, the Charging Station connects to a new CSMS.
4	Description	This use case describes how a Charging Station can be instructed to connect to a new CSMS, by changing the order of NetworkConnectionProfiles in `NetworkConfigurationPriority`.
	Actors	Charging Station, CSMS 1, CSMS 2
	Scenario description	**A Charging Station supports at least two network configuration slots, that are identified by a number. The available slot numbers are reported by the Charging Station in the *valuesList* of variable NetworkConfigurationPriority.** **For example: *valuesList* = "0,1,2" in the base report tells CSMS that three configuration slots, numbered 0, 1 and 2, are available (but not necessarily set).** **The *value* of NetworkConfigurationPriority reports the order in which network configurations are tried to make a connection.** **For example: *value* = "1,0" means that Charging Station will first try configuration slot 1 and if that fails after the number of attempts configured in NetworkProfileConnectionAttempts, it will try to connect with configuration slot 0.** 1. CSMS 1 sets a new value for the `NetworkConfigurationPriority` configuration variable via SetVariablesRequest, such that the NetworkConnectionProfile for CSMS 2 becomes first in the list and the existing connection to CSMS 1 becomes second in the list. 2. The Charging Station responds with a SetVariablesResponse with status *Accepted* 3. CSMS 1 instructs the Charging Station to perform a `Reset OnIdle`. 4. The Charging Station reboots and connects via the new primary NetworkConnectionProfile to CSMS 2.
5	Prerequisites	Use case B09 - Setting a new NetworkConnectionProfile was executed successfully prior to this use case The data supplied by the CSMS matches the Charging Station’s capabilities
6	Postcondition(s)	The Charging Station is connected via a different NetworkConnectionProfile.

5. Use case C Authorization

5.1. Page 90 - (v1) C07 requirements for *certificateStatus* missing [680]

In case of ISO 15118 Plug&Charge the AuthorizeResponse returns a *certificateStatus* of type AuthorizeCertificateStatusEnumType. Requirement C07.FR.04 states that an authorization status must be returned, but the exact values are not defined.

Requirements have been added that describe which values to use for *certificateStatus*.

New requirements

ID	Precondition	Requirement definition	Note
C07.FR.13	C07.FR.04 AND the certificate chain (provided in *certificate* or *iso15118CertificateHashData*) is valid AND authorization status of *idToken* is one of `Blocked`, `Expired`, `Invalid`, `Unknown`	CSMS SHALL return an AuthorizationResponse containing a *certificateStatus* = `ContractCancelled` and the authorization status in *idTokenInfo.status*.	Certificate is valid, but EMAID is not accepted.
C07.FR.14	C07.FR.04 AND the certificate chain (provided in *certificate* or *iso15118CertificateHashData*) is valid AND authorization status of *idToken* is NOT one of `Blocked`, `Expired`, `Invalid`, `Unknown`	CSMS SHALL return an AuthorizationResponse containing a *certificateStatus* = `Accepted` and the authorization status in *idTokenInfo.status*.	Charging can still not be allowed if *idTokenInfo.status* is other than `Accepted` (e.g. `ConcurrentTx` or `NotAtThisLocation`).
C07.FR.15	C07.FR.04 AND the certificate chain (provided in *certificate* or *iso15118CertificateHashData*) has expired	CSMS SHALL return an AuthorizationResponse containing a *certificateStatus* = `CertificateExpired` and an *idTokenInfo.status* = `Expired`	If certificate is expired, then status of *idToken* is also reported expired.
C07.FR.16	C07.FR.04 AND the certificate chain (provided in *certificate* or *iso15118CertificateHashData*) has been revoked	CSMS SHALL return an AuthorizationResponse containing a *certificateStatus* = `CertificateRevoked` and an *idTokenInfo.status* = `Invalid`	If certificate is revoked, then status of *idToken* is reported as invalid.
C07.FR.17	C07.FR.04 AND the certificate chain (provided in *certificate* or *iso15118CertificateHashData*) cannot be verified or is invalid	CSMS SHALL return an AuthorizationResponse containing a *certificateStatus* = `CertChainError` and an *idTokenInfo.status* = `Invalid`	If certificate is cannot be verified, then status of *idToken* is reported as invalid.

5.1.1. Page 408 - AuthorizeCertificateStatusEnumType

The enumeration AuthorizeCertificateStatusEnumType contains some values that are not used. These enumeration values continue to exist, so as not to change the JSON schema, but their description is changed to show that these values have no meaning.

Updated text in **bold**:

AuthorizeCertificateStatusEnumType

Value	Description
Accepted	Positive response
SignatureError	**<not used>**
CertificateExpired	If the **contract certificate in the AuthorizeRequest** is expired.
CertificateRevoked	**If the Charging Station or CSMS determine (via a CRL or OCSP response) that the contract certificate in the AuthorizeRequest is marked as revoked**.
NoCertificateAvailable	**<not used>**
CertChainError	**If the contract certificate contained in the AuthorizeRequest message is not valid.**
ContractCancelled	If the EMAID provided by EVCC **is invalid, unknown, expired or blocked.**

5.2. Page 97 - (v1) Requirement C10.FR.06 needs to be removed [685]

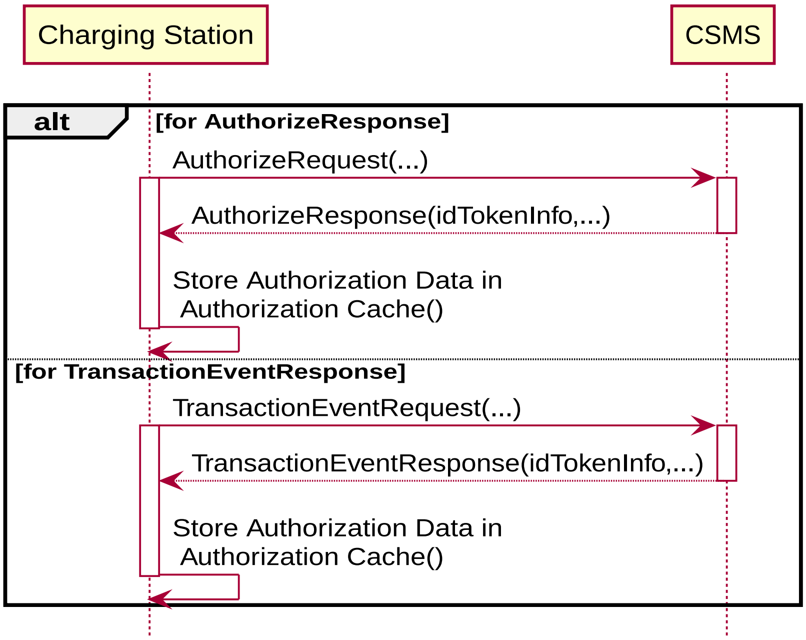
Requirement C10.FR.06 is an invalid requirement, because the ReserveNowRequest does not contain IdTokenInfo, so there is no information to update the Authorization Cache with.

Deleted requirement

ID	Precondition	Requirement definition	Note
~~C10.FR.06~~	Upon receipt of ReserveNowRequest.	The Charging Station SHALL update the Authorisation Cache entry.	The update is to be done with the IdTokenInfo value from the request as described under Authorization Cache.

5.2.1. Page 96 - Update sequence diagram

As a result of the above, the "for ReserveNowRequest" part has been removed from sequence diagram "Figure 31".


Charging Station
CSMS
alt
[for AuthorizeResponse]
AuthorizeRequest(...)
AuthorizeResponse(idTokenInfo,...)
Store Authorization Data in
Authorization Cache()
[for TransactionEventResponse]
TransactionEventRequest(...)
TransactionEventResponse(idTokenInfo,...)
Store Authorization Data in
Authorization Cache()

5.3. Page 102 - (v1) Requirement C13.FR.04 enhanced [701]

Requirement C13.FR.04 suggests that any identifier must be accepted, but that was not the intention. In fact, it is in conflict with use case C15 that describes offline authorization of an unknown identifier. Requirement C15.FR.08 says that any **unknown** identifier not in Authorization Cache or Local Authorization List (prerequisite of C15) must be accepted. C13.FR.04 is updated to reflect this.

Changed requirement

	ID	Precondition	Requirement definition	Note
Old text	C13.FR.04	If configuration variable `OfflineTxForUnknownIdEnabled` is true AND The Charging Station is offline.	Any identifier SHALL be allowed to authorize a transaction.	
New text	C13.FR.04	If configuration variable `OfflineTxForUnknownIdEnabled` is true AND The Charging Station is offline.	Any identifier **that is present in neither the Authorization Cache nor the Local Authorization List** SHALL be allowed to authorize a transaction.	See also C15.FR.08

6. Use Case E Transactions

6.1. Page 147 - (v1) Use case E07 - Scenario description step order incorrect [704]

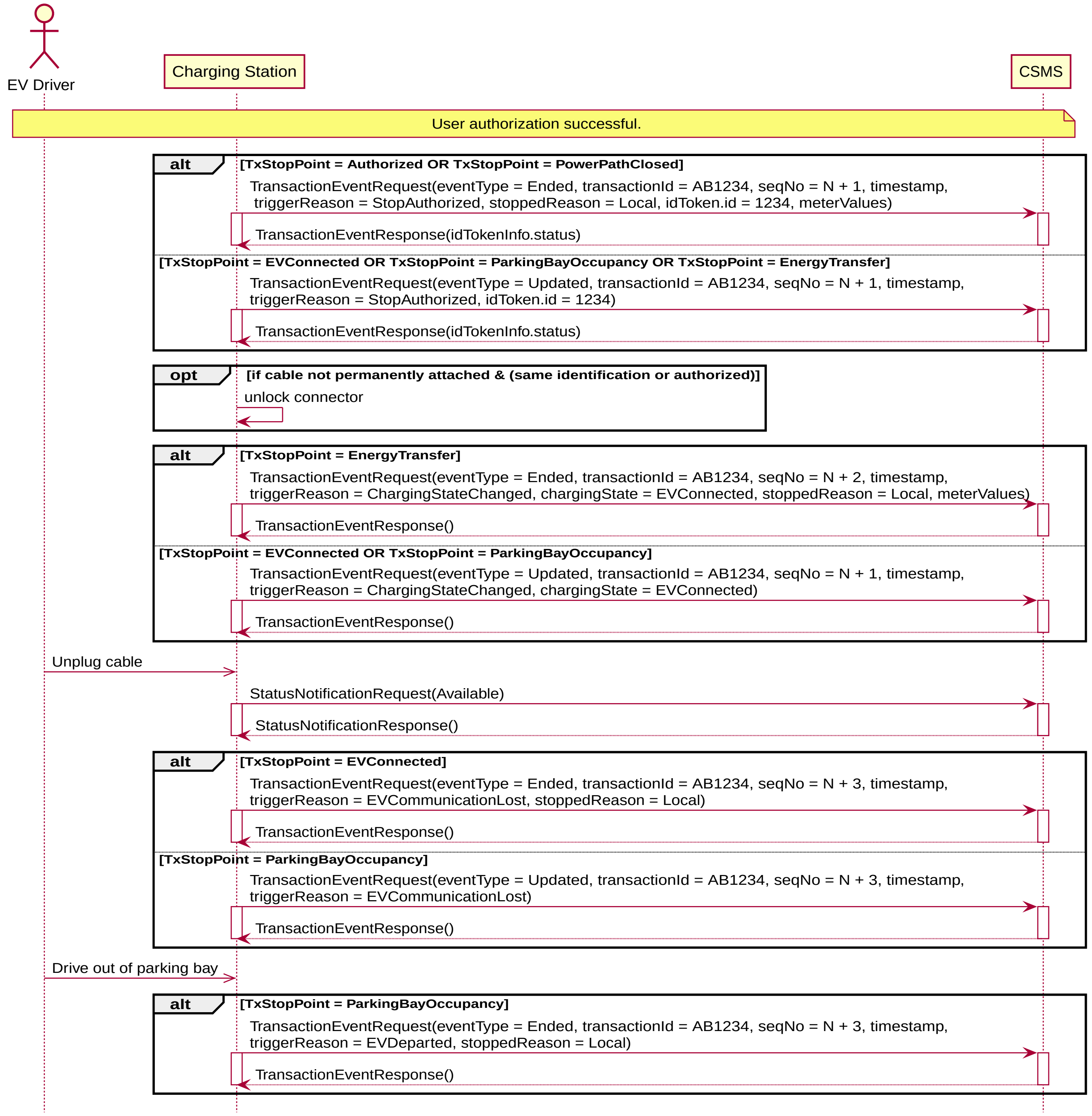
The Charging Station must first stop the energy transfer as described by step 4, before transmitting the TransactionEventRequest(eventType = Ended) message from step 2 and 3.

	No.	Type	Description
Old text		*Scenario description* TxStopPoint = `Authorized` (or `PowerPathClosed`)	**1.** The EV Driver presents IdToken a second time to end charging. **2.** The Charging Station sends a TransactionEventRequest (eventType = Ended) with *triggerReason* = `StopAuthorized` and *stoppedReason* = `Local`. **3.** The CSMS responds with a TransactionEventResponse. **4.** The Charging Station stops the energy transfer and if the cable is not permanently attached, the Charging Station unlocks the cable.
New text		*Scenario description* TxStopPoint = `Authorized` (or `PowerPathClosed`)	**1.** The EV Driver presents IdToken a second time to end charging. **2.** The Charging Station stops the energy transfer and if the cable is not permanently attached, the Charging Station unlocks the cable. **3.** The Charging Station sends a TransactionEventRequest (eventType = Ended) with *triggerReason* = `StopAuthorized` and *stoppedReason* = `Local`. **4.** The CSMS responds with a TransactionEventResponse.

6.2. Page 148 - (v1) Use case E07: Wrong triggerReason shown in sequence diagram fig. 56 [687]

The fourth TransactionEventRequest in sequence diagram Figure 56 contains an incorrect *triggerReason* and should not have an *idToken*. Changed to *triggerReason* = `ChargingStateChanged`, *chargingState* = `EVConnected`.

Figure 56. Sequence Diagram: Transaction locally stopped by IdToken with TransactionEventRequest reported strictly by TxStopPoint configuration


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 = ChargingStateChanged, chargingState = EVConnected)
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()

6.3. Page 150 - (v1) Use case E07: Clarify 'normal' and 'correct' for *stoppedReason* [693]

Some requirements in E07 mention "ended in a normal way" and "set to a correct value", but do not explain what normal and correct is.

	ID	Precondition	Requirement definition	Note
Old text	E07.FR.04	If a transaction is ended in a normal way.	The stoppedReason element MAY be omitted.	e.g. EV-driver presented IdToken to stop the transaction.
New text	E07.FR.04	If a transaction **is stopped on request of the EV driver at the Charging Station**.	**Charging Station MAY omit the stoppedReason element from the final TransactionEventRequest with *eventType* = Ended**	e.g. EV-driver presented IdToken to stop the transaction **or pressed a "stop" button (not the emergency stop)**. **See use case F03 for remotely stopping.**

	ID	Precondition	Requirement definition	Note
Old text	E07.FR.05	If a transaction is ended in a normal way	The stoppedReason SHOULD be assumed 'Local'.	e.g. EV-driver presented IdToken to stop the transaction.
New text	E07.FR.05	If a transaction **is stopped on request of the EV driver at the Charging Station**.	**Charging Station SHOULD use a stoppedReason = `Local` in the final TransactionEventRequest with *eventType* = Ended.**	e.g. EV-driver presented IdToken to stop the transaction **or pressed a "stop" button (not the emergency stop)**. **See use case F03 for remotely stopping.**
Old text	E07.FR.06	If the transaction is *not* ended normally.	stoppedReason SHOULD be set to a correct value.	
New text	E07.FR.06	If a transaction is **stopped, but not on request of the EV driver at the Charging Station**.	**Charging Station SHOULD use the most appropriate value from ReasonEnumType for *stoppedReason* in the final TransactionEventRequest with *eventType* = `Ended`**.	**Apart from remotely stopping (`Remote`), CSMS revoking authorization (`DeAuthorized`) or disconnecting the EV (`EVDisconnected`), most other reasons are related to technical faults or energy limitations.**

6.3.1. Page 403 - TransactionType field *stoppedReason*

The description for field *stoppedReason* in TransactionEventRequest has been improved to make clear that this event does not have to concur with the TransactionEventRequest(Ended) or TxStopPoint, but may have happened some time before.

TransactionType

	Field Name	Field Type	Card.	Description
Old text	**stoppedReason**	ReasonEnumType	0..1	Optional. This contains the reason why the transaction was stopped. MAY only be omitted when Reason is "Local".
New text	**stoppedReason**	ReasonEnumType	0..1	Optional. **The *stoppedReason* is the reason/event that initiated the process of stopping the transaction. It will normally be the user stopping authorization via card (Local or MasterPass) or app (Remote), but it can also be CSMS revoking authorization (DeAuthorized), or disconnecting the EV when TxStopPoint = EVConnected (EVDisconnected). Most other reasons are related to technical faults or energy limitations.** MAY only be omitted when ***stoppedReason*** is "Local"

7. Use Case F Remote Control

7.1. Page 180 - (v1) Requirement F03.FR.03 contains wrong precondition [700]

The precondition of requirement F03.FR.03 was incorrectly merged from Errata v2 into Edition 2, and the associated Note was not relevant for this situation.
It needs to be changed as follows:

Changed requirement

	ID	Precondition	Requirement definition	Note
Old text	F03.FR.03	F03.FR.01 AND TxStopPoint configuration causes transaction to end (E.g. TxStopPoint is NOT `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.
New text	F03.FR.03	F03.FR.01 AND TxStopPoint configuration causes transaction to end (E.g. TxStopPoint is ~~NOT~~ `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.~~

7.2. Page 187 - (v1) Requirement F06.FR.12 is too strict [707]

Requirement F06.FR.12 explicitly tells a Charging Station to reject a TriggerMessageRequest for a *requestedMessage* `StatusNotification` without *evse* or *evse.connectorId*. There is no need to require this from a Charging Station, since F06.FR.13 already mandates that CSMS shall provide an *evse.connectorId* (and an *evse.id*, because that is mandatory in the *evse* object) in this message.

The requirement definition of F06.FR.12 has been relaxed from SHALL to a MAY, so that a Charging Station implementation that is able to handle to a request without *evse.connectorId* and an implementation that rejects this, are both allowed, since a CSMS is not allowed to send a request without *evse_connectorId*.

Changed requirement

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

8. Use Case G Availability

8.1. Page 192 - (v1) G01.FR.08 contradicts H01.FR.24 [692]

Requirement G01.FR.08 states that a StatusNotification must be sent when a connector becomes reserved. However, this topic is already covered in use case H01 in a slightly different way. Therefore, the "becomes reserved" must be removed from G01.FR.08 and left to H01.FR.24.

Table 1. G01 - Requirements

	ID	Precondition	Requirement definition
Old text	G01.FR.08	When a connector of an EVSE becomes reserved or a cable is plugged-in AND 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.
New text	G01.FR.08	When a **cable is plugged in to a connector** **of** an EVSE AND 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. Use Case H Reservation

9.1. Page 205 - (v1) Missing option to send NotifyEvent instead of StatusNotification [699]

Instead of StatusNotificationRequest it is also allowed to send a NotifyEvent(AvailabilityState) for the connector, which will become the preferred method in future OCPP releases. This option was missing from use case H and is added to the following requirements.

Changed requirements

	ID	Precondition	Requirement definition	Note
Old text	H01.FR.20	H01.FR.04 AND 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 text	H01.FR.20	H01.FR.04 AND amount of EVSEs available equals the amount of reservations	The Charging Station SHALL send **for all connectors of the EVSE:** - a StatusNotificationRequest with *connectorStatus* = `Reserved`, **OR** **- a NotifyEventRequest with *component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta", *actualValue* = "Reserved"**	If an EVSE is reserved, all of its connectors are reported as reserved.
Old text	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.
New text	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 **for all connectors of the EVSE**: - a StatusNotificationRequest with *connectorStatus* = `Reserved`, **OR** **- a NotifyEventRequest with *component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta", *actualValue* = "Reserved"**	If an EVSE is reserved, all of its connectors are reported as reserved.
Old text	H01.FR.24	H01.FR.06 AND 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.
New text	H01.FR.24	H01.FR.06 AND amount of reservations for a specific *connectorType* equals the amount of available EVSEs with that specific *connectorType*	The Charging Station SHALL send **for all connectors of the EVSEs that have the specific *connectorType*** - a StatusNotificationRequest with *connectorStatus* = `Reserved`, **OR** **- a NotifyEventRequest with *component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta", *actualValue* = "Reserved"**	If an EVSE is reserved for a specific *connectorType*, all connectors on the EVSE are reported as reserved.

9.1.1. Page 203 - (v1) Added option to use case description to send NotifyEventRequests

Use case H01 scenario S2 only mentions StatusNotificationRequests, but the use of NotifyEventRequests is also allowed. This has been added in **bold**, similarly to how this was done in use case G01 StatusNotification.

S2	*Scenario objective*	Reserve a specific EVSE at a Charging Station

	Scenario description	1. EV Driver asks the CSMS to reserve a specific EVSE at the Charging Station. 2. The CSMS sends ReserveNowRequest with a EVSE to a Charging Station. 3. Upon receipt of ReserveNowRequest, the Charging Station responds with ReserveNowResponse with status *Accepted*. 4. The Charging Station sends StatusNotificationRequest with the status *Reserved* for all Connectors of that EVSE. 5. The CSMS responds with StatusNotificationResponse to the Charging Station.
	Alternative scenario description	**Steps 1, 2 and 3 as above.** **4. 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*, *variable* = "AvailabilityState" and *actualValue* = "Reserved".** **5a. Optionally, Charging Station can also report a NotifyEventRequest for *component* = "EVSE", *variable* = "AvailabilityState" and *actualValue* = "Reserved" , and when applicable, also report this for *component* = "ChargingStation".**
	Prerequisite(s)	The specified EVSE of the Charging Station has status *Available*
	Postcondition(s)	Successful postcondition: The Charging Station has accepted the ReserveNowRequest AND sent StatusNotificationRequests with status *Reserved*. Failure postcondition: The Charging Station has rejected the ReserveNowRequest OR The Charging Station has NOT sent StatusNotificationRequests with status *Reserved*.

9.2. Page 209 - (v1) Remark about authorization in use case H03 [711]

Use case H01 has a remark that says: "It is RECOMMENDED to validate the Identifier with an AuthorizeRequest after reception of ReserveNowRequest and before the start of the transaction." Use case H03 about using a reservation does not have a recommendation to validate before starting the transaction.

In order to be consistent with H01, this has been added to the remark of H03, as shown in **bold**:

7	Error handling	n/a
8	Remark(s)	**It is RECOMMENDED to validate the Identifier with an AuthorizeRequest after reception of ReserveNowRequest and before the start of the transaction.**

9.3. Page 210 - (v1) Requirement H03.FR.08 is not clear about groupIdToken lookup [684]

Requirement H03.FR.08 can mistakenly be interpreted as having to look up the ***groupIdToken*** in the Local Authorization List or Authorization Cache. However, the intention is to look up the incoming *idToken* to get its associated *groupIdToken*, if any.

The requirements H03.FR.07 and H03.FR.08 exist to make clear, that for a reserved EVSE or connector a lookup or authorize request for *idToken* is needed when a *groupIdToken* is involved.

Changed requirement

	ID	**Precondition**	**Requirement definition**
Old text	H03.FR.08	H03.FR.07 AND If it is not found in the Local Authorization List or Authorization Cache.	The Charging Station SHALL send an AuthorizeRequest for the incoming IdToken to the CSMS in order to get its associated *groupIdToken*.
New text	H03.FR.08	H03.FR.07 AND If **the incoming IdToken** is not found in the Local Authorization List or Authorization Cache.	The Charging Station SHALL send an AuthorizeRequest for the incoming IdToken to the CSMS in order to get its associated *groupIdToken*. (Note: This AuthorizeRequest may already have been performed when the idToken was presented for authorization.)

10. Use Case J Meter Values

10.1. Page 228 - (v1) Requirement J01.FR.14 is unclear that meter values for all EVSEs must be sent [674]

J01 is not clear about the fact that MeterValuesRequest for clock-aligned data always need to be sent for all locations, including the grid energy meter, which is designated by *evseId* = 0. It is stated in the text in par. 2.3: "When a Charging Station can measure the same measurand on multiple locations or phases, all possible locations and/or phases SHALL be reported when configured in one of the relevant Configuration Variables." The requirement J01.FR.14 has been extended to refer to all possible locations and phases.

Changed requirement

	ID	Precondition	Requirement definition	Note
Old text	J01.FR.14	When configured to send MeterValuesRequest, See: Meter Values - Configuration	The Charging Station SHALL send MeterValuesRequest messages to the CSMS as configured.	
New text	J01.FR.14	When configured to send MeterValuesRequest, See: Meter Values - Configuration	The Charging Station SHALL send MeterValuesRequest messages to the CSMS as configured **in Meter Values - Configuration, for all *evseIds*, locations and phases for which a configured measurand is supported.**	**It is possible that certain measurands are not available for every location. For example, *evseId* = 0 (grid meter) will not have a "Current.Offered" or "SoC" measurand.**

10.2. Page 230 - (v1) Requirement J02.FR.10 refers to all TransactionEventRequest messages, but should be specific to only eventType = Updated [705]

A TransactionEventRequest(Started/Update) should only have sampled values that are part of the same sampling interval. Ideally, this would mean that all sampled values have the same timestamp, and can thus be part of a single *meterValue* element. In practice, however, when multiple measurands or meters are sampled the associated timestamps may differ slightly. This is acceptable, as long as the samples belong to the same sampling interval.

This was the intention of J02.FR.10 with the phrase "belong to the timestamp in the message", but it could also be interpreted as requiring identical timestamps. Also, it forgot to mention that it only applies to Started and Updated events, since an Ended event can contain *metervalues* for multiple timestamps.

Changed requirement

	ID	Precondition	Requirement definition	Note
Old text	J02.FR.10		The meterValue measurements in the same TransactionEventRequest message SHALL all belong to the timestamp in the message	meterValues for other timestamps should be sent in separate TransactionEventRequest messages.
New text	J02.FR.10	**If a TransactionEventRequest message with eventType = Started or eventType = Update contains multiple *meterValue* elements, rather than one *meterValue* with one or more *sampledValue* elements**	**All *meterValue* elements SHALL have a timestamp that is within the current sampling interval, i.e.: (transaction event timestamp - `SampledDataTxUpdatedInterval`) < *meterValue.timestamp* <= transaction event timestamp**	**Only for *eventType* = Ended can a TransactionEventRequest have meter values for multiple intervals.**

10.3. Page 231 - (v1) J01 misses requirement that meter value must be for current transaction [673]

It is perhaps obvious, but not stated anywhere. Transaction-related meter values reported in the TransactionEventRequest must only report the measurand(s) associated with the *evse* of the TransactionEventRequest.

New requirement

ID	Precondition	Requirement definition	Note
J02.FR.22		Meter values reported in a TransactionEventRequest message SHALL all be related to EVSE on which the transaction is taking place.	

11. Use Case K Smart Charging

11.1. Page 238 - (v1) Text in section 3.3 does not match ChargingProfileKindEnumType description [708]

The description of the ChargingProfileKindEnumType `Relative` was updated in Edition 2 to be more exact. This update was unfortunately not performed in section 3.3 Charging Profile Recurrency that introduces the charging profile kinds.

Below is the updated text shown in bold:

	ChargingProfile Kind	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.** **No value for *startSchedule* should be supplied.**

12. Use Case L FirmwareManagement

12.1. Page 287 - (v1) Improved title of figure 119 [695]

Figure 119 shows the transitions between all FirmwareStatusEnumType values. As such, it is a state transition diagram. The title, however, calls it "Firmware update process", which is not correct, because it does not cover all steps for performing a firmware update.

Old text	Figure 119. Firmware update process
New text	Figure 119. Firmware **status transitions**

13. Use Case M ISO 15118 CertificateManagement

13.1. Page 310 - (v1) M04.FR.07 has an incorrect requirement definition [703]

Requirement M04.FR.07 mentions a hash algorithm used during installation, but no hash algorithm is used to install a certificate. The intention of this requirement was, as is suggested by the note, that the CSMS, when deleting a certificate, uses the same *hashAlgorithm* as the Charging Station when generating the *certificateHashData* for a certificate.

	ID	Precondition	Requirement definition	Note
Old text	M04.FR.07	When deleting a certificate	The CSMS SHALL use the *hashAlgorithm*, which was used to install the certificate.	When a new firmware is installed it is RECOMMENDED that the CSMS requests the certificate first using GetInstalledCertificateIdsRequest to be sure of the used *hashAlgorithm*.
New text	M04.FR.07	When deleting a certificate	The CSMS SHALL use **the same *hashAlgorithm* as the Charging Station uses to report the certificateHashData for the certificate in the GetInstalledCertificateIdsResponse.**	**This ensures CSMS uses a *hashAlgorithm* that is supported by the Charging Station.**

14. Use Case N Diagnostics

14.1. Page 317 - (v1) N01.FR.10 not clear when to report UploadFailure [696]

Requirement N01.FR.10 does not make clear whether the LogStatusNotification about failure to upload should be sent after all retry attempts or at each failure. Both options are allowed, but it is recommended to do this after all retry attempts have failed. This has been added to the note.

	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 *UploadFailure*, *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 **the status only after all retry attempts have failed.** **A Charging Station MAY send a new `Uploading` status upon each retry attempt.**

14.2. Page 331 - (v1) Requirement N09.FR.04 has been rephrased [688]

Requirement N09.FR.04 for CSMS states that a reference to a customer by either *idToken*, *customerCertificate* or *customerIdentifier* is needed, but it does not tell what to do if that is not obeyed.

A new requirement has been added for Charging Station for this case.

New requirement

ID	Precondition	Requirement definition	Note
N09.FR.09	When CustomerInformationRequest contains none of *idToken*, *customerCertificate* or *customerIdentifier* OR CustomerInformationRequest contains more than one of *idToken*, *customerCertificate* or *customerIdentifier*	Charging Station SHALL respond with *status* = `Invalid`	Only one value for either *idToken*, *customerCertificate* or *customerIdentifier* may be provided. Charging Station counterpart requirement of N09.FR.04.

15. Messages

15.1. Page 353 - (v1) Clarification for use of *certificate* and *iso15118CertificateHashData* in AuthorizeRequest [675]

In case of ISO 15118 Plug&Charge the AuthorizeRequest has two optional fields: *certificate* and *iso15118CertificateHashData*. The behaviour is described in requirements C07.FR.05 and C07.FR.06, but it was not clear enough that only one of these fields is needed.

The field *certificate* contains the entire contract certificate chain. It is only needed in case of central contract validation, where Charging Station cannot locally validate the contract certificate, e.g. because it is lacking the root certificate. If *certificate* is provided, it is no longer needed to provide *iso15118CertificateHashData*.

Text in **bold** is added to the description.

AuthorizeRequest

Field Name	Field Type	Card.	Description
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. **Only needed in case of central contract validation when Charging Station cannot validate the contract certificate.**
idToken	IdTokenType	1..1	Required. This contains the identifier that needs to be authorized.
iso15118CertificateHashData	OCSPRequestDataType	0..4	Optional. Contains the information needed to verify the EV Contract Certificate via OCSP. **Not needed if *certificate* is provided.**

15.2. Page 381 - (v1) Updated description for idToken in TransactionEventRequest [709]

The *idToken* in a TransactionEventRequest is only supposed to be sent after an id token has been authorized, either locally or centrally. This happens when starting and stopping the authorization for a transaction. CSMS then returns the validity status of the *idToken* in the TransactionRequestResponse. When a transaction is stopped via a RequestStopTransactionRequest or a ResetRequest, no id token is involved and as a result no *idToken* should be provided in the TransactionEventRequest, because CSMS does not need to check validity.

The description of *idToken* has been updated to make this clear.

	Field Name	Field Type	Card.	Description
Old 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.
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, **so that CSMS can return the *idTokenInfo* in the TransactionEventResponse.** ***idToken* should not be present in the TransactionEventRequest when a transaction is ended by a RequestStopTransactionRequest or a ResetRequest.**

16. Data Types

16.1. Page 386 - (v1) issuerKeyHash in CertificateHashDataType must be type identifierString [691]

The field type of *issuerKeyHash* in CertificateHashDataType must be "identifierString[0..128]", instead of "string[0..128]". The difference is, that identifierString is case-insensitive. This is, however, not checked by the JSON schema, and as a result this change does not affect the JSON schema.

Changed field type for issuerKeyHash:

CertificateHashDataType

Field Name	Field Type	Card.	Description
hashAlgorithm	HashAlgorithmEnumType	1..1	Required. Used algorithms for the hashes provided.
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.
issuerKeyHash	**identifierString[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.
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.

16.2. Page 396 - (v1) NetworkConnectionProfileType [683]

The data type NetworkConnectionProfileType has two fields that do not serve a purpose.

- The field *ocppVersion* has no use, because the selection of the OCPP version that a charging station will use, is done during the websocket handshake. It is not determined by the NetworkConnectionProfile.
- The field *ocppInterface* is mandatory, but in most cases a CSMS will not even be aware of which interfaces a charging station supports or should use to connect. It is a mandatory field, so CSMS must provide something, but that might not match with the capability of the charging station. To remedy this, a charging station is allowed to use a different interface if it cannot connect via the given *ocppInterface*.

The descriptions of these fields have been updated with text in bold to make this clear.

Changed descriptions in NetworkConnectionProfileType

Field Name	Field Type	Card.	Description
ocppVersion	OCPPVersionEnumType	1..1	Required. ~~Defines the OCPP version used for this communication function.~~ **This field is ignored, since the OCPP version to use is determined during the websocket handshake.**
…			
ocppInterface	OCPPInterfaceEnumType	1..1	Required. Applicable Network Interface. **Charging Station is allowed to use a different network interface to connect if the given one does not work.**
…			

17. Enumerations

17.1. Page 419 - (v1) Description for idTokenEnumType MacAddress [664]

A description is missing for value `MacAddress` of IdTokenEnumType.

Value	Description
MacAddress	The MacAddress of the EVCC (Electric Vehicle Communication Controller) that is connected to the EVSE. This is used as a token type when the MAC address is used for authorization ("Autocharge").

18. Referenced Components and Variables

18.1. Page 436 - (v1) Websocket-related variables in Part 4 [690]

Add the following note below section heading "General":

NOTE WebSocket-related variables are described in "OCPP-2.0.1 Part 4 JSON over WebSockets".

18.1.1. Page 430 - 2.1.13 WebSocketPingInterval

This configuration variable at this location has "Required = No", but that is confusing, because it is required for a WebSocket implementation. All WebSocket configuration variables are described in Part 4.

Replace table describing this variable with a reference to Part 4, as follows:

This configuration variable is described in "OCPP-2.0.1 Part 4 JSON over WebSockets".

18.2. Page 444 - (v1) SecurityCtrlr.BasicAuthPassword and Identity should have dataType=string

The *dataType* of SecurityCtrlr.BasicAuthPassword is mistakenly shown as "passwordString". The content is similar to a passwordString as defined in part 2, but the device model dataType is "string". The same applies to SecurityCtrlr.Identity which shows *dataType* "identifierString".

Replace the descriptions of `BasicAuthPassword` and `Identity` by the updated text below. This change has also been made in Part 2 Appendix chapter 3 "Standardized Components".

Updated dataType:
(change shown in ***bold italic***)

`BasicAuthPassword`

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

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	BasicAuthPassword	
	variableAttributes	**mutability**	WriteOnly
	variableCharacteristics	**dataType**	***string***
		maxLimit	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 (alphanumeric 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. This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.

Updated dataType:
(change shown in ***bold italic***)

Identity

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	Identity	
	variableAttributes	**mutability**	ReadOnly or ReadWrite
	variableCharacteristics	**dataType**	***string***
		maxLimit	48 (Charging Station Identity)
Description	The Charging Station identity. Identity is an identifierString, however because this value is also used as the basic authentication username, the colon character ':' SHALL not be used. Maximum length was chosen to ensure compatibility with EVSE ID from [EMI3-BO] "Part 2: business objects".		

18.3. Page 452 - (v1) Incomplete description TxStopPoint Authorized and PowerPathClosed [704]

A transaction shall not end while energy transfer is still ongoing, otherwise it is not possible to report a correct final meter value for the transaction. TxStopPoints Authorized and PowerPathClosed will trigger the transaction to be ended after a StopAuthorized or Deauthorized event, but the Charging Station must wait until the energy transfer has been ended, before transmitting the TransactionEventRequest with eventType = Ended, so that this message can contain the final meter values.

The description of these TxStopPoints has been enhanced to make this clear.

2.6.6.2 TxStopPoint values

Value	Description
Authorized	Driver or EV is no longer authorized, this can also be some form of anonymous authorization like a start button. **The end of authorization will cause the Charging Station to stop the energy transfer, after which the TransactionEventRequest with eventType = Ended will be transmitted.**
PowerPathClosed	All preconditions for charging are no longer met. 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. **This will cause the Charging Station to stop the energy transfer, after which the TransactionEventRequest with eventType = Ended will be transmitted.** It is exactly the same as having the values `EVConnected`, `Authorized` in `TxStopPoint`. Despite its name, this event is not related to the state of the power relay.

19. Appendix 1

19.1. Page 2 - (v1) InvalidFirmwareSignature/SigningCertificate are critical security evente [682]

The column "Critical" must be set to "yes" for a security event InvalidFirmwareSignature and InvalidFirmwareSigningCertificate, because of the SHALL-requirements L01.FR.02 and L01.FR.03.

Security Event	Description	Critical
InvalidFirmwareSignature	The firmware signature is not valid	**Yes**
InvalidFirmwareSigningCertificate	The certificate used to verify the firmware signature is not valid	**Yes**

20. Appendix 3

20.1. Page 9 - (v1) OCPPCommCtrlr.ActiveNetworkProfile must be of type integer [697]

ActiveNetworkProfile was mistakenly shown as having type string. This must be integer.

OCPPCommCtrlr

Description		
Logical Component responsible for configuration relating to information exchange between Charging Station and CSMS.		
Variables	**Type**	**Description**
ActiveNetworkProfile	**integer**	[…]

20.2. Page 10 - (v1) SecurityCtrlr.BasicAuthPassword and Identity should have dataType=string [698]

BasicAuthPassword was shown as type "passwordString" and for Identity as type "identifierString". The type for the device model variable in both cases must be "string".

SecurityCtrlr

Description		
Logical Component responsible for configuration relating to security of communications between Charging Station and CSMS.		
Variables	**Type**	**Description**
BasicAuthPassword	**string**	[…]
Identity	**string**	[…]