OCPP 2.0.1

Part 2 - Errata

v1.0, 2021-10-01

Table of Contents

1. Disclaimer

Copyright © 2010 – 2021 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	Author	Description
1.0	2021-10-01	Franc Buve, Milan Jansen *(OCA)*	Release of R2.0.1 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.

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. 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. 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. 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>

4. Use case A Security

4.1. Page 19 - 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 - 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 string that can only contain characters from the following character set: "a-z", "A-Z", "0-9" or any of the following limited set of symbols: * - _ = : + | @

4.2.2. Page 20 - 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 - 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 - Note about changed password encoding between OCPP 1.6 and 2.0.1 [431]

4.3.1. Page 20 - 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 - Security Profile 2

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

7	**Remark(s)**	[…] 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 - 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 - Implicit requirement for CSMS made explicit [440]

4.5.1. Page 20 - 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 - 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 - 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 AND 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** AND 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 AND 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** AND 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 - 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 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.

	ID	Precondition	Requirement definition
New text	A00.FR.407	**NOT A00.FR.429 AND** 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 - 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 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 - 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 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`.

ID	Precondition	Requirements
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 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).
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 - 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 - Clarification for reconnection with new certificate [149]

4.10.1. Page 33 - 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 - 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 - 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** ***interval* > 0**	The Charging Station SHALL adjust the heartbeat interval in accordance with the interval from the response message.	

5.1.1. Page 421 - 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 - Return first instance when no instance specified [476]

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 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.14	B06.FR.01 AND 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 56 - 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.

5.4. Page 56 - 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** When the Charging Station receives a getReportRequest for supported *criteria*	The Charging Station SHALL send a getReportResponse with Accepted

5.5. Page 66 - 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) **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) **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 - 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 (a) the IdToken used for stopping the transaction is the same as the IdToken that started the transaction OR (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 AND (a) the presented idToken is the same as the idToken that started the authorization OR (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.

6.2. Page 75 - 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.

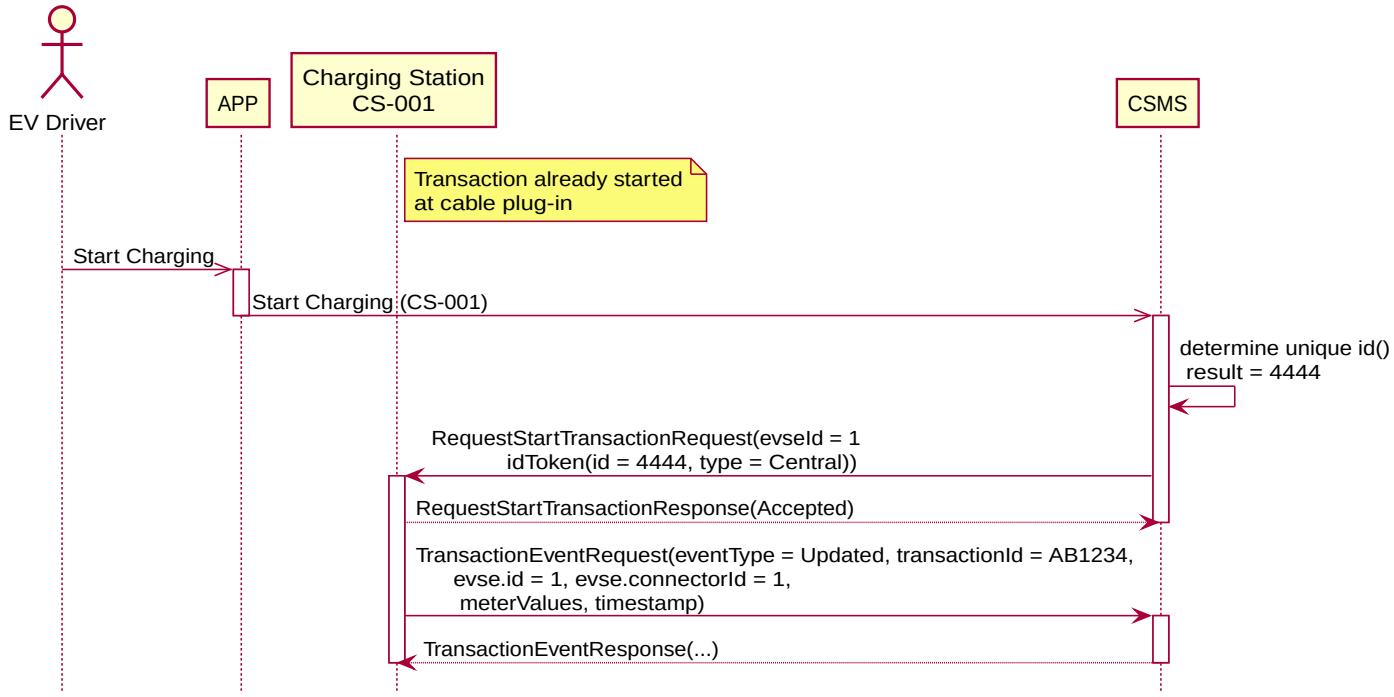
NOTE This errata is closely related to Page 126 - Missing requirement about authorization during transaction [514], which deals with the situation when a new *idToken* is authorized, despite the requirements in this section.

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 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 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*.

6.3. Page 83 - 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.



6.4. Page 84 - Description and remarks of use case C05 [503, 504]

6.4.1. 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. **2.** The app sends a start request to the CSMS. **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). **4.** The CSMS sends a RequestStartTransactionRequest with the IdToken from the previous step to the Charging Station. **5.** The Charging Station accepts the RequestStartTransactionRequest by sending a RequestStartTransactionResponse with Accepted. **7. The Charging Station starts charging and sends a TransactionEventRequest (eventType = Updated) to notify the CSMS that *chargingState* has changed.**

6.4.2. Remarks use case C05

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

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

6.5. Page 94 - 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 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 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.**

6.6. Page 105 - ChargingState in requirement C15.FR.03 [464]

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 The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND StopTxOnInvalidId is *true* AND 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 The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND StopTxOnInvalidId is *true* AND 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. Use of SuspendedEVSE in this situation will become deprecated in the next OCPP release.

7. Use case E Transactions

7.1. Page 115 - 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 - 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'.

7.2. Page 116 - Using *seqNo* in TransactionEventRequest when EVSE is not known [525]

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.

7.3. Page 126 - 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.

7.4. Page 126 - 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.	

7.5. Page 126 - Missing requirement about authorization during transaction [514]

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.19	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).

7.6. Page 130 - 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
Delete requirement	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

	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.	

7.7. Page 143 - 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 AND 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 AND (**Connection between Charging Station and EV is lost** **OR** **Authorization has ended or idToken is deauthorized)**	The Charging Station SHALL stop the transaction and send a TransactionEventRequest (eventType = Ended) to the CSMS.

7.8. Page 143 - Requirements for eventType=Started do not belong in use case E06/E07 [453]

7.8.1. Page 143 - 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 AND 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
E06.FR.17	When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration AND 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.

7.8.2. Page 145 - 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 AND 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
E07.FR.13	When configured to send meter data in the TransactionEventRequest (eventType = Started), See: Meter Values - Configuration AND 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.

7.9. Page 144 - 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 diagram shows the alternative sequences when TxStopPoint = Authorized or when TxStopPoint = EVConnected.

Figure 1 shows the sequence when the change in *chargingState* is reported as explicit TransactionEvents with *triggerReason* ChargingStateChanged. Figure 2 shows the sequence when the change in *chargingState* is reported as part of the *triggerReason* StopAuthorized and EVCommunicationLost.

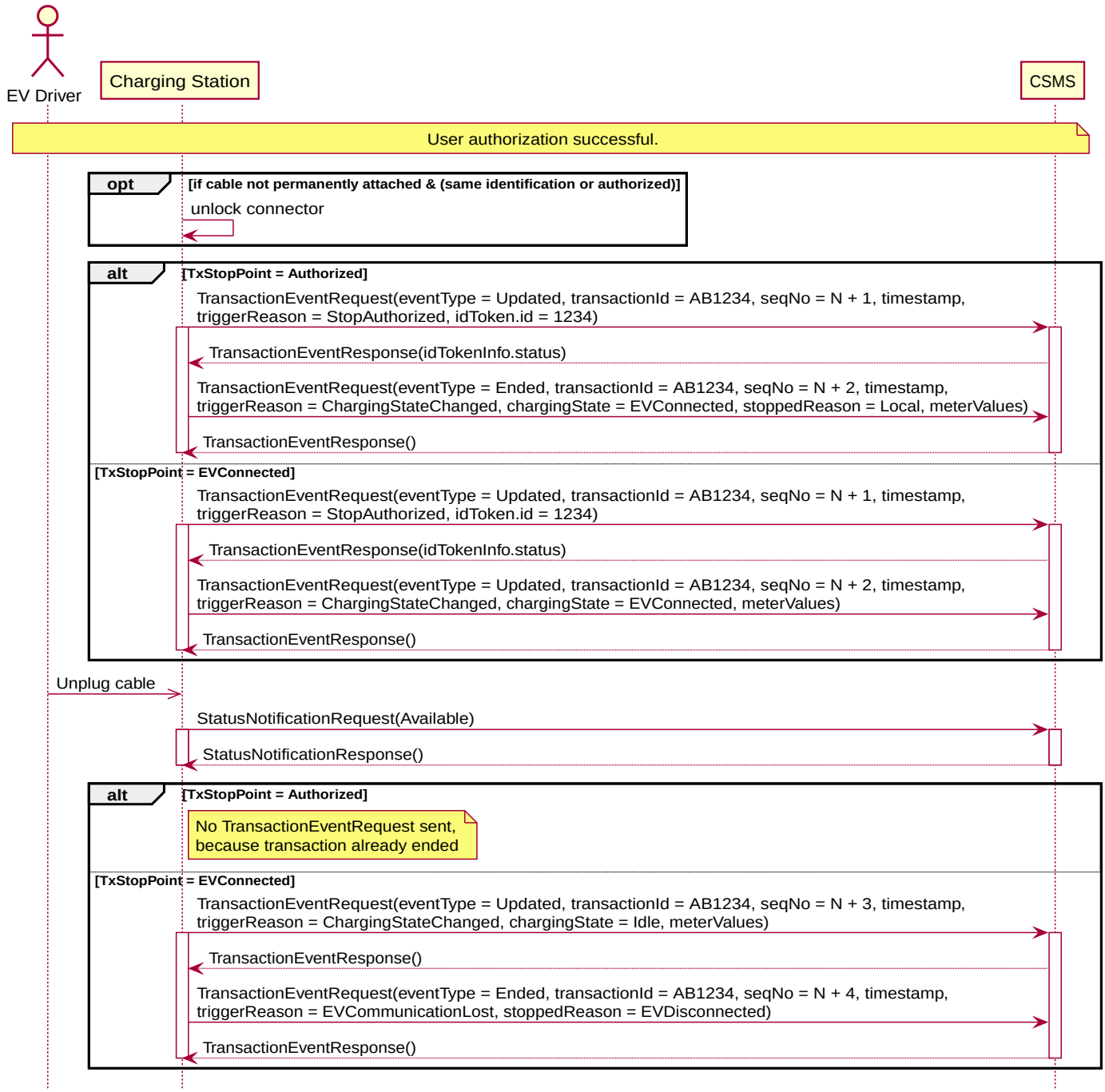


Figure 1. Sequence diagram with explicit ChargingStateChanged triggers

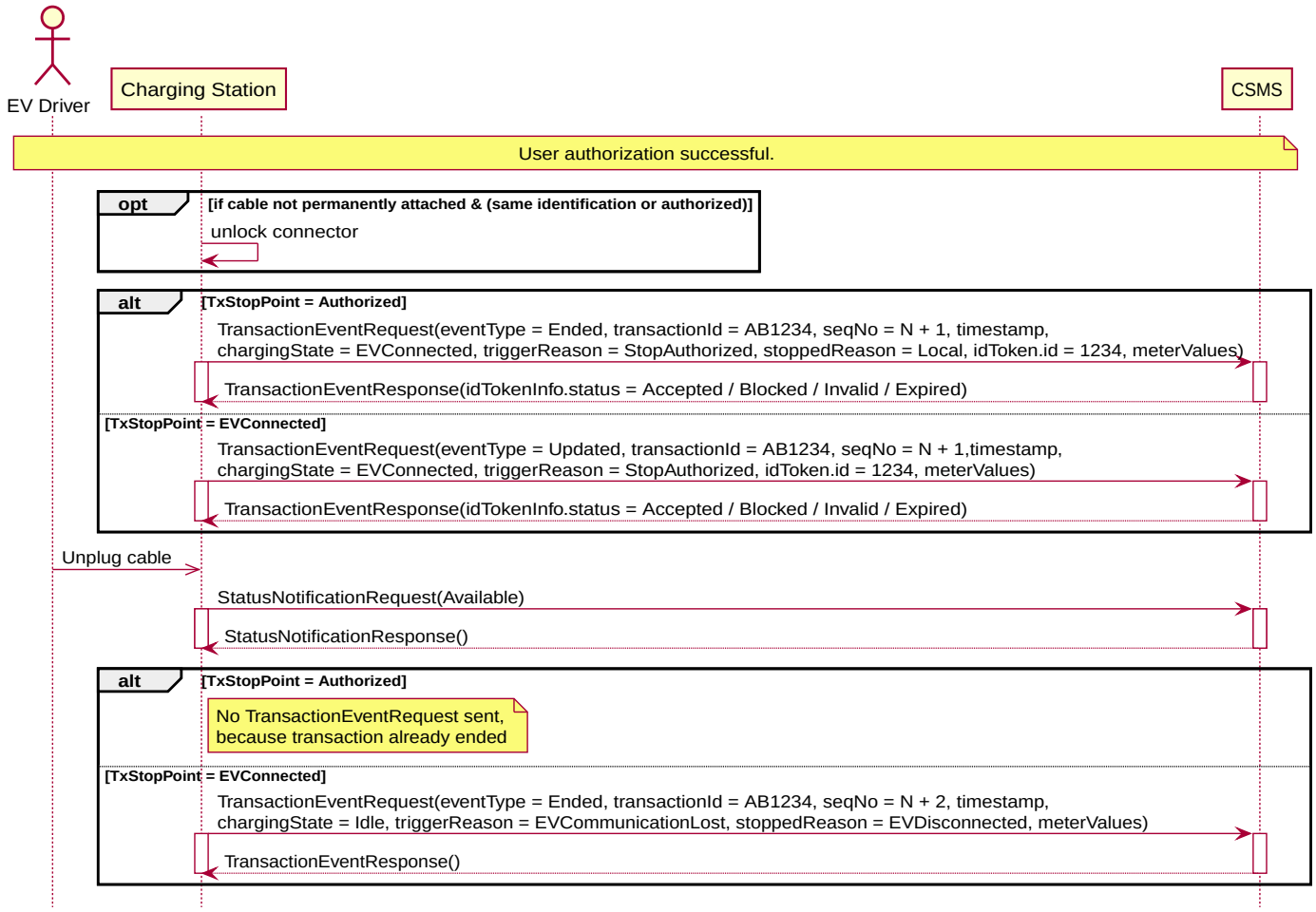


Figure 2. Sequence diagram with chargingState reported as part of other trigger

7.10. Page 162 - Misspelled field name in requirements [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 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 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 The transaction with that *transactionId* has not stopped yet	The Charging Station's response SHALL have *ongoing* = true.
New text	E14.FR.02	The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND The transaction with that *transactionId* has not stopped yet	The Charging Station's response SHALL have ***ongoingIndicator*** = true.

	ID	Precondition	Requirements
Old text	E14.FR.03	The Charging Station receives a GetTransactionStatusRequest with a *transactionId* AND 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 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.

8. Use case F Remote Control

8.1. Page 169 - 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 *evseId* 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 RequestStartTransaction Request

8.2. Page 170 - 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.	

8.2.1. Page 358 - 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.

8.3. Page 171 - Sequence diagram error [435]

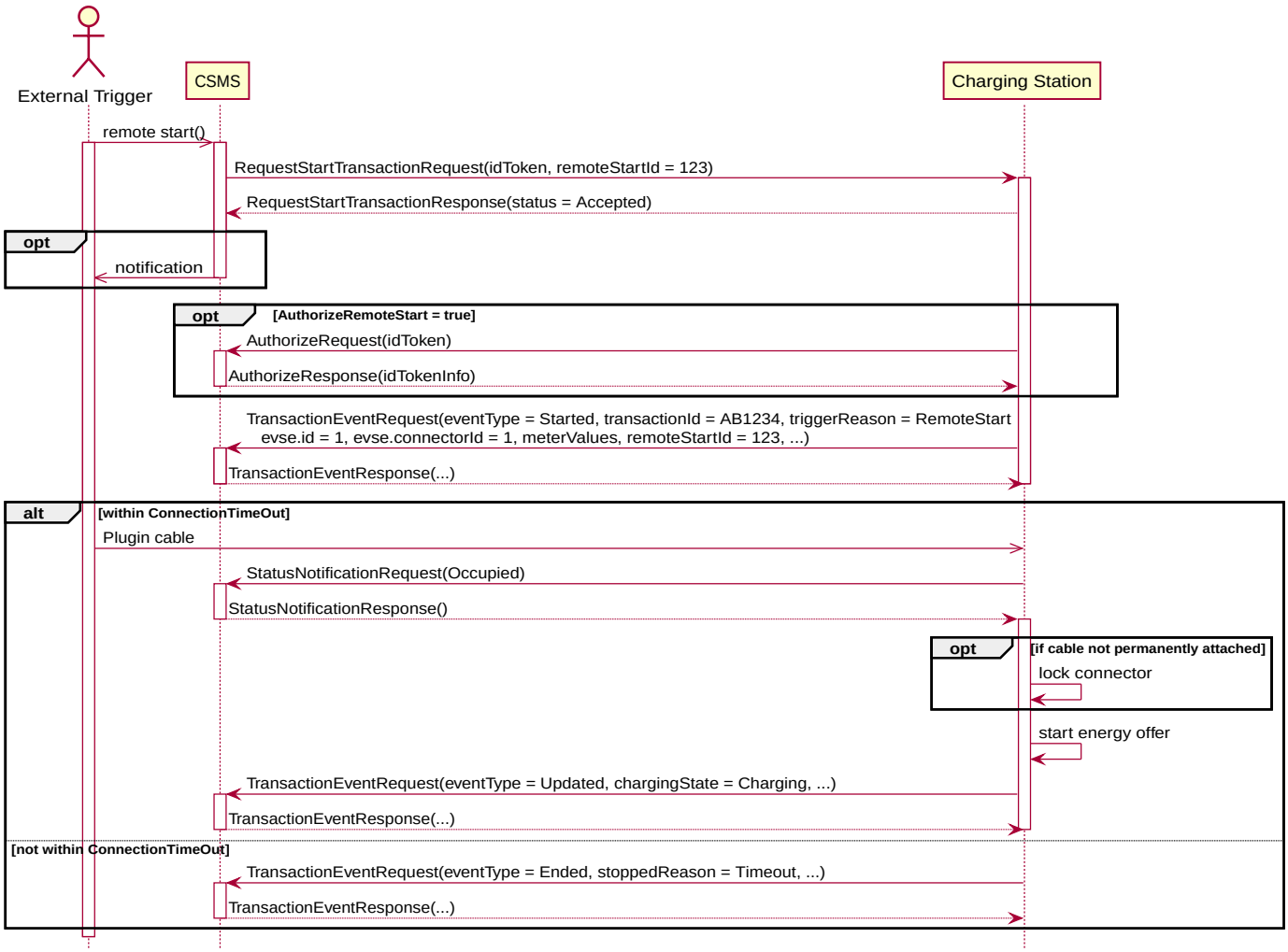
The sequence diagram on Figure 66 shows that a StatusNotification(Occupied) is sent after the RequestStartTransaction, but before the cable has been plugged in. There is no requirement that the connector should be reported occupied already at this moment. Step 5 and 6 have therefore been deleted from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

8.3.1. 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. **2.** The CSMS sends RequestStartTransactionRequest to the Charging Station. **3.** The Charging Station responds with RequestStartTransactionResponse to the CSMS. **4.** The EV Driver is authorized by the CSMS, dependent on the Configuration Variable settings. **5.** ~~The Charging Station sends StatusNotificationRequest to the CSMS to inform it about a Connector became *Occupied*.~~ **6.** ~~The CSMS sends StatusNotificationResponse to the Charging Station~~ **7.** The Charging Station sends a TransactionEventRequest (eventType = Started) notifying the CSMS about a transaction that has started **8.** The cable is plugged in. **8a. Charging Station sends a StatusNotificationRequest with *Occupied*.** **8b. CSMS sends a StatusNotificationResponse to the Charging Station** **9.** The energy offer is started. **10.** The Charging Station sends a TransactionEventRequest (eventType = Updated, chargingState = Charging) message to inform the CSMS that the charging has started. **11.** The CSMS sends TransactionEventResponse to the Charging Station

8.3.2. Use case F02 sequence diagram



8.4. Page 172 - Some requirements of use case E03 apply also to F02 [441]

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.	
F02.FR.08	F02.FR.07	The Charging Station SHALL deauthorize the transaction and send a TransactionEventRequest (*triggerReason* = `EVConnectionTimeout`) to the CSMS.	

8.5. Page 180 - Requirement F06.FR.06 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`.	

8.6. Page 180 - 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**

8.7. Page 180 - 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: *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: *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.

9. Use case G Availability

9.1. Page 188 - 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 - 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
H01.FR.13	When receiving a ReserveNowRequest AND (all) targeted EVSEs have status *Occupied*	The Charging Station SHALL return *Occupied*.	

10.2. Page 197 - 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 - 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 AND 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** 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 requirement

ID	Precondition	Requirement definition	Note
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.

11. Use case I Tariff And Cost

11.1. Page 207 - 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. **2.** The Charging Station shows the current total cost to the EV Driver.

11.1.1. Page 208 - 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(Updated)	The Charging Station SHALL show the current running cost to the EV Driver.

11.1.2. Page 367 - 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*. 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 - 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 216 - 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.

12.2.1. Page 439 - 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. 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).		

12.3. Page 218 - 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.

12.4. Page 218 - Wrong placement of Aligned/SampledDataSignReadings

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

12.4.1. Page 218 - Requirement J01.FR.16 belongs to use case J02

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

12.4.2. Page 220 - Requirement J02.FR.15 belongs to use case J01

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

12.5. Page 221 - 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.

12.5.1. Use case J03 row #7

No.	Type	Description
7	**Combined scenario description**	**15118** **1.** The EV sends a ChargingStatusReq (in case of AC charging) message to the Charging Station. **2.** The EV sends a MeteringReceiptReq to the Charging Station. **OCPP** **3.** Between the Charging Station and the CSMS, the TransactionEventRequest(eventType = Updated) message is being exchanged. 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** **1.** The EV sends a ChargingStatusReq (in case of AC charging) message to the Charging Station. **2.** The EV sends a MeteringReceiptReq to the Charging Station. **OCPP** **3.** Between the Charging Station and the CSMS, the TransactionEventRequest(eventType = Updated) message is being exchanged.
New text	**15118** **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. **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. **2.** The EV sends a MeteringReceiptReq to the Charging Station to acknowledge receipt of the meter value. ~~**OCPP**~~ ~~**3.** Between the Charging Station and the CSMS, the TransactionEventRequest(eventType = Updated) message is being exchanged.~~

12.5.2. 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.

12.5.3. 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 233 - Use case K01: Recommendation to not limit duration of TxProfile [519]

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.

13.2. Page 234 - 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.	**ChargingStationExternalConstraints profile cannot be replaced.**

13.3. Page 236 - 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.

13.4. Page 245 - 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)**	[...] 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.

13.4.1. Page 373 - 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.

13.5. Page 250 - 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** 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)

13.6. Page 251 - 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.	
New text	K09.FR.03		The CSMS SHALL specify in *chargingProfile* criteria in GetChargingProfilesRequest either: - a (list of) *chargingProfileId(s)* OR - 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 `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).
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*.**	

13.6.1. Page 346 - 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.

	Field Name	Field Type	Card.	Description
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*.**

13.7. Page 252 - 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 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 **NOT K10.FR.08 AND** 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 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 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.

13.8. Page 253 - 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 NotifyEVChargingNeedsRequest 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 NotifyEVChargingNeedsRequest 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.

13.9. Page 254 - 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.

13.10. Page 259 - 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:

13.10.1. Usecase K15

At Table 188.

Requirements are added for above-mentioned recommendations.

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 AND 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.

13.10.2. Usecase K16

At table of requirements

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 AND Charging Station sends a NotifyEVChargingScheduleRequest	The CSMS SHALL send a SetChargingProfileRequest.	[..]

Version	ID	Precondition	Requirements	Note
New	K16.FR.12	K16.FR.09 AND Charging Station sends a **NotifyEVChargingNeeds Request**	The CSMS SHALL send a SetChargingProfileRequest.	[..]

13.10.3. Usecase K17

At Table 191.

New requirement

ID	Precondition	Requirements	Note
K17.FR.16	K17.FR.07 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.

13.11. Page 260 - 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.

13.11.1. Page 260 - 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.**

13.12. Page 260 - 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.**

13.13. Page 261 - 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.
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 273 - 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`	

14.2. Page 274 - Changed note of L01.FR.13 [456]

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.**

14.3. Page 275 - 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.

New requirement

ID	Precondition	Requirement definition	Note
L01.FR.29	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.

15. Use case M ISO15118 Certificate Management

15.1. Page 288 - 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.

15.2. Page 292 - 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.

15.2.1. Page 292 - M01

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

15.2.2. Page 292 - 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."

15.2.3. Page 293 - M02

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

15.2.4. Page 294 - 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."

15.3. Page 294 - 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

15.4. Page 296 - 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 **OR 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.**

Changed description

Version	Description
Old	Processing failure.
New	**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.**

15.5. Page 296 - 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 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.**

16. Use case N Diagnostics

16.1. Page 303 - 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 - 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	

ID	Precondition	Requirement definition	Note
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. Basic authorization can be added to the URL as follows: http://*username*:*password*@csms.org/logs
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: Content-Type: application/octet-stream Content-Disposition: form-data; name="uploadedfile"; filename="logfile_20210420.zip"

16.3. Page 305 - 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** When the Charging Station receives a getMonitoringReportRequest for supported *monitoringCriteria* OR without *monitoringCriteria*	The Charging Station SHALL send a getMonitoringReportResponse with Accepted.

16.4. Page 307 - 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**.	

16.5. Page 308 - 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.

16.6. Page 308 - 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 without an Id **AND N04.FR.08**	The Charging Station will generate an Id and return it in the SetVariableMonitoringResponse.	

16.7. Page 308 - 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 AND 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 with an Id AND a monitor exists matching the given Id AND** 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.

16.8. Page 310 - 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.

16.9. Page 311 - 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.

16.10. Page 312 - 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 A reboot occurred AND 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.

	ID	Precondition	Requirement definition
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.

16.11. Page 312 - Added remark 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 remark is added to these requirements, as follows:

Changed requirements

ID	Precondition	Requirement definition	Remark
N07.FR.16	When there is a monitor with type UpperThreshold on a Component/Variable combination AND 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 AND 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.

16.12. Page 312 - 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 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.
N07.FR.21	When there is a monitor with type PeriodicClockAligned on a Component/Variable combination AND 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**

16.12.1. Page 313 - 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** AND The severity number of the monitor is equal to or lower than the severity number set in the Configuration Variable `OfflineMonitoringEventQueueingSeverity` AND 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 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 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 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 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. Messages

17.1. Page 344 - 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.

17.2. Page 351 - 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. Use the message CertificateSignedRequest for that."

17.3. Page 365 - 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. Datatypes

18.1. Page 380 - 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).

18.2. Page 386 - 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. **The value is allowed to be an empty string ("").** The Configuration Variable ConfigurationValueSize can be used to limit SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal.

19. Enumerations

19.1. Page 400 - 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**

19.2. Page 404 - 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).

19.3. Page 409 - 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, as shown below:

Value	Description
OCPP12	OCPP version 1.2
OCPP15	OCPP version 1.5
OCPP16	OCPP version 1.6
OCPP20	OCPP version 2.0

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 OCPP 2.0.1 does have its own Websocket subprotocol name: `ocpp2.0.1`.

20. Referenced Components and Variables

20.1. Page 420 - 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.

20.2. Page 420 - 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.

20.2.1. (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**	**supportsMonitoring**
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."

20.3. Page 422 - 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.**

20.4. Page 426 - Variable ClockCtrlr.TimeAdjustmentReportingThreshold [492]

The optional variable ClockCtrlr.TimeAdjustmentReportingThreshold needs to be mentioned here, because it occurs as a SecurityEvent "SettingSystemTime".

20.4.1. `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.		

20.5. Page 427 - 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. This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.		

20.6. Page 428 - 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	[…] ***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.***		

20.7. Page 429 - 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`.

20.8. Page 430 - 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.

`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. 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.		

20.9. Page 431 - 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

20.9.1. 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.		

20.9.2. 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.		

20.10. Page 434 - 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. 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`. 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. 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.

20.11. Page 435 - 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.

20.12. Page 439 - 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.**

20.13. Page 445 - 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**	MonitoringBase	
	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**	MonitoringLevel	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	Shows the currently used MonitoringLevel. Valid values are severity levels of SetMonitoringLevelRequest: 0-9.		

20.14. Page 446 - 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.

20.15. Page 447 - 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`.**

20.16. 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.

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: “<uri>,<major>,<minor>”. This is 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"		

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	ProtocolAgreed	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	A string with the following comma-separated items: “<uri>,<major>,<minor>”. This is the protocol uri and version information that was agreed upon between EV and EVSE in the supportedAppProtocolReq handshake from ISO 15118. Example: "urn:iso:15118:2:2013:MsgDef,2,0"		

20.17. Page 449 - New configuration variables for ISO15118Ctrlr

20.17.1. `ISO15118CtrlrEvseId` [180, 433]

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.

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.		

20.17.2. `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 is enabled. If this variable is *false*, then the Charging Station won't initiate ISO 15118 CSRs.		

20.17.3. `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. Appendix 3

21.1. Page 8 - 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.

21.2. Page 8 - 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.

21.3. Page 9 - 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. This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.

21.4. Page 14 - 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) *Six bytes, represented as hexbinary encoded string, e.g. "010203040A0B"*	Vehicle ID (H'710 + H'711 + H'712) *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) *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) EAmount (AC)	-
DepartureTime	dateTime	DepartureTime *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	-

Variable	Unit	ISO 15118-2 value	CHAdeMO value
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 *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	-

22. Appendix 5 Reason Codes

22.1. 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)

23. 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

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
<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.
<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.

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
<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 %.
<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

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
<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
<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.

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
<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.
<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.

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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.
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 authorization is switched off.
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.
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)

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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.
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.
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.

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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
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.

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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	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.
FiscalMetering	EnergyExport		no	decimal	Wh, kWh	Total energy transferred: e.g. from EV during (ongoing or terminated) charging session (in wH by default)
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	EvseId		no	string		The ID of the EVSE in string format as defined by ISO15118.
ISO15118Ctrlr	MaxScheduleEntries		no	integer		Maximum number of allowed schedule periods.
ISO15118Ctrlr	ProtocolAgreed		no	string		A comma-separated string with 'uri,major,minor' of agreed ISO 15118 protocol between EV and EVSE.
ISO15118Ctrlr	ProtocolSupportedEV	<Priority>	no	string		A comma-separated string with 'uri,major,minor' of supported ISO 15118 protocols by EV.
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.
LocalAuthListCtrlr	Available		no	boolean		Local Authorization List is available.
LocalAuthListCtrlr	BytesPerMessage		yes	integer		Maximum number of bytes in a SendLocalList message.
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

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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.
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	no	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)
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.
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

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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 Charging Station side when the cable is unplugged at the EV
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.
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
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)

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
SecurityCtrlr	AdditionalRootCertificateCheck		no	boolean		Required for all security profiles except profile 1.
SecurityCtrlr	BasicAuthPassword		no	string		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	string		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.
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		Max StackLevel of a ChargingProfile. The number defined also indicates the max allowed number of installed charging schedules per Charging Profile Purposes.
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.

Specific Component	Variable	Instance	Required?	Data Type	Unit	Description
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).
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