OCPP 2.0

Part 2 - Errata

v1.1, 2020-03-31

1. Disclaimer

Copyright © 2010 – 2020 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
1.1	2020-03-31	Franc Buve (*OCA*) Milan Jansen (*OCA*)	The final version of the OCPP 2.0 errata. The errata from this version of the document have been merged into the specification to create the OCPP 2.0.1 specification.
1.0	2019-12-20	Franc Buve (*OCA*) Milan Jansen (*OCA*) Robert de Leeuw (*ihomer*)	Version 1.0

2. Scope

This document contains errata on "part 2: Specification" of the OCPP 2.0 documentation.

All these errata have been merged in the documentation, which has resulted in the release of OCPP 2.0.1.

2.1. Terminology and Conventions

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

3. Erratas

3.1. (Minor) - Page 3, Section 1, Scope text unclear about mandatory JSON

Old text	The specification does not define the communication technology. Any technology will do, as long as it supports the message structures and communication patterns described in this specification.
New text	This part of the specification does not define the communication technology. In order to ensure widespread compatibility OCPP 2.0 is limited to JSON. In order to avoid clutter the specifications for the JSON implementation are in "Part 4 - JSON over WebSockets implementation guide".

3.2. (Major) - Page 4, Appendix 3.1, Controller Components

The statement in Appendix 3, section 3.1, that "it is NOT allowed to implement additional *Controller* components/variables" is incorrect. This sentence, starting with "Please note:" and ending with "components/variables" shall be removed.

3.3. (Medium) - Page 4, section 2.1.3, Primitive Datatypes, string incorrectly refers to the Unicode character set

The Primitive DataType 'string' must use the UTF-8 character set, NOT the Unicode character set.

Changed description 'string':

Old Description	The characters defined in the ***Unicode*** character set are allowed to be used.
New Description	The characters defined in the ***UTF-8*** character set are allowed to be used.

3.4. (Minor) - Page 4, section 2.1.3, Primitive Datatypes, Not specified if identifierString is case-sensitive or case-insensitive

The Primitive DataType 'identifierString' must be case-insensitive.

Changed description 'identifierString':

Old Description	Only the following character set is allowed: a-z, A-Z, 0-9, '*', '-', '_', '=', ':', '+', '@', '.', '|'
New Description	**This is a case-insensitive dataType and can only contain characters from the following character set:** a-z, A-Z, 0-9, '*', '-', '_', '=', ':', '+', '@', '.', '|'

3.5. (Major) - Page 14, The specification is unclear about how to respond to certain types of "invalid" messages

It is clear that a system needs to send a RPC framework CALLERROR when a message is invalid according to the JSON schema. However it is not clear whether a system needs to respond with a RPC framework CALLERROR or a CALLRESULT with (for example) the status *Rejected*, in the case a message in invalid on the protocol level (For example; requirements, configured limits, etc.) Often implementers are choosing to sent a RPC framework CALLERROR, because this way they are able to provide additional information about why the message was invalid or rejected. Below requirements are created to clarify when to sent which kind of response. For incorrect messages on the protocol level the CALLRESULT needs to be used. To accommodate the need for providing additional information, a statusInfo dataType will be included to every response message containing a status field. The statusInfo dataType contains a free format additionalInfo field and a reasonCode field for which a standardized list of reason codes will be used. This list can be found at part 2 appendices.

New requirements:

Page	ID	Precondition	Requirement definition	Note
14	FR.02	When the Charging Station receives a valid OCPP request message according to the JSON schemas / RPC Framework AND the other system is not causing a security violation	The Charging Station SHALL respond with a RPC Framework: CALLRESULT.	If the Charging Station/CSMS needs to provide additional information, this can be done in the statusInfo element of the response message.
14	FR.03	When the Charging Station/CSMS receives an invalid OCPP message according to the JSON schemas / RPC Framework OR the other system causes a security violation	The Charging Station/CSMS SHALL respond with a RPC Framework: CALLERROR.	
14	FR.04	When the CSMS did not accept the BootNotificationRequest from the Charging Station AND The Charging Station sends a message other than BootNotificationRequest	The CSMS SHALL respond with a RPC Framework: CALLERROR: SecurityError.	

New DataType 'StatusInfoType':

Field	Type	Card.	Description
reasonCode	string[0..20]	1..1	Required. A predefined code for the reason why the status is returned in this response. The string is case-insensitive.
additionalInfo	string[0..512]	0..1	Optional. Additional text to provide detailed information.

New Schema files are available.

During the addition of the StatusInfoType, below enum values were removed. Because these are just rejection reasons for not being able to set a properly addressed variable. That kind of information can be provided at the StatusInfoType. The addressing errors are kept at the StatusEnumTypes.

Removed Enum values:

Page	section	Message/dataType	Value
371	2.74	SetMonitoringStatusEnumType	OutOfRange
372	2.76	SetVariableStatusEnumType	OutOfRange
372	2.76	SetVariableStatusEnumType	InvalidValue

Changed requirements:

Version	Id	Precondition	Requirement definition	Note
Old	B05.FR.07	When the Charging Station receives a SetVariablesRequest with a *value* that is incorrectly formatted for the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: InvalidValue.	
New	B05.FR.07	When the Charging Station receives a SetVariablesRequest with a *value* that is incorrectly formatted for the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: **Rejected**. **(More information can be provided in the optional *statusInfo* element.)**	
Old	B05.FR.08	When the Charging Station receives a SetVariablesRequest with a *value* that is lower or higher than the range of the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: OutOfRange.	
New	B05.FR.08	When the Charging Station receives a SetVariablesRequest with a *value* that is lower or higher than the range of the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: **Rejected**. **(More information can be provided in the optional *statusInfo* element.)**	

Version	Id	Precondition	Requirement definition	Note
Old	B10.FR.02	On receipt of a SetVariablesRequest, containing Configuration Variable `NetworkConfigurationPriority`.	The Charging Station SHALL validate if the requested NetworkProfile slots contain valid configurations. If not, the Charging Station SHALL send SetVariablesResponse with status *InvalidValue*.	
New	B10.FR.02	On receipt of a SetVariablesRequest, containing Configuration Variable `NetworkConfigurationPriority` AND any of the NetworkProfile slots in the message does not contain a valid configuration	The Charging Station SHALL send SetVariablesResponse with status **Rejected**.	**The optional element *statusInfo* can be used to provide more information.**
Old	N04.FR.06	When the Charging Station receives a SetVariableMonitoringRequest with monitor type UpperThreshold or LowerThreshold AND the *monitorValue* is lower or higher than the range of the given Variable	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: OutOfRange.	
New	N04.FR.06	When the Charging Station receives a SetVariableMonitoringRequest with monitor type UpperThreshold or LowerThreshold AND the *monitorValue* is lower or higher than the range of the given Variable	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: **Rejected**.	**More information can be provided in the optional *statusInfo* element.**
Old	N04.FR.14	When the Charging Station receives a SetVariableMonitoringRequest with type Delta and value contains a negative value.	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: OutOfRange.	
New	N04.FR.14	When the Charging Station receives a SetVariableMonitoringRequest with type Delta and value contains a negative value.	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: **Rejected**.	**More information can be provided in the optional *statusInfo* element.**

3.6. (Minor) - Page 14, Missing standard Variable for MessageTimeout

Almost every OCPP implementation seems to have a configuration variable that is used to define the MessageTimeout.
Page 14 has section 3.2. Message Timeouts. But no variable is defined for this.

Configuration variable 'DefaultMessageTimeout':

Required	yes		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	MessageTimeout	
	variableInstance	Default	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**unit**	seconds
		dataType	integer
Description	The purpose of the message timeout is to be able to consider a request message as not sent and continue with other tasks when the message did not arrive due to communication errors or software failure. The message timeout setting in a Charging Station can be configured in the messageTimeout field in the *NetworkConnectionProfile*.		

3.7. (Major) - Page 18, Security requirement about installing a new root CA certificate could cause implementation issues

The specification describes that when the root certificate of the CPO hierarchy is updated, the new root certificate needs to be signed by the old, so that its authenticity can be verified. (A00.FR.605) However normal practice is that the root certificate is self-signed. This can create two practical problems:

- Some TLS libraries may only work with self-signed certificates.
- External certificate authorities (CA) normally will only provide self-signed root certificates.

Removed requirements:

ID	Precondition	Requirement definition
A00.FR.605	If a new CSO root certificate is installed	The new CSO root certificate MUST be signed with the private key belonging to the public key of the old root certificate.
A00.FR.606	A00.FR.605	The Charging Station MUST check the validity of the new certificate using this signature.

3.7.1. Allow implementers to add an additional security check for a new Root certificate

It should be possible to implement "removed requirement A00.FR.605", for topologies where the CSMS also has the role of Certificate Authority. In this case it is possible to sign the new Root certificate with the old one.

New Configuration Variable 'AdditionalRootCertificateCheck':

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	AdditionalRootCertificateCheck	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	When set to true, only one certificate (plus a temporarily fallback certificate) of certificateType CSMSRootCertificate is allowed to be installed at a time. When installing a new CSMS Root certificate, the new certificate SHALL replace the old one AND the new CSMS Root Certificate MUST be signed by the old CSMS Root Certificate it is replacing. This configuration variable is required unless only "security profile 1 - Unsecured Transport with Basic Authentication" is implemented. Please note that security profile 1 SHOULD only be used in trusted networks. *Note: When using this additional security mechanism please be aware that the Charging Station needs to perform a full certificate chain verification when the new CSMS Root certificate is being installed. However, once the old CSMS Root certificate is set as the fallback certificate, the Charging Station needs to perform a partial certificate chain verification when verifying the server certificate during the TLS handshake. Otherwise the verification will fail once the old CSMS Root (fallback) certificate is either expired or removed.*		

New requirements:

ID	Precondition	Requirement definition
M05.FR.09	When `AdditionalRootCertificateCheck` is true	Only one certificate (plus a temporarily fallback certificate) of certificateType CSMSRootCertificate is allowed to be installed at a time.
M05.FR.10	When `AdditionalRootCertificateCheck` is true AND installing a new certificate of certificateType CSMSRootCertificate	The new CSMS Root certificate SHALL replace the old CSMS Root certificate AND the new Root Certificate MUST be signed by the old Root Certificate it is replacing
M05.FR.11	M05.FR.10 AND the new CSMS Root certificate is NOT signed by the old CSMS Root certificate	The Charging Station SHALL NOT install the new CSMS Root Certificate and respond with status *Rejected*.
M05.FR.12	M05.FR.10 AND the new CSMS Root certificate is signed by the old CSMS Root certificate	The Charging Station SHALL install the new CSMS Root Certificate AND temporarily keep the old CSMS Root certificate as a fallback certificate AND respond with status *Accepted*

ID	Precondition	Requirement definition
M05.FR.13	M05.FR.12 AND the Charging Station successfully connected to the CSMS using the new CSMS Root certificate	The Charging Station SHALL remove the old CSMS Root (fallback) certificate.
M05.FR.14	M05.FR.12 AND The Charging Station is attempting to reconnect to the CSMS (NOT migrating to another CSMS with Use Case B10 - Migrate to new CSMS), but determines that the server certificate provided by the CSMS is invalid when using the new CSMS Root certificate to verify it	The Charging Station SHALL try to use the old CSMS Root (fallback) certificate to verify the server certificate.
M05.FR.15	M05.FR.12 AND When the Charging Station is migrating to another CSMS with Use Case B10 - Migrate to new CSMS, but determines that the server certificate provided by the CSMS is invalid when using the new CSMS Root certificate to verify it	The Charging Station SHALL use the `NetworkProfileConnectionAttempts` mechanism as described at Use Case B10 - Migrate to new CSMS .
M05.FR.16	M05.FR.15 AND If after the number of attempts the connection fails AND If it goes back to the old NetworkConnectionProfile (*See B10.FR.03*)	The Charging Station SHALL use the old CSMS Root (fallback) certificate to verify the server certificate.

Removed requirement:

ID	Precondition	Requirement definition
M05.FR.05	When a new certificate gets installed AND a certificate of the same type is already installed in the Charging Station.	The Charging Station SHALL NOT overwrite the already installed certificate.

3.8. (Minor) - Page 19, The Charging Station identity may not contain ":" because of Basic Authentication

Basic Authentication uses the ":" as a separator between the username and password, but if both the password and username are able to contain this separator character, then the CSMS is not able to tell the difference anymore between the username and password section of the username/password combination.

Changed requirements 'A00.FR.204 & A00.FR.303':

Old requirement definition	The username SHALL be equal to the Charging Station identity, which is the identifying string of the Charging Station as it uses it in the OCPP-J connection URL.
New requirement definition	The username SHALL be equal to the Charging Station identity, which is the identifying string of the Charging Station as it uses it in the OCPP-J connection URL. **When using Basic Authentication, the Charging Station identity may not contain the character ":". Otherwise the CSMS may be unable to separate the username from the password.**

3.9. (Medium) - Page 20, Section 1.3.5, Unclear how to implement BasicAuthPassword

Changed requirements:

Version	Id	Precondition	Requirement definition
Old	A00.FR.205	A00.FR.203	The password SHALL be a 20-byte key stored in the BasicAuthPassword Configuration Variable.

Version	Id	Precondition	Requirement definition
New	A00.FR.205	A00.FR.203	The password SHALL be **stored in the `BasicAuthPassword` Configuration Variable. At least 16 bytes and at most 20 bytes long, It is strongly advised to be randomly generated binary to get maximal entropy. In hexadecimal representation (20 bytes maximum, represented as a string of up to 40 hexadecimal digits).**
Old	A00.FR.304	A00.FR.302	The password SHALL be a 20-byte key stored in the BasicAuthPassword Configuration Variable.
New	A00.FR.304	A00.FR.302	The password SHALL be **stored in the `BasicAuthPassword` Configuration Variable. At least 16 bytes and at most 20 bytes long, It is strongly advised to be randomly generated binary to get maximal entropy. In hexadecimal representation (20 bytes maximum, represented as a string of up to 40 hexadecimal digits).**

Changed description configuration variable 'BasicAuthPassword':

Old text	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	The basic authentication password is used for HTTP Basic Authentication. The password 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). 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.

Changed DataType configuration variable 'BasicAuthPassword' & 'Identity':

Old dataType	New dataType
String	identifierString

3.10. (Medium) - Page 20, A00, Use FQDN for `commonName` instead of URL

The following requirements about `commonName` must be updated, as follows:

Section	Page	ID	Precondition	Requirement definition
1.3.5	20	A00.FR.309		The Charging Station SHALL verify that the `commonName` includes the CMSM's **FQDN**.
1.3.7	22	A00.FR.412		The Charging Station SHALL verify that the `commonName` includes the CSMS's **FQDN**.
1.4.1	24	A00.FR.510		For the CSMS certificate, the subject field SHALL contain the **FQDN** of the server in the `commonName` RDN.

3.11. (Minor) - Page 20, Functional Block A, Security Profile 2 & 3, Incorrect remark

Below remark is no longer valid for OCPP 2.0, 2.0 only supports JSON, so the CSMS is always the server, never the client.

Page	Section	Old remark	Action
20	1.3.4	When CSMS needs to act as client, it's not possible to authenticate the Charging Station it connects to.	removed
22	1.3.6	When CSMS needs to act as client, it's not possible to authenticate the Charging Station it connects to.	removed

3.12. (Minor) - page 20, section 1.3.4: Unclear how to implement Basic Authentication in combination with WebSockets

For the implementation of Basic Authentication the spec refers to RFC 2617 and for setting up a WebSocket connection the spec refers to RFC 6455. However the specification does not describe how these two coincide.

The Charging Station should include the same header as used in Basic Auth RFC 2617, while requesting to upgrade the http connection to a websocket connection as described in RFC 6455. The server first needs to validate the Authorization header before upgrading the connection.

Example:

```
GET /ws HTTP/1.1
Remote-Addr: 127.0.0.1
UPGRADE: websocket
CONNECTION: Upgrade
HOST: 127.0.0.1:9999
ORIGIN: http://127.0.0.1:9999
SEC-WEBSOCKET-KEY: Pb4obWo2214EfaPQuazMjA==
SEC-WEBSOCKET-VERSION: 13
AUTHORIZATION: Basic <Base64 encoded(<ChargePointId>:<AuthorizationKey>)>
```

3.13. (Minor) - Page 20, Section 1.3.5, Requirement not applicable for Security Profile 2

Below requirement is not relevant when using a TLS connection.

Removed requirement:

ID	Precondition	Requirement definition
A00.FR.305	A00.FR.302	With HTTP Basic, the username and password are transmitted in clear text, encoded in base64 only. Hence, it is RECOMMENDED that this mechanism will only be used over connections that are already secured with other means, such as VPNs.

3.14. (Minor) - Page 21, Requirements (A00.FR.318, A00.FR.319) and (A00.FR.421, A00.FR.422) cause that the CSMS needs to have two CSMSCertificates

Requirements (A00.FR.318, A00.FR.319) and (A00.FR.421, A00.FR.422) cause that the CSMS needs to have two CSMSCertificates, one for each signature type. But this is not immediately clear. So this needs to be specifically mentioned.

Changed requirement 'A00.FR.318' & 'A00.FR.421' :

Old requirement definition	The CSMS SHALL support at least the following four cipher suites: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384
New requirement definition	The CSMS SHALL support at least the following four cipher suites: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384 **Note: The CSMS will have to provide 2 different certificates to support both cipher suites. Also when using security profile 2, the CSMS should be capable of generating client side certificates for both cipher suites.**

3.15. (Minor) - Page 21, ISO15118-2 prescribes to implement different cipher suites for the communication between EV and Charging Station

ISO15118-2 prescribes to implement different cipher suites for the communication between EV and Charging Station. A note needs to be added which describes that this has to be taken into account when implementing ISO15118-2.

Changed requirement 'A00.FR.319' & 'A00.FR.422' :

Old requirement definition	The Charging Station SHALL support at least the cipher suites: (**TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** AND **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**) OR (**TLS_RSA_WITH_AES_128_GCM_SHA256** AND **TLS_RSA_WITH_AES_256_GCM_SHA384**) Note: TLS_RSA does not support forward secrecy, therefore TLS_ECDHE is RECOMMENDED. Furthermore, if the Charging Station detects an algorithm used that is not secure, it SHOULD trigger an InvalidTLSCipherSuite security event (See part 2 appendices for the full list of security events).
New requirement definition	The Charging Station SHALL support at least the cipher suites: (**TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** AND **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384**) OR (**TLS_RSA_WITH_AES_128_GCM_SHA256** AND **TLS_RSA_WITH_AES_256_GCM_SHA384**) **Note 1**: TLS_RSA does not support forward secrecy, therefore TLS_ECDHE is RECOMMENDED. Furthermore, if the Charging Station detects an algorithm used that is not secure, it SHOULD trigger an InvalidTLSCipherSuite security event (See part 2 appendices for the full list of security events). **Note 2: Please note that ISO15118-2 prescribes to implement the following cipher suites for the communication between EV and Charging Station:** **TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,** **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256**

3.16. (Minor) - Page 22, OCPP Security, Improved definition of requirements

Requirement A00.FR.405 and A00.FR.511 are conflicting. A00.FR.405 states the commonName should contain the unique identifier of the Charging Station and A00.FR.511 states it should be the serialNumber. The commonName needs to contain the serialNumber of the Charging Station, because the unique identifier of the Charging Station may be changed and the SerialNumber will always remain the same. The following changes have to be made:

In table 18 change the following requirements:

	ID	Precondition	Requirement definition
Old	A00.FR.404		The CSMS SHALL verify in the subject field that the certificate is owned by the CSO or an organization trusted by the CSO
New	A00.FR.404		The CSMS SHALL verify that the certificate is owned by the CSO (or an organization trusted by the CSO) by checking that the O (`organizationName`) RDN in the subject field of the certificate contains the CSO name.

	ID	Precondition	Requirement definition
Old	A00.FR.405		The CSMS SHALL verify that the commonName field contains the unique identifier of the Charging Station (see [certificate_properties])
New	A00.FR.405		The CSMS SHALL verify that the certificate belongs to this Charging Station by checking that the CN (commonName) RDN in the subject field of the certificate contains the unique serial number of the Charging Station (see [certificate_properties]).

In table 20 change the following requirements:

	ID	Precondition	Requirement definition
Old	A00.FR.509		The subject field of the certificate SHALL contain the organization name of the certificate owner in the organizationName RDN.
New	A00.FR.509		The subject field of the certificate SHALL contain the organization name of the certificate owner in the O (organizationName) RDN.
Old	A00.FR.511		For the Charging Station certificate, the subject field SHALL contain a commonName RDN which consists of a unique serial number for the Charging Station. This serial number SHALL NOT be in the format of a URL or an IP address so that Charging Station certificates can be differentiated from CSMS certificates.
New	A00.FR.511		For the Charging Station certificate, the subject field SHALL contain a CN (commonName) RDN which consists of the unique serial number of the Charging Station. This serial number SHALL NOT be in the format of a URL or an IP address so that Charging Station certificates can be differentiated from CSMS certificates.

New requirement:

ID	Precondition	Requirement definition
A00.FR.429	When Charging Station supports Security Profile 3	The manufacturer is required to give every Charging Station a unique Serial Number.

3.17. (Minor) - Page 24, requirement A00.FR.511: subjectAltName should be possible for the CSMS

There are scenarios where using the subjectAltName is very useful for an CSMS.

Addition to note	It is allowed to use the subjectAltName extension of type dnsName for a CSMS, when the CSMS has multiple network paths to reach it. (for example, via a private APN + VPN using its IP address in the VPN and via public Internet using a named URL)

3.18. (Medium) - Page 25, section 1.5, Requirement A00.FR.706 is no longer applicable

The signingCertificate has been included at the UpdateFirmwareRequest, so A00.FR.706 is no longer applicable.

Removed requirement:

A00.FR.706		Online verification for the Firmware Signing certificate SHOULD be performed. As part of the firmware updated process, the Charging Station downloads the Firmware Signing certificate whenever it performs a firmware update.

3.19. (Minor) - Page 25, section 1.5, Not clear why OCPP 2.0 uses fast expiration of certificates

Fast expiration has a large impact on the implementation of OCPP 2.0. 2.0 already contains a note, at the bottom of the page, that tries to explain the reason, but that note was not clear enough for all the readers.

Old note	Note 1: With fast expiration, the certificate is only valid for a short period, less than 24 hours. After that the server needs to request a new certificate from the Certificate Authority, which may be the CSO itself (see section Certificate Hierarchy). In this way, if the certificate is compromised, the impact is reduced to only a short period. Also, the Charging Stations do not need to implement revocation lists or online certificate verification. This simplifies the implementation of certificate management at the Charging Station and reduces communication costs at the Charging Station side.
New note	Note 1: With fast expiration, the certificate is only valid for a short period, less than 24 hours. After that the server needs to request a new certificate from the Certificate Authority, which may be the CSO itself (see section Certificate Hierarchy). This prevents the Charging Stations from needing to implement revocation lists or online certificate verification. This simplifies the implementation of certificate management at the Charging Station and reduces communication costs at the Charging Station side. By requiring fast expiration, if the certificate is compromised, the impact is reduced to only a short period. When the certificate chain should becomes compromised, attackers could used forged certificates to trick a Charging Station to connect to a "fake" CSMS. By using fast expiration, the time a Charging Station is vulnerable is greatly reduced. The Charging Station always communicates with the Certificate Authority through the CSMS, this way, if the Charging Stations is compromised, the Charging Station cannot attack the CA directly.

3.20. (Minor) - Page 27, Use case A01, scenario step 1 is incorrect

AuthorizationKey is the old name of the configuration variable. It should be *BasicAuthPassword*.

Old	1. The CSMS sends a SetVariablesRequest(Component=ChargingStation, Variable=AuthorizationKey) to the Charging Station with the BasicAuthPassword Configuration Variable.
New	1. The CSMS sends a SetVariablesRequest(**ComponentName=SecurityCtrlr, VariableName=BasicAuthPassword**) to the Charging Station.

3.21. (Minor) - Page 27, Change to A01.FR.01

The description of A01.FR.01 was contradictory: 'SHALL' and 'as an option' do not go well together.

Version	Id	Precondition	Requirement definition
Old	A01.FR.01		The password SHALL be stored as an option under the `BasicAuthPassword`.
New	A01.FR.01		The password SHALL be stored in the configuration variable `BasicAuthPassword`.

3.22. (Minor) - page 28, use case A01, requirements changed

Version	ID	Precondition	Requirement definition
Old	A01.FR.04	A01.FR.02 AND The Charging Station responds to this SetVariablesRequest with a SetVariablesResponse with status *Rejected* or *UnsupportedVariableType*.	The CSMS SHALL assume that the Charging Station has NOT changed the password. Therefore the CSMS SHALL keep accepting the old credentials.
New	A01.FR.04	A01.FR.02 AND The Charging Station responds to this SetVariablesRequest with a SetVariablesResponse with status **other than *Accepted***	The CSMS SHALL assume that the Charging Station has NOT changed the password. Therefore the CSMS SHALL keep accepting the old credentials.
Old	A01.FR.06		Different user names and passwords SHOULD be used for different Charging Stations.

Version	ID	Precondition	Requirement definition
New	A01.FR.06		Different **passwords** SHOULD be used for different Charging Stations.

3.23. (Minor) - page 28, use case A01, requirement: A01.FR.11: note about password should have been a requirement

A01.FR.11 contains a note indicating that the CP should not disclose the password in its logging. This should be a requirement.

Obsolete note: "The Charging Station should not disclose the BasicAuthPassword in its logging. This is to prevent exposure of key material to persons that may have access to a diagnostics file."

New requirement:

ID	Precondition	Requirement definition	Note
A01.FR.12	A01.FR.11	The Charging Station SHALL NOT disclose the content of the BasicAuthPassword in its logging. This is to prevent exposure of key material to persons that may have access to a diagnostics file.	

3.24. (Minor) - Page 29, Use Case A01, Bad recommendation in requirement

The CSMS should not "keep trying" to update the credentials.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	A01.FR.05	A01.FR.04	While the CSMS SHALL still accepts a connection from the Charging Station, it MAY restrict the functionality that the Charging Station can use. The CSMS can use the BootNotification state: Pending for this. During the Pending state, the CSMS can for example keep trying to update the credentials.
New	A01.FR.05	A01.FR.04	While the CSMS SHALL still accepts a connection from the Charging Station, it MAY restrict the functionality that the Charging Station can use. The CSMS can use the BootNotification state: Pending for this. During the Pending state, the CSMS can for example **retry** to update the credentials.

3.25. (Minor) - Page 31, use case A02, requirement changes

Version	Id	Precondition	Requirement definition
Old	A02.FR.04		The CSMS SHOULD NOT sign the certificate itself, but instead forwards the CSR to a dedicated certificate authority server managing the certificates for the Charging Station infrastructure.
New	A02.FR.04		The CSMS SHOULD NOT sign the certificate itself, but instead forwards the CSR to a dedicated certificate authority server managing the certificates for the Charging Station infrastructure. **The dedicated authority server MAY be operated by the CSO.**

3.26. (Major) - Page 30, ISO 15118 certificate management fixes

This part of the errata sheet contains errata related to the 15118 certificate management.

OCPP does not support handling for more than one V2G root certificate and certificate chain, while ISO 15118 requires that the SECC needs to provide storage for at least 10. This issue can mainly be solved by changing descriptions and requirements, however the GetInstalledCertificateIds message pair is not able to handle certificate chains. So that needs to be fixed on the message/dataType level.

A related issue is that OCPP supports two ways of installing Sub CA certificates. Either the certificates can be included in the certificate chain of Use case A02/A03 or they can be installed using Use case M05. It not a good practice to have two ways of doing the same, therefore the installation of Sub CA certificates using Use case M05 will be dropped.

3.26.1. Page 30 & 262, Use case A02 & M05, Incorrect remark

Changed remark:

Old remark	Even though the messages CertificateSignedRequest (see use cases A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station) and InstallCertificateRequest (use case M05) are both used to send certificates, their purposes are different. CertificateSignedRequest is used to return the the Charging Stations own public certificate signed by a Certificate Authority. InstallCertificateRequest is used to send other Root / SubCA certificates to trust other connections.
New remark	Even though the messages CertificateSignedRequest (see use cases A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station) and InstallCertificateRequest (use case M05) are both used to send certificates, their purposes are different. CertificateSignedRequest is used to return the the Charging Stations own public certificate **and V2G certificate(s)** signed by a Certificate Authority. InstallCertificateRequest is used to install **Root certificates**.

3.26.2. Page 30, Use case A02, Incorrect remark

Changed remark:

Old remark	For (Sub-)CA certificate handling see use cases M03 - Retrieve list of available certificates from a Charging Station, M04 - Delete a specific certificate from a Charging Station, M05 - Install CA certificate in a Charging Station and M06 - Get Charging Station Certificate status.
New remark	For **V2G** certificate handling see use cases M03 - Retrieve list of available certificates from a Charging Station, **M04 - Delete a specific certificate from a Charging Station and M06 - Get Charging Station Certificate status.**

3.26.3. Page 30 & 32, Use case A02 & A03, the requirements for ChargingStationCertificate and V2GCertificate should be separated

Changed requirements:

Version	Id	Precondition	Requirement definition
Old	A02.FR.09	If the Charging Station contains more than one valid certificate of the same type.	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period Keys used in OCPP.
New	A02.FR.09	If the Charging Station contains more than one valid certificate of the **ChargingStationCertificate** type.	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period Keys used in OCPP.
Old	A02.FR.10	When the Charging Station has validated that the new certificate works	The Charging Station MAY discard a certificate. It is RECOMMENDED to store old certificates for one month, as fallback.
New	A02.FR.10	**A02.FR.09 AND** When the Charging Station has validated that the new certificate works	The Charging Station MAY discard **the old** certificate. It is RECOMMENDED to store old certificates for one month, as fallback.
Old	A03.FR.09	If the Charging Station contains more than one valid certificate of the same type.	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period Keys used in OCPP.
New	A03.FR.09	If the Charging Station contains more than one valid certificate of the **ChargingStationCertificate** type.	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period Keys used in OCPP.
Old	A03.FR.10		The Charging Station MAY discard a certificate one month after its validity has expired.
New	A03.FR.10	**A03.FR.09 AND** **When the Charging Station has validated that the new certificate works**	**The Charging Station MAY discard the old certificate. It is RECOMMENDED to store old certificates for one month, as fallback.**

New requirements:

Id	Precondition	Requirement definition
A02.FR.15	If the Charging Station contains more than one valid **V2G certificate, derived from the same root certificate.**	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period Keys used in OCPP.
A03.FR.15	If the Charging Station contains more than one valid **V2G certificate, derived from the same root certificate.**	The Charging Station SHALL use the newest certificate, as measured by the start of the validity period Keys used in OCPP.

3.26.4. Page 260 & 261 & 262 & 263, Use case M03 & M04 & M05 & M06, Incorrect remark

Changed remark:

Old remark	For updating the (V2G) Charging Station Certificate, see use cases A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station.
New remark	For **installing a** (V2G) Charging Station Certificate, see use cases A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station. **The V2G certificate chain SHOULD NOT include the V2GRootCertificate. This SHOULD be installed using Use case M05 - Install CA certificate in a Charging Station.**

3.26.5. Page 261, Use case M04, It should be possible to delete the V2G Charging Station Certificate using Use case M04

It should be possible to delete the certificates from the V2G certificate chain, installed using Use case A02/A03.

Removed requirement:

ID	Precondition	Requirement definition
M04.FR.05		Deletion of the V2G Charging Station Certificate SHALL NOT be possible via a DeleteCertificateRequest.

3.26.6. Page 306, Section 1.22, GetInstalledCertificateIds does not support getting V2G certificate chains.

The CSMS should be able to request request all installed V2G certificate chains and it should also be possible to request different certificate types in one message.

CertificateUseEnumType needs to be split into two separate EnumTypes; GetCertificateIdUseEnumType and InstallCertificateUseEnumType. The values *CSOSubCA1* and *CSOSubCA2* need to be removed from InstallCertificateUseEnumType and GetCertificateIdUseEnumType needs to contain the value *V2GCertificateChain*, instead of the values *CSOSubCA1* and *CSOSubCA2*. The Sub CA certificates can be installed using Use case A02/A03.

InstallCertificateUseEnumType:

Value	Description
V2GRootCertificate	Use for certificate of the V2G Root, a V2G Charging Station Certificate MUST be derived from one of the installed V2GRootCertificate certificates.
MORootCertificate	Use for certificate from an eMobility Service provider. To support PnC charging with contracts from service providers that not derived their certificates from the V2G root.
CSMSRootCertificate	Root certificate for verification of the CSMS certificate.
ManufacturerRootCertificate	Root certificate for verification of the Manufacturer certificate.

GetCertificateIdUseEnumType:

Value	Description
V2GCertificateChain	ISO 15118 V2G certificate chain (excluding the V2GRootCertificate).
V2GRootCertificate	Use for certificate of the V2G Root.

Value	Description
MORootCertificate	Use for certificate from an eMobility Service provider. To support PnC charging with contracts from service providers that not derived their certificates from the V2G root.
CSMSRootCertificate	Root certificate for verification of the CSMS certificate.
ManufacturerRootCertificate	Root certificate for verification of the Manufacturer certificate.

Changed field type:

Page	Section	Message/dataType	Field	Old field type	New field type
306	1.22.1	GetInstalledCertificateIdsRequest	typeOfCertificate	CertificateUseEnumType	GetCertificateIdUseEnumType
310	1.30.1	InstallCertificateRequest	certificateType	CertificateUseEnumType	InstallCertificateUseEnumType

Changed cardinality:

Page	Section	Message/dataType	Field	Old Card.	New Card.	Old Description	New Description
306	1.22.1	GetInstalledCertificateIdsRequest	typeOfCertificate	1..1	0..*	Required. Indicates the type of certificates requested.	Optional. Indicates the type of certificates requested. When omitted, all certificate types are requested.

There could be multiple V2GCertificateChains installed, so the GetInstalledCertificateIdsResponse should be able to return a list of certificate chains.

Page	Section	Message/dataType	Field	Type	Card.	Description	Action
306	1.22.2	GetInstalledCertificateIdsResponse	certificateHashData	CertificateHashDataType	0..*	Optional. The Charging Station includes the Certificate information for each available certificate.	removed
306	1.22.2	GetInstalledCertificateIdsResponse	certificateHashDataChain	CertificateHashDataChainType	0..*	Optional. The Charging Station includes the Certificate information for each available certificate.	added

New DataType 'CertificateHashDataChainType':

Field	Type	Card.	Description
certificateType	GetCertificateIdUseEnumType	1..1	Required. Indicates the type of the requested certificate(s).
certificateHashData	CertificateHashDataType	1..1	Required. Information to identify a certificate.
childCertificateHashData	CertificateHashDataType	0..4	Optional. Information to identify the child certificate(s).

New Schema files are available.

A requirement needs to be added which explains the use of the newly added V2GCertificateChain enum value.

New requirement:

Id	Precondition	Requirement definition
M03.FR.05	When the Charging Station receives a GetInstalledCertificateIdsRequest with typeOfCertificate V2GCertificateChain	The Charging Station SHALL include the hash data for each installed certificate belonging to a V2G certificate chain. Sub CA certificates SHALL be placed as a childCertificate under the V2G Charging Station certificate.

3.27. (Medium) - Page 30, The specification does not specify a CSR format

The specification refers to the X.509 standard, but this standard does not specify a CSR format. However there already is a reference to RFC 2986 in which the ASN.1 notation is described, but it is not referenced in the specification.

Changed requirement 'A02.FR.03 & A03.FR.03':

Old requirement definition	The Charging Station SHALL send the public key in form of a Certificate Signing Request (CSR) as described in the X.509 standard [19] using the SignCertificateRequest message.
New requirement definition	The Charging Station SHALL send the public key in form of a Certificate Signing Request (CSR) as described in **RFC 2986 [22]** using the SignCertificateRequest message.

Changed description:

Page	Section	Message/dataType	Field Name	Old description	New description
323	1.60.1	SignCertificateRequest	csr	Required. The Charging Station SHALL send the public key in form of a Certificate Signing Request (CSR) as described in the X.509 standard [19] using the SignCertificateRequest message.	Required. The Charging Station SHALL send the public key in form of a Certificate Signing Request (CSR) as described in **RFC 2986 [22]** using the SignCertificateRequest message.

3.28. (Minor) - Page 30, A02.FR.14 & A03.FR.14: Charging Station should be CSMS

The requirements A02.FR.14 & A03.FR.14 describe an incorrect actor.

Old text	It is RECOMMENDED for the **Charging Station** to set the typeOfCertificate field in the CertificateSignedRequest to the type of certificate in the SignCertificateRequest.
New text	It is RECOMMENDED for the **CSMS** to set the certificateType field in the CertificateSignedRequest to the type of certificate in the SignCertificateRequest.

3.29. (Minor) - page 30, use case A02, 8 remark: incorrect description about Certificate Authority

The remark about the Certificate Authority signing the CSR and sending it via an OCPP message is incorrect.

Old text	The applicable Certification Authority SHALL check the information in the CSR. If it is correct, the Certificate Authority SHALL sign the CSR, **and send it back in the SignCertificateResponse message**
New text	The applicable Certification Authority SHALL check the information in the CSR. If it is correct, the Certificate Authority SHALL sign the CSR, **send it to the CSO, the CSO sends it back to the Charging Station in the CertificateSignedRequest message.**

3.30. (Minor) - Page 30 & 32, Use Case A02 & A03, Missing prerequisite

Use Case A02 and A03 require the Charging Station to send a CSR, this CSR needs to contain the CSO name, this can be set via the standard configuration variable: `OrganizationName`. This needs to be defined as a prerequisite.

```
The standard configuration variable 'OrganizationName' MUST be set.
```

3.31. Page 30 & 32, Use case A02 & A03, When the CA rejects the CSR the Charging Station will never get to know

Addition to Error handling:

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

3.32. (Minor) - Page 31, Use Case A02 & A03, the validity of a certificate is determined by more than only the 'Not valid before' field in the certificate

Changed requirement A02.FR.08 & A03.FR.08:

Old requirement definition	The Charging Station SHALL switch to the new certificate as soon as **it is valid. A certificate is deemed valid when** the current date and time is after the 'Not valid before' field in the certificate.
New requirement definition	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.

3.33. (Minor) - page 32, requirement A03.FR.01: forcing the Charging Station to start the update overlaps with A02.FR.01

There are 2 requirements: A02.FR.01 and A03.FR.01, both are the same, they require the CSMS and the Charging Station to start the update of the Charging Station certificate. This is double. It is not needed to have both required to start this process.

Old text	A03.FR.01: A key update **SHOULD** be performed after installation of the Charging Station, to change the key from the one initially provisioned by the manufacturer (possibly a default key).
New text	A03.FR.01: A key update **MAY** be performed after installation of the Charging Station, to change the key from the one initially provisioned by the manufacturer (possibly a default key).

3.34. (Minor) - page 32, use case A04: missing requirement to log security events.

The was a requirement in use case N01 about logging security events, but that was not the correct place for this requirement. This requirement should have been part of A04.

New requirement:

ID	Precondition	Requirement definition	Note
A04.FR.06	When a security event happens (also non-critical)	The Charging Station SHALL store the security event in a security log.	It is recommended to implement this log in a rolling format.

3.35. (Medium) - Page 33, Changing Security Profile

Upgrading the Security Profile to a more secure one should be possible, so a Use case needs to be added for this. But downgrading the Security Profile to a less secure one should not be possible.

3.35.1. New Use Case A05 - Upgrade Charging Station Security Profile

The CPO wants to increase the security of the OCPP connection between CSMS and a Charging Station.
This use case is especially relevant when migrating from OCPP 1.6 without security profiles to OCPP 1.6 with security profiles or OCPP 2.0, before migrating to a security profile the prerequisites, like installed certificates or password need to be configured. The CPO ensures the prerequisite(s) for going to a higher security certificates are met before sending the command to change to a higher security profile.

NOTE For security reasons it is not allowed to change to a lower Security Profile over OCPP.

New requirements:

ID	Precondition	Requirement definition
B09.FR.04	On receipt of the SetNetworkProfileRequest AND the NetworkConnectionProfile contains a lower securityProfile than stored at the configuration variable `SecurityProfile`	The Charging Station SHALL respond by sending a SetNetworkProfileResponse message, with status *Rejected*
A05.FR.02	The Charging Station receives SetVariablesRequest for `NetworkConfigurationPriority` containing a profile slot for a NetworkConnectionProfile with a 'securityProfile' value higher than the current value AND new value is 2 or 3 AND No valid CSMSRootCertificate installed	The Charging Station SHALL respond with SetVariablesResponse(Rejected), and not update the value for `SecurityProfile` and/or reconnect to the CSMS.
A05.FR.03	The Charging Station receives SetVariablesRequest for `NetworkConfigurationPriority` containing a profile slot for a NetworkConnectionProfile with a 'securityProfile' value higher than the current value AND new value is 3 AND No valid ChargingStationCertificate installed	The Charging Station SHALL respond with SetVariablesResponse(Rejected), and not update the value for `SecurityProfile` and/or reconnect to the CSMS.
A05.FR.04	The Charging Station receives SetVariablesRequest for `NetworkConfigurationPriority` containing profile slots for NetworkConnectionProfiles with a 'securityProfile' value equal to or higher than the current value AND all prerequisites are met	The Charging Station SHALL respond with SetVariablesResponse(Accepted)
A05.FR.05	A05.FR.04 AND After a reboot	The Charging Station SHALL begin connecting to the first entry of `NetworkConfigurationPriority`
A05.FR.06	A05.FR.05 AND The Charging Station successfully connected to the CSMS using the (new) NetworkConnectionProfile	The Charging Station SHALL update the value of the configuration variable `SecurityProfile` AND it SHALL remove all NetworkConnectionProfiles with a lower securityProfile than stored at `SecurityProfile` AND update `NetworkConfigurationPriority` accordingly.
A05.FR.07	A05.FR.06	The CSMS SHALL NOT allow the Charging Station to connect with a lower security profile anymore.

New configuration variable 'SecurityProfile':

Required	yes		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	SecurityProfile	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	This configuration variable is used to report the security profile used by the Charging Station.		

Field added to DataType 'NetworkConnectionProfileType':

Field Name	Field Type	Card.	Description
securityProfile	integer	1..1	Required. This field specifies the security profile used when connecting to the CSMS with this NetworkConnectionProfile.

3.35.2. page 18, requirement A00.FR.005 needs to be updated

Requirement A00.FR.005 needs to be updated, because with the above errata it will become possible to change the security profile using OCPP.

Old text	Changing the security profile that is used is for security reasons not part of the OCPP specification, and must be done through another method, not via OCPP. It is NOT allowed to make this configurable via SetVariablesRequest.
New text	Lowering the security profile that is used to a less secure profile is, for security reasons, not part of the OCPP specification, and MUST be done through another method, not via OCPP. OCPP messages SHALL NOT be used for this (e.g. SetVariablesRequest or DataTransferRequest).

3.36. (Major) - Page 35, Use case B01, Error in scenario description

Step 5 in the scenario description was not correct. If the powering-up is caused by a power outage while a transaction was ongoing, then Charging Station may resume this transaction (if so configured).

Old	**5.** The Charging Station sends StatusNotificationRequest to the CSMS for each Connector. If the status was set to *Unavailable* or *Reserved* from the CSMS prior to the (re)boot, the Connector should return to this status, otherwise the status should be *Available*.
New	**5.** The Charging Station sends StatusNotificationRequest to the CSMS for each Connector. If the status was set to *Unavailable* or *Reserved* from the CSMS prior to the (re)boot, the Connector should return to this status, otherwise the status should be *Available* **or, when it resumes a transaction that was ongoing, the status should be *Occupied*.**

3.37. (Minor) - Page 36, Use case B01-03, Message received before Charging Station has been accepted

A requirement is missing that specifies what happens when a Charging Station sends a message before it has received a BootNotificationResponse with status *Accepted*. This needs to be added to use cases B01, B02 and B03.

Add the following requirement to table 34.

ID	Precondition	Requirement definition	Note
B01.FR.10	The Charging Station has received a BootNotificationResponse in which **status** is not *Accepted* AND the Charging Station sends a RPC Framework: CALL message that is NOT a BootNotificationRequest or a message triggered by one of the following messages: TriggerMessageRequest, GetBaseReportRequest, GetReportRequest.	The CSMS SHALL respond with RPC Framework: CALLERROR: SecurityError.	The Charging Station is not allowed to initiate sending other messages before being accepted.

Add the following requirement to table 36.

ID	Precondition	Requirement definition	Note
B02.FR.09	The Charging Station has received a BootNotificationResponse with status *Pending* AND the Charging Station sends a RPC Framework: CALL message that is NOT a BootNotificationRequest or a message triggered by one of the following messages: TriggerMessageRequest, GetBaseReportRequest, GetReportRequest.	The CSMS SHALL respond with RPC Framework: CALLERROR: SecurityError.	The Charging Station is not allowed to initiate sending other messages before being accepted.

Add the following requirement to table 38.

ID	Precondition	Requirement definition
B03.FR.07	B03.FR.03 AND Charging Station sends a message that is not a BootNotificationRequest	CSMS SHALL respond with RPC Framework: CALLERROR: SecurityError.
B03.FR.08	B03.FR.03 AND CSMS sends a message that is not a TriggerMessageRequest(requestedMessage = BootNotification)	Charging Station SHALL respond with RPC Framework: CALLERROR: SecurityError.

3.38. (Medium) - Page 37, Use case B01, Missing requirements to validate the Serial Number in the BootNotificationRequest

New requirements:

ID	Precondition	Requirement definition
B01.FR.11	B01.FR.01 AND Security profile 3 is used	The CSMS SHALL check the SerialNumber in the BootNotificationRequest against the Serial Number in the Certificate Common Name.
B01.FR.12	B01.FR.11 AND the SerialNumber in the BootNotificationRequest does NOT equal the Serial Number in the Certificate Common Name	The CSMS SHALL close WebSocket connection.

3.39. (Medium) - Page 38, Added requirement that *Reserved* state is persistent

New requirement:

ID	Precondition	Requirement definition
B01.FR.13	When an EVSE has been reserved	The *Reserved* state MUST be persistent across reboots.

3.40. (Minor) - Page 39, Use B02, Step #6 of scenario removed

The following step is moved from scenario description of the use case to the remarks, because it is not part of the scenario:

6. When the CSMS returns with BootNotificationResponse with the status *Accepted*, B01 - Cold Boot Charging Station applies.

3.41. (Minor) - Page 40, Use Case B02, Incorrect requirements

The requirement B02.FR.02 states that the Charging Station is not allowed to send a request message to the CSMS. But it needs to be allowed to respond to for example a GetBaseReportRequest, which triggers NotifyReportRequest.

Requirement B02.FR.08 suggests that a new BootNotificationRequest can be sent immediately, which was wrong.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	B02.FR.02	While the CSMS has not yet responded to a BootNotificationRequest with an Accepted status in the BootNotificationResponse.	The Charging Station SHALL NOT send messages (Except BootNotificationRequest) to the CSMS, unless it has been instructed by the CSMS to do so with TriggerMessageRequest.
New	B02.FR.02	While the CSMS has not yet responded to a BootNotificationRequest with an Accepted status in the BootNotificationResponse.	The Charging Station SHALL NOT send messages (Except BootNotificationRequest) to the CSMS, unless it has been instructed by the CSMS to do so, **using one of the following messages; TriggerMessageRequest, GetBaseReportRequest, GetReportRequest.**
Old	B02.FR.07	If the interval in the BootNotificationRequest equals 0, and the status is other than *Accepted*,	The Charging Station SHALL choose a waiting interval on its own, in a way that avoids flooding the CSMS with requests.
New	B02.FR.07	If the interval in the **BootNotificationResponse** equals 0, and the status is other than *Accepted*,	The Charging Station SHALL choose a waiting interval on its own, in a way that avoids flooding the CSMS with requests.
Old	B02.FR.08	If the interval in the BootNotificationRequest > 0, and the status is other than *Accepted*,	The Charging Station SHALL send a BootNotificationRequest.
New	B02.FR.08	If the interval in the **BootNotificationResponse** > 0, and the status is other than *Accepted*,	The Charging Station SHALL send a BootNotificationRequest **after the set interval has past.**

3.42. (Minor) - Page 41, Use B03, Step #5 of scenario removed

The following step is moved from scenario description of the use case to the remarks, because it is not part of the scenario:

5. The CSMS returns with BootNotificationResponse with the status *Accepted*, B01 - Cold Boot Charging Station applies.

3.43. (Minor) - Page 42, Use case B03, Requirements changes

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	B03.FR.04	B03.FR.03	The Charging Station MAY disconnect the WebSocket until it needs to send the next BootNotificationRequest.
New	B03.FR.04	B03.FR.03	The Charging Station MAY **close the connection** until it needs to send the next BootNotificationRequest.

3.44. (Minor) - Page 44, Use cases B, Section 2.2, Configuring a Charging Station

To the start of section 2.2, before use case B05 - Set Variables, add the following note:

NOTE	For managing the configuration of a Charging Station a basic understanding of Device Model concepts is essential. These concepts are explained in "OCPP 2.0: Part 1 - Architecture & Topology", chapter 4.

3.45. (Medium) - Page 45, Use case B05, requirements changes

Version	Id	Precondition	Requirement definition
Old	B05.FR.07	When the Charging Station receives a SetVariablesRequest with a *value* that cannot be set on the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: InvalidValue.
New	B05.FR.07	When the Charging Station receives a SetVariablesRequest with a *value* that **is incorrectly formatted for** the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: **Rejected. (More information can be provided in the optional *statusInfo* element.)**
Old	B05.FR.08	When the Charging Station receives a SetVariablesRequest with a *value* that is lower or higher than the range of the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: OutOfRange.
New	B05.FR.08	When the Charging Station receives a SetVariablesRequest with a *value* that is lower or higher than the range of the given Variable in the SetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: **Rejected. (More information can be provided in the optional *statusInfo* element.)**
Old	B05.FR.09	When the Charging Station receives a SetVariablesRequest for a Variable in the SetVariableData that is *ReadOnly*	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: Rejected.
New	B05.FR.09	**NOT (B05.FR.04 to B05.FR.08) AND** When the Charging Station receives a SetVariablesRequest for a Variable in the SetVariableData, but is not able to set it	The Charging Station SHALL set the *attributeStatus* field in the corresponding SetVariableResult to: Rejected. **(This happens if the variable is *ReadOnly*, but may also occur when setting the variable fails because of technical problems.)**

3.46. (Medium) - page 48/49, Use case B07, requirements changes

Removed requirements:

ID	Precondition	Requirement definition
B07.FR.06	B07.FR.01	The Charging Station SHALL support all *reportBase* reports.

Changed requirements:

Version	Id	Precondition	Requirement definition	Note
Old	B07.FR.01		The Charging Station SHALL support all required reportBase reports.	The Charging Station is allowed to reject a GetBaseReportRequest for a supported reportBase if it is temporarily unable to execute the request.
New	B07.FR.01	**When the Charging Station receives a getBaseReportRequest for a supported *reportBase* AND NOT B07.FR.13**	**The Charging Station SHALL send a getBaseReportResponse with `Accepted`.**	
Old	B07.FR.03		The Charging Station SHALL send the requested information via one or more NotifyReportRequest messages to the CSMS.	
New	B07.FR.03	**B07.FR.01**	The Charging Station SHALL send the requested information via one or more NotifyReportRequest messages to the CSMS.	
Old	B07.FR.08	B07.FR.01 AND When *reportBase* is FullInventory	Then the Charging Station SHALL respond with a NotifyReportRequest to report on all component-variables including their *VariableCharacteristics*.	

Version	Id	Precondition	Requirement definition	Note
New	B07.FR.08	B07.FR.01 AND When *reportBase* is FullInventory	Then the Charging Station SHALL respond with a NotifyReportRequest to report on all component-variables including their *VariableCharacteristics*.	**As a minimum the required variables mentioned in Section 2.13 Charging Infrastructure related shall be reported as well as the required variables in Section 1 Controller Components that are relevant to each functional block that has been implemented.**

New requirements:

Id	Precondition	Requirement definition	Note
B07.FR.11	B07.FR.08	All attribute types of a variable, that are supported by the Charging Station, SHALL be reported, even if they have no value (are unset).	This allows a CSMS to know which attribute types are supported by the Charging Station.
B07.FR.12		The Charging Station SHALL support at least the base reports: ConfigurationInventory and FullInventory.	
B07.FR.13	When the Charging Station is temporarily unable to execute a report request	The Charging Station SHALL send a getBaseReportResponse with `Rejected`.	

Changed descriptions 'ReportBaseEnumType' values:

Enum value	Old description	New description
FullInventory	A (configuration) report that lists all Components/Variables that can be set by the operator.	**Required.** A (full) report that lists everything except monitoring settings.
ConfigurationInventory	A (full) report that lists everything except monitoring settings.	**Required.** A (configuration) report that lists all Components/Variables that can be set by the operator.
SummaryInventory	A (summary) report that lists Components/Variables relating to the Charging Station's current charging availability, and to any existing problem conditions. For the Charging Station Component: - AvailabilityState. For each EVSE Component: - AvailabilityState. For each Connector Component: - AvailabilityState (if known and different from EVSE). For all Components in an abnormal State: - Active (Problem, Tripped, Overload, Fallback) variables. - Any other diagnostically relevant Variables of the Components. - Include TechCode and TechInfo where available. All monitored Component.Variables in Critical or Alert state shall also be included. - Charging Stations that do not have Monitoring implemented are NOT REQUIRED to include Connector Availability, monitoring alerts, and MAY limit problem reporting detail to just the active Problem boolean Variable.	**Optional.** A (summary) report that lists Components/Variables relating to the Charging Station's current charging availability, and to any existing problem conditions. For the Charging Station Component: - AvailabilityState. For each EVSE Component: - AvailabilityState. For each Connector Component: - AvailabilityState (if known and different from EVSE). For all Components in an abnormal State: - Active (Problem, Tripped, Overload, Fallback) variables. - Any other diagnostically relevant Variables of the Components. - Include TechCode and TechInfo where available. All monitored Component.Variables in Critical or Alert state shall also be included. - Charging Stations that do not have Monitoring implemented are NOT REQUIRED to include Connector Availability, monitoring alerts, and MAY limit problem reporting detail to just the active Problem boolean Variable.

3.47. (Minor) - Page 49, Use case B07, Incorrect requirement note

Changed requirement note B07.FR.09:

Old note	A (summary) report that lists Components/Variables relating to the Charging Station's current charging availability, and to any existing problem conditions. For the Charging Station Component: - AvailabilityState. For each EVSE Component: - AvailabilityState. For each Connector Component: - AvailabilityState (if known and different from EVSE). For all Components in an abnormal State: - Active (Problem, Tripped, Overload, Fallback) variables. - Any other diagnostically relevant Variables of the Components. - Include TechCode and TechInfo where available. All monitored Component.Variables in Critical or Alert state shall also be included. - Charging Stations that do not have Monitoring implemented are NOT REQUIRED to include Connector Availability, monitoring alerts, and MAY limit problem reporting detail to just the active Problem boolean Variable.
New note	A (summary) report that lists Components/Variables relating to the Charging Station's current charging availability, and to any existing problem conditions. For the Charging Station Component: - AvailabilityState. For each EVSE Component: - AvailabilityState. For each Connector Component: - AvailabilityState (if known and different from EVSE). For all Components in an abnormal State: - Active (Problem, Tripped, Overload, Fallback) variables. - Any other diagnostically relevant Variables of the Components.

3.48. (Minor) - page 49, Use Case B06, Incorrect/unclear requirements

Changed requirement:

Version	ID	Precondition	Requirement definition
Old	B06.FR.13	B06.FR.12 AND When the Charging Station has no attributeValue for a requested attributeType of a componentvariable	Then the Charging Station SHALL return an empty string as attributeValue. Example: For a temperature sensor a Charging Station might only be able to report an Actual value. For GetVariablesRequest for attributeType other than Actual it will respond with an empty string as attributeValue.
New	B06.FR.13	NOT B06.FR.08 AND the Charging Station has no *attributeValue* for the requested *attributeType* of the componentvariable	Charging Station SHALL return an empty string as *attributeValue*. Note: this can happen, for example, when the *attributeType* `Target` has not yet been set, even though it is supported.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	B06.FR.06	When the Charging Station receives a GetVariablesRequest with an unknown Component in the GetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding GetVariableResult to: UnknownComponent.
New	B06.FR.06	When the Charging Station receives a GetVariablesRequest with an unknown Component in the GetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding GetVariableResult to: UnknownComponent **AND SHALL omit the *attributeValue***.

Version	Id	Precondition	Requirement definition
Old	B06.FR.07	When the Charging Station receives a GetVariablesRequest with a Variable that is unknown for the given Component in the GetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding GetVariableResult to: UnknownVariable.
New	B06.FR.07	When the Charging Station receives a GetVariablesRequest with a Variable that is unknown for the given Component in the GetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding GetVariableResult to: UnknownVariable **AND SHALL omit the *attributeValue***.
Old	B06.FR.08	When the Charging Station receives a GetVariablesRequest with an *attributeType* that is unknown for the given Variable in the GetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding GetVariableResult to: NotSupportedAttributeType.
New	B06.FR.08	When the Charging Station receives a GetVariablesRequest with an *attributeType* that is unknown for the given Variable in the GetVariableData	The Charging Station SHALL set the *attributeStatus* field in the corresponding GetVariableResult to: NotSupportedAttributeType. **AND SHALL omit the *attributeValue***.

Removed requirement:

Id	Precondition	Requirement definition
B06.FR.12	B06.FR.01	Any attributeType (Actual, Target, MinSet, MaxSet) shall be supported in GetVariablesRequest for any component-variable.

3.48.1. Page 353, Section 1.26. GetVariableResultType, description must be inline with above requirement changes

Changed description:

Old description	Optional. Value of requested attribute type of componentvariable. This field can only be empty when the given status is NOT accepted. The Configuration Variable ValueSize can be used to limit the VariableCharacteristicsType.ValueList and all AttributeValue fields. The max size of these values will always remain equal. The default max size is set to 1000.
New description	Optional. Value of requested attribute type of componentvariable. This field can only be **omitted** when the given status is NOT accepted. The Configuration Variable ValueSize can be used to limit the VariableCharacteristicsType.ValueList and all AttributeValue fields. The max size of these values will always remain equal. The default max size is set to 1000.

3.49. (Minor) - Page 51, B08 Get Custom Report, Unclear how to filter components using the *componentCriteria*

The use case does not specify in what way the Charging Station should filter the components based on the *componentCriteria*.

New requirement:

ID	Precondition	Requirement definition
B08.FR.13	B08.FR.01 AND More than one *componentCriteria* is given.	The Charging Station SHALL report all components that have at least one of the given criteria (logical OR).

3.50. (Medium) - Page 51, B08 Get Custom Report, Allow rejection of report

New requirement:

ID	Precondition	Requirement definition
B08.FR.16	When the Charging Station is temporarily unable to execute a report request	The Charging Station SHALL send a getBaseReportResponse with `Rejected`.

3.51. (Minor) - Page 53, Use case B10, Improved requirement definition

Changed requirement:

Version	ID	Precondition	Requirement definition
Old	B10.FR.05		It is RECOMMENDED to make sure the transaction message queue in the Charging Station is empty before sending the ResetRequest. Otherwise the Charging Station might send transaction related message to the new CSMS that has not received the start of the Transaction, and the old system will miss the ended messages. To determine if there are still transaction for an ongoing transaction in the queue, the getTransactionStatusRequest message can be used.
New	B10.FR.05		It is RECOMMENDED to **set the Charging Station to Inoperative (via ChangeAvailabilityRequest) to ensure that no new transactions can be started and wait until** the transaction message queue in the Charging Station is empty before sending the ResetRequest. Otherwise the Charging Station might send transaction related messages to the new CSMS that has not received the start of the Transaction, and the old system will miss the ended messages. To determine if there are still transaction for an ongoing transaction in the queue, the getTransactionStatusRequest message can be used.

3.52. (Minor) - Page 53, Use case B10, Missing requirement about having only one open connection when migrating to another CSMS

New requirements:

ID	Precondition	Requirement definition
B10.FR.06		The Charging Station SHALL disconnect from the old CSMS, before trying to connect to the new CSMS.

3.53. (Major) - Page 54, 2.3 ResetRequest for Charging Station or EVSE

The ResetRequest has an optional parameter *evseId*, which is used to request a reset for only that EVSE ID.

Added configuration variable 'AllowReset':

Required	no		
Component	**componentName**	EVSE	
	evse	*	
Variable	**variableName**	AllowReset	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	Component can be reset. Can be used to announce that an EVSE can be reset individually.		

In Use Case B11 for the rows below, change the text in the **Description** column, as shown in **bold face**:

Table 54 B11 - Reset - Without Ongoing Transaction

No.	Type	Description
3	**Objective(s)**	To enable the CSMS to request a Charging Station **or EVSE** to reset itself, while there is no ongoing transaction.
4	**Description**	This use case covers how the CSMS can request the Charging Station to reset itself **or an EVSE** by sending ResetRequest. **(If ResetRequest contains an optional paramater *evseId*, then only a reset of the specific EVSE is requested.)** This could for example be necessary if the Charging Station is not functioning correctly.
	Scenario description	**1.** The CSO requests the CSMS to reset the Charging Station **or EVSE**. **2.** The CSMS sends ResetRequest requesting the Charging Station to reset itself **or EVSE**. **3.** The CSMS requests for an OnIdle or Immediate reset. **4.** The Charging Station responds with ResetResponse, indicating whether the Charging Station is able to reset itself **or EVSE**. **5.** The CSMS sends an optional notification to the CSO. **6. Only if no *evseId* was supplied, then** after the reset, the Charging Station will proceed as in use case B01.
6	**Postcondition(s)**	**Successful postcondition:** The Charging Station was able to reset itself **or EVSE**. **Failure postcondition:** The Charging Station *not* was able to reset itself **or EVSE**.

The following requirements of B11 need to be updated, as shown:
(unchanged requirements are omitted from the table)

ID	Precondition	Requirement definition
B11.FR.03	B11.FR.01 **AND no *evseId* parameter is supplied** AND ResetResponse was *Accepted*.	The Charging Station SHALL start a reboot.
B11.FR.05	If the status of an EVSE was *Reserved*.	After a reboot of the Charging Station **or EVSE**, the EVSE**(s)** SHALL return to the state *Reserved*.
B11.FR.08 **(NEW)**	B11.FR.01 AND an *evseId* parameter is supplied AND ResetResponse was *Accepted*.	The Charging Station SHALL start a reboot of EVSE that is referred to by *evseId* parameter.
B11.FR.09 **(NEW)**	B11.FR.01 AND an *evseId* parameter is supplied AND Charging Station does not support resetting an individual EVSE	The Charging Station SHALL return a ResetResponse *Rejected*
B11.FR.10 **(NEW)**	When the Charging Station supports resetting of an individual EVSE	The Charging Station SHOULD set the device model variable *AllowReset* to true for the EVSE.

In Use Case B12 for the rows below, change the text in the **Description** column, as shown in **bold face**:

Table 55 B12 - Reset - With Ongoing Transaction

No.	Type	Description
3	**Objective(s)**	To enable the CSMS to request a Charging Station **or EVSE** to reset itself, while there is an ongoing transaction.
4	**Description**	This use case covers how the CSMS can request the Charging Station to reset itself **or an EVSE** by sending ResetRequest. **(If ResetRequest contains an optional paramater *evseId*, then only a reset of the specific EVSE is requested.)** This could for example be necessary if the Charging Station is not functioning correctly. The CSMS has the possibility to let the Charging Station end all transactions itself and reboot or wait until all ongoing transactions are ended normally (by an EV user) and then reboot.

No.	Type	Description
	Scenario description	**1.** The CSO requests the CSMS to reset the Charging Station **or EVSE**. **2.** The CSMS sends ResetRequest requesting the Charging Station to reset itself **or EVSE**. **3a.** On receipt of an OnIdle reset, the Charging Station responds with ResetResponse(Scheduled), indicating the Charging Station will try to reset itself **or EVSE** after all ongoing transactions have ended. The Charging Station continues charging and sets all EVSEs **(or only the one provided in the request, if *evseId* was supplied)** that are Available to status *Unavailable*, waits until all transactions are finished and all TransactionEventRequest (eventType = Ended) messages are sent. **3b.** On receipt of an Immediate reset, the Charging Station responds with ResetResponse(Accepted), indicating the Charging Station **or EVSE** will try to reset itself. The Charging Station attempts to terminate any transaction **(or only those running on the EVSE provided in the request, if *evseId* was supplied)** in progress, and sending a TransactionEventRequest (eventType = Ended) message. **4. Only if no *evseId* was supplied** the Charging Station reboots and returns to a state as just having been booted, B01 - Cold Boot Charging Station applies.
6	**Postcondition(s)**	**Successful postcondition:** The Charging Station was able to reset itself **or EVSE**. **Failure postcondition:** The Charging Station *not* was able to reset itself **or EVSE**.

The following requirements of B12 need to be updated, as shown:
(unchanged requirements are omitted from the table)

ID	Precondition	Requirement definition
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.
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**.
B12.FR.03	**If no *evseId*** is supplied AND If any transaction is in progress and an OnIdle reset is received.	The transaction of the Charging Station SHALL be terminated normally, before the reboot, as in E06 - Stop Transaction.
B12.FR.04	**If no *evseId*** is supplied AND If any transaction is in progress and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate any transaction in progress and send a TransactionEventRequest (eventType = Ended) message before performing a reboot.
B12.FR.05	If an Immediate Reset is received and the TransactionEventResponse is not received within timeout.	The Charging Station SHALL queue the TransactionEventRequest, reboot and resend the TransactionEventRequest after the reboot.
B12.FR.06	If the status was set to *Inoperative* by the CSMS.	After a reboot of the Charging Station **or EVSE**, the EVSE**(s)** SHALL return to the state *Unavailable* as prior to the reboot.
B12.FR.07 **(NEW)**	If an *evseId* is supplied AND If a transaction is in progress on the EVSE and an OnIdle reset is received.	The transaction on the EVSE SHALL be terminated normally, before the reboot, as in E06 - Stop Transaction.
B12.FR.08 **(NEW)**	If an *evseId* is supplied AND If a transaction is in progress on the EVSE and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate the transaction in progress on the EVSE and send a TransactionEventRequest (eventType = Ended) message before performing a reboot.
B12.FR.09 **(NEW)**	B12.FR.01 AND an *evseId* parameter is supplied AND Charging Station does not support resetting an individual EVSE	The Charging Station SHALL return a ResetResponse *Rejected*

3.54. (Major) - Page 60, Enhancements AuthorizationRequest

Since the AuthorizeResponse can return a list of *evseId* at which the *idToken* is allowed to charge (assuming it is `Accepted`), there is no need anymore for the optional parameter *evseId* in the AuthorizeRequest.

3.54.1. evseId should have been placed in *IdTokenInfoType*, not directly in *AuthorizeResponse*, to be able to use it for cache and local list

We added evseId to AuthorizeResponse as a way for the CSMS to tell the CS that the IdToken is only valid for one or more specific EVSE's (i.e. not valid on all EVSE's in the Charging Station). As it would be valuable to be able to specify this also in the local list, as well as to let the Charging Station cache this info, the evseId should have been added to the IdTokenInfoType.

Page	Section	Message/dataType	Field	Action
297	1.1.2	AuthorizeResponse	evseId	removed
337	1.28	IdTokenInfoType	evseId	added

New Schema files are available.

3.54.2. Examples of *idTokenInfo* in various cases

The following example needs to added to the remarks:

Assuming *idToken* is valid for charging and the Charging Station has 3 EVSEs, what is the content of *idTokenInfo*, when *idToken* is allowed to charge

1. at all EVES: *idTokenInfo.status* = `Accepted`.
2. at EVSE 1: *idTokenInfo.status* = `Accepted`, *idTokenInfo.evseId* = [1] .
3. at EVSE 1 + 2: *idTokenInfo.status* = `Accepted`, *idTokenInfo.evseId* = [1, 2].
4. at none of the EVSEs: *idTokenInfo.status* =`NotAtThisLocation`.

3.54.3. Page 64, C01 - EV Driver Authorization using RFID - Requirements

Update the requirements of C01 as shown in the table below:

Table 1. C01 - Requirements

	ID	Precondition	Requirement definition	Note
Changed	C01.FR.01	**Configuration setting AuthEnabled is true.**	The Charging Station SHALL only offer energy after authorization.	
Deleted	C01.FR.14			
Deleted	C01.FR.15			
Deleted	C01.FR.16			
New	C01.FR.18	If the IdToken is valid AND the EV driver is NOT allowed to charge at the type of EVSE(s) this Charging Station provides.	The CSMS SHALL send an AuthorizeResponse with idTokenInfo.status *NotAllowedTypeEVSE*.	
New	C01.FR.19	*idToken* is allowed for any EVSE of the Charging Station	The CSMS SHALL send an AuthorizeResponse in which *idTokenInfo* has an empty (or absent) *evseId* list.	This will be the most common case. Even though the *idToken* might be allowed on any EVSE, the *idTokenInfo.status* still needs to be `Accepted` before charging is allowed.

	ID	Precondition	Requirement definition	Note
New	C01.FR.20	*idToken* is allowed for a subset of EVSEs of the Charging Station	The CSMS SHALL send an AuthorizeResponse in which *IdTokenInfo* has an *evseId* list with the allowed EVSEs.	Note the difference between validity of an *idToken* and the fact whether this (type of) token is allowed on an EVSE. The *idTokenInfo.status* still needs to be `Accepted` before charging is allowed.
New	C01.FR.21	C01.FR.20	The Charging Station SHALL only allow charging on the EVSEs mentioned in the AuthorizeResponse.	
New	C01.FR.22	*idToken* is not allowed for any EVSE of the Charging Station	The CSMS SHALL send an AuthorizeResponse in which *idTokenInfo.status* is `NotAtThisLocation` and *evseId* list is empty (or absent).	Status `NotAtThisLocation` needed in order to differentiate with the situation in which *idToken* is allowed on all EVSEs.

3.55. (Minor) Page 61, Change to 'NOTE'

The restriction that identifiers in local authorization list cannot be added to authorization cache has been removed.

This NOTE:

NOTE

The Authorization Cache and Local Authorization List are **distinct** logical data structures. Identifiers known in the Local Authorization List SHALL NOT be added to the Authorization Cache. When both Authorization Cache as well as Local Authorization List are supported, a Charging Station SHALL treat Local Authorization List entries as having priority over Authorization Cache entries for the same identifiers.

has been replaced by the following:

NOTE

The Authorization Cache and Local Authorization List are **distinct** logical data structures. When both Authorization Cache as well as Local Authorization List are supported, a Charging Station SHALL treat Local Authorization List entries as having priority over Authorization Cache entries for the same identifiers.

3.56. (Minor) Page 64, Use case C03, Improved definition of C01.FR.03

Version	ID	Precondition	Requirement definition	Note
Old	C01.FR.03	When stopping a transaction.	The Charging Station SHALL NOT send an AuthorizeRequest when the IdToken used for stopping the transaction is the same as the IdToken that started the transaction OR when the GroupIdToken used for stopping the transaction and the GroupIdToken that started the transaction are present in either the Local Authorization List or the Authorization Cache and are the same.	
New	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.**	

3.57. (Major) Page 76, Use case C07

3.57.1. Page 76, Use case C07: Changes to use case

Version	Row	Description
Old	*Actors*	Primary actors: EV, Charging Station, HMI Secondary actors: CSMS
New	*Actors*	Actors: EV, Charging Station, CSMS, OCSP
Old	**Remark(s)**	In edition 1 of 15118, the message timeout of the PaymentDetailsReq/Res message is 5 seconds.
New	**Remark(s)**	In edition 1 of 15118, the message timeout of the PaymentDetailsReq/Res message is 5 seconds. In case certificate verification cannot be completed in that time it is possible to complete this during the AuthorizationReq/Res, which can be extended up to 60 seconds.

3.57.2. Page 76, Use case C07: extended diagram with certificate check

Figure 27 was extended with an CA actor that is involved in checking the certificate status.

3.57.3. Page 77, Use case C07: changes to requirements

Removed requirements

C07.FR.03 was not needed and causing confusion.

ID	Precondition	Requirement definition
C07.FR.03		AuthorizeRequest SHOULD only be used for the authorization of an identifier for charging.

New requirement

An explicit requirement to check certificate status with a CA was missing.

ID	Precondition	Requirement definition
C07.FR.05	C07.FR.02	The CSMS SHALL verify validity of the certificate and certificate chain via real-time or cached OCSP data.

3.58. (Medium) - Page 80, Use C09, Requirements changes:

Removed requirements:

ID	Precondition	Requirement definition
C09.FR.01		Message data elements of the IdToken class (including GroupId) SHALL contain any data, subject to the constraints of the data-type.

Changed requirements:

Version	ID	Precondition	Requirement definition
Old	C09.FR.02		IdTokens SHALL be grouped for authorization purposes by specifying a common group identifier in the optional GroupId element in IdTokenInfo
New	C09.FR.02		IdTokens **that are part of the same group for authorization purposes SHALL have** a common group identifier in the optional *groupIdToken* element in IdTokenInfo
Old	C09.FR.03	When a transaction has been authorized/started with a certain IdToken.	An EV Driver with a different IdToken, but with the same groupIdToken SHALL be authorized to stop the transaction.

Version	ID	Precondition	Requirement definition
New	C09.FR.03	When a transaction has been authorized/started with a certain IdToken.	An EV Driver with a different, **valid** IdToken, but with the same groupIdToken SHALL be authorized to stop the transaction.
Old	C09.FR.07	When stopping a transaction.	The Charging Station SHALL NOT send an AuthorizeRequest when the IdToken used for stopping the transaction is the same as the IdToken that started the transaction OR when the GroupIdToken used for stopping the transaction and the GroupIdToken that started the transaction are both known by the Charging Station and the same.
New	C09.FR.07	**C09.FR.03**	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 requirements:

ID	Precondition	Requirement definition
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.

3.59. (Medium) - Page 83, Use case C10, requirement changes

The local authorization list and authorization cache have been decoupled from each other and some requirements have been relaxed to allow for more freedom in implementation.

Changed requirements:

Version	ID	Precondition	Requirement definition	Note
Old	C10.FR.04	Upon receipt of AuthorizeResponse.	The Charging Station SHALL update the cache entry, if the IdToken is not in the Local Authorization List.	The update is to be done with the IdTokenInfo value from the response as described under Authorization Cache.
New	C10.FR.04	Upon receipt of AuthorizeResponse.	The Charging Station SHALL update **the Authorisation Cache entry.**	The update is to be done with the IdTokenInfo value from the response as described under Authorization Cache.
Old	C10.FR.05	Upon receipt of TransactionEventResponse.	The Charging Station SHALL update the cache entry, if the IdToken is not in the Local Authorization List.	The update is to be done with the IdTokenInfo value from the response as described under Authorization Cache.
New	C10.FR.05	Upon receipt of TransactionEventResponse.	The Charging Station SHALL update the **Authorisation Cache entry.**	The update is to be done with the IdTokenInfo value from the response as described under Authorization Cache.
Old	C10.FR.06	Upon receipt of ReserveNowRequest.	The Charging Station SHALL update the cache entry, if the IdToken is not in the Local Authorization List.	The update is to be done with the IdTokenInfo value from the request as described under Authorization Cache.
New	C10.FR.06	Upon receipt of ReserveNowRequest.	The Charging Station SHALL update **the Authorisation Cache entry.**	The update is to be done with the IdTokenInfo value from the request as described under Authorization Cache.

Version	ID	Precondition	Requirement definition	Note
Old	C10.FR.07	If new identifier authorization data is received and the Authorization Cache is full.	The Charging Station SHALL remove any entries with status other than *Accepted*, and then, if necessary, the oldest valid entries to make space for the new entry.	This happens for example when the maximum cache size as configured by the Charging Station Manufacturer has been reached
New	C10.FR.07		The Charging Station SHALL **have a mechanism to accept new cache entries even when it is full, by deleting older entries.**	**It is suggested to remove any entries with status other than *Accepted* first, and then the oldest valid entries to make space for the new entry.**
Old	C10.FR.10	When the validity of an Authorization Cache entry expires.	The Authorization Cache entry SHALL be changed to expired in the Cache.	
New	C10.FR.10	When the validity of an Authorization Cache entry expires.	The Authorization Cache entry SHALL be **removed from the cache or changed to `Expired`.**	

3.60. (Minor) - Page 84, Use case C11, minor change to objective of use case

The text in the objective of the use case has changed.

Old text	**Objective(s)**	To clear all latest received IdTokens in the Authorization Cache.
New text	**Objective(s)**	To clear all IdTokens in the Authorization Cache.

3.61. (Medium) - Page 84, Use case C11, rephrasing requirements and adding one

Changed requirements:

Version	ID	Precondition	Requirement definition
Old	C11.FR.03		The Charging Station SHALL send ClearCacheResponse message with the status *Accepted* in case it was able to clear its Authorization Cache.
New	C11.FR.03	**C11.FR.02 AND Charging Station successfully cleared its Authorization Cache.**	The Charging Station SHALL send ClearCacheResponse message with the status *Accepted*.
Old	C11.FR.04		The Charging Station SHALL send ClearCacheResponse message with the status *Rejected* in case it was not able to clear its Authorization Cache.
New	C11.FR.04	**C11.FR.02 AND Configuration variable `AuthCacheEnabled` is false**	The Charging Station SHALL send ClearCacheResponse message with the status *Rejected*.

New requirement:

ID	Precondition	Requirement definition
C11.FR.05	C11.FR.02 AND Charging Station failed to clear its Authorization Cache.	The Charging Station SHALL send ClearCacheResponse message with the status *Rejected*.

3.62. (Medium) - Page 86, Use case C12, changes to requirements

Inconsistencies and errors have been fixed.

Changed requirements:

Version	ID	Precondition	Requirement definition	Note
Old	C12.FR.02	When an identifier is presented that is stored in the Authorization Cache as *Accepted*.	The Charging Station SHALL send a TransactionEventRequest to the CSMS.	
New	C12.FR.02	When an identifier is presented that is stored in the Authorization Cache as *Accepted*.	The Charging Station SHALL send a TransactionEventRequest **with *idToken*** to the CSMS.	
Old	C12.FR.03	C12.FR.03	The CSMS SHALL check the authorization status of the IdToken when processing this TransactionEventRequest.	
New	C12.FR.03	**C12.FR.02**	The CSMS SHALL check the authorization status of the IdToken when processing this TransactionEventRequest.	
Old	C12.FR.04	C12.FR.03 AND The cable is plugged in.	The Charging Station SHALL start the energy offer.	
New	C12.FR.04	**C12.FR.02** AND The cable is plugged in.	The Charging Station SHALL start the energy offer.	

Deleted requirements:

ID	Precondition	Requirement definition	Note
C12.FR.01	When the status of a connector changes.	The Charging Station SHALL send a StatusNotificationRequest to the CSMS.	E.g. when a cable is plugged in or an IdToken is presented.
C12.FR.07	If new identifier authorization data is received and the Authorization Cache is full.	The Charging Station SHALL remove any not *Accepted* entries, and then, if necessary, the oldest entries with status *Accepted* to make space for the new entry.	
C12.FR.08	Identifiers that were valid but are expired due to passage of time	Identifiers SHALL be rejected.	

3.63. (Minor) - Page 87, Use case C13, improved description

The text in the description of the use case has changed.

Old text	**Description**	This use case describes how to authorize an IdToken, while communication with the CSMS is not possible. The Local Authorization List is a list of idTokens that can be synchronized with the CSMS. The list contains the authorization status of all (or a selection of) idTokens and their authorization status/expiration date.
New text	**Description**	This use case describes how to authorize an IdToken, when communication with the CSMS is not possible. The Local Authorization List is a list of idTokens that can be synchronized with the CSMS. The list contains the authorization status of **a selected set of idTokens as managed by the CSMS.**

3.64. (Medium) - Page 88, Use case C13, changes to requirements

Updated for use with configuration variable `OfflineTxForUnknownIdEnabled`.

Changed requirement:

Version	ID	Precondition	Requirement definition	Note
Old	C13.FR.02	If *Offline*.	Identifiers that are present in a Local Authorization List that have a status other than *Accepted* SHALL be rejected.	
New	C13.FR.02	**If configuration variable `OfflineTxForUnknownIdEnabled` is false AND** The Charging Station is offline.	**Only identifiers that are present in a Local Authorization List that have a status *Accepted* SHALL be allowed to start a transaction.**	

New requirements:

ID	Precondition	Requirement definition	Note
C13.FR.04	If configuration variable `OfflineTxForUnknownIdEnabled` is true AND The Charging Station is offline.	Any identifier SHALL be allowed to start a transaction.	

3.65. (Medium) - Page 91, Use case C15, changes to requirements

Fixed errors in requirements.

Changed requirement:

Version	ID	Precondition	Requirement definition	Note
Old	C15.FR.03	When 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 transaction and send TransactionEventRequest (eventType = Updated) with *trigger* set to *Deauthorized* and state set to *SuspendedEVSE*.	
New	C15.FR.03	**C15.FR.02 AND** When 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 transaction and send TransactionEventRequest (eventType = Updated) with *trigger* set to *Deauthorized* and state set to *SuspendedEVSE*.	
Old	C15.FR.04	When the authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is *true* AND `TxStopPoint` does contain: (Authorized OR PowerPathClosed OR EnergyTransfer)	The Charging Station SHALL stop the transaction and send TransactionEventRequest (eventType = Ended) with stopReason set to *Deauthorized*.	
New	C15.FR.04	**C15.FR.02 AND** When the authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is *true* AND `TxStopPoint` does contain: (Authorized OR PowerPathClosed OR EnergyTransfer)	The Charging Station SHALL stop the transaction and send TransactionEventRequest (eventType = Ended) with stopReason set to *Deauthorized*.	
Old	C15.FR.06	When `StopTxOnInvalidId` is set to *false*	Energy delivery to the EV SHALL be stopped.	
New	C15.FR.06	**C15.FR.01 AND** `StopTxOnInvalidId` is set to *false* **AND** **`MaxEnergyOnInvalidId` is not set.**	Energy delivery to the EV SHALL be stopped immediately.	

Version	ID	Precondition	Requirement definition	Note
Old	C15.FR.07	In the case of an invalid identifier.	An operator MAY choose to charge the EV with a limited amount of energy so the EV is able to drive away. This amount is controlled by the optional Configuration Variable: `MaxEnergyOnInvalidId`.	
New	C15.FR.07	**C15.FR.01 AND** `StopTxOnInvalidId` is set to *false* **AND** **`MaxEnergyOnInvalidId` is set.**	**Energy delivery to the EV SHALL be allowed until the amount of energy specified in `MaxEnergyOnInvalidId` has been reached.**	

3.66. (Medium) - Page 117, Use case E05, Requirements regarding 'MaxEnergyOnInvalidId' incorrect/incomplete

Changed requirement:

Version	ID	Precondition	Requirement definition	Note
Old	E05.FR.02	When receiving a TransactionEventResponse with authorization status NOT Accepted AND MaxEnergyOnInvalidId has not been set or has been exceeded.	The Charging Station SHALL stop the energy delivery to the EV.	This is a not authorized use of the Charging Station.
New	E05.FR.02	E05.FR.01 AND The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is set to *false* AND `MaxEnergyOnInvalidId` is not implemented or has been exceeded. `TxStopPoint` does NOT contain: (PowerPathClosed OR EnergyTransfer)	The Charging Station SHALL stop the energy delivery to the EV immediately and send TransactionEventRequest (eventType = Updated) with *triggerReason* set to *ChargingStateChanged* and chargingState set to *SuspendedEVSE*	
Old	E05.FR.03	When receiving a TransactionEventResponse with authorization status NOT Accepted AND MaxEnergyOnInvalidId has NOT been exceeded.	The Charging Station SHALL continue the energy delivery to the EV until this value has been reached.	This is a not authorized use of the Charging Station.
New	E05.FR.03	E05.FR.01 AND The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is set to *false* AND `MaxEnergyOnInvalidId` is set and has NOT been exceeded.	Energy delivery to the EV SHALL be allowed until the amount of energy specified in `MaxEnergyOnInvalidId` has been reached.	

Removed requirement:

ID	Precondition	Requirement definition
E05.FR.07	If `MaxEnergyOnInvalidId` is supported AND set	The Charging Station SHALL allow charging up to the given amount of energy.

Added requirements:

ID	Precondition	Requirement definition
E05.FR.09	E05.FR.01 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*.
E05.FR.10	E05.FR.01 AND The authorization status in TransactionEventResponse is not *Accepted* AND The transaction is still ongoing AND `StopTxOnInvalidId` is *true* AND `TxStopPoint` does contain: (Authorized OR PowerPathClosed OR EnergyTransfer)	The Charging Station SHALL stop the transaction and send TransactionEventRequest (eventType = Ended) with *triggerReason* set to *Deauthorized* and stoppedReason set to *DeAuthorized*.
E05.FR.11	E05.FR.10 AND If the Charging Station has the possibility to lock the Charging Cable	The Charging Station SHOULD keep the Charging Cable locked until the owner presents his identifier.

3.67. (Minor) - Page 94, Use case C16: removed duplicate requirement

Removed requirement

Requirement C16.FR.02 and C16.FR.06 where the same requirement.

ID	Precondition	Requirement definition
C16.FR.06	User presents an IdToken that has groupId equal to `MasterPassGroupId` AND the Charging Station does not have a UI.	The Charging Station SHALL stop all ongoing transactions.

3.68. (Minor) - Page 97, Use case D01: Missing requirements about *versionNumber*

Change requirement D01.FR.01 to mention the versionNumber and add a requirement that versionNumber must be greater than zero.

Changed requirement:

Version	ID	Precondition	Requirement definition	Note
Old	D01.FR.01		SendLocalListRequest SHALL contain the type of update (full or differential) that the Charging Station MUST associate with he Local Authorization List after it has been updated.	
New	D01.FR.01		SendLocalListRequest SHALL contain the type of update (**updateType**) and a version number (**versionNumber**) that the Charging Station MUST associate with the Local Authorization List after it has been updated.	
Old	D01.FR.03	If the *status* in SendLocalListResponse is `Failed` or `VersionMismatch` and the *updateType* was `Differential`	The CSMS SHALL send the full Local Authorization List.	When this list is too large for one message (see D01.FR.11), it shall start by sending an initial list with *updateType* `Full` and adding identifiers using *updateType* `Differential` until the list is completely sent.

Version	ID	Precondition	Requirement definition	Note
New	D01.FR.03	If the *status* in SendLocalListResponse is `Failed` or `VersionMismatch` and the *updateType* was `Differential`	**It is RECOMMENDED that the CSMS sends** the full Local Authorization List.	When this list is too large for one message (see D01.FR.11), it shall start by sending an initial list with *updateType* `Full` and adding identifiers using *updateType* `Differential` until the list is completely sent.

New requirement:

ID	Precondition	Requirement definition	Note
D01.FR.18		**versionNumber** in a SendLocalListRequest SHALL be greater than 0.	In GetLocalListVersionResponse the versionNumber = 0 has a special meaning: No Local List installed. So the value 0 should never be used.

3.69. (Minor) - Page 97, Use case D01: Requirements changes related to *versionNumber*

Some requirements in this use case were already covered by use case C. They have been removed.

Removed requirements:

ID	Precondition	Requirement definition	Note
D01.FR.07		The Charging Station MAY authorize the IdToken locally without involving the CSMS.	As described in C13 - Offline Authorization through Local Authorization List.
D01.FR.08	When an IdTokenType is presented that is stored in the Local Authorization List as invalid and the Charging Station is online.	AuthorizeRequest SHALL be sent to the CSMS to check the current state of the identifier.	
D01.FR.14		Identifiers known in the Local Authorization List SHALL NOT be added to the Authorization Cache.	

Changes to clarify the usage of *versionNumber Changed requirements:*

Version	ID	Precondition	Requirement definition	Note
Old	D01.FR.04	If no localAuthorizationList (or an empty one) is given and the updateType is *Full*.	The Charging Station SHALL remove all IdTokens from the list.	
New	D01.FR.04	If no localAuthorizationList (or an empty one) is given and the *updateType* is `Full`.	The Charging Station SHALL remove all IdTokens from the list.	**Note, that the version number of the list is still updated to value of *versionNumber* in the request.**
Old	D01.FR.05		Requesting a Differential update without or with empty localAuthorizationList SHALL have no effect on the list.	
New	D01.FR.05		Requesting a Differential update without or with empty localAuthorizationList SHALL have no effect on the list.	**Note, that the version number of the list is still updated to value of *versionNumber* in the request.**

Version	ID	Precondition	Requirement definition	Note
Old	D01.FR.16	If the Charging Station receives a SendLocalListRequest with updateType is *Differential* AND localAuthorizationList contains AuthorizationData elements with idTokenInfo	The Charging Station SHALL update its Local Authorization List with these elements.	Add them if not yet present, update with new information when already present in the Local Authorization List.
New	D01.FR.16	If the Charging Station receives a SendLocalListRequest with *updateType* is `Differential` AND localAuthorizationList contains AuthorizationData elements with idTokenInfo	The Charging Station SHALL update its Local Authorization List with these elements **and set the version number to the value specified in the message.**	Add them if not yet present, update with new information when already present in the Local Authorization List.
Old	D01.FR.17	If the Charging Station receives a SendLocalListRequest with updateType is *Differential* AND localAuthorizationList contains AuthorizationData elements without idTokenInfo	The Charging Station SHALL remove these elements from its Local Authorization List.	
New	D01.FR.17	If the Charging Station receives a SendLocalListRequest with *updateType* is `Differential` AND localAuthorizationList contains AuthorizationData elements without idTokenInfo	The Charging Station SHALL remove these elements from its Local Authorization List **and set the version number to the value specified in the message.**	

New requirement added to describe how to deal with a version mismatch.

New requirement:

ID	Precondition	Requirement definition	Note
D01.FR.19	If the Charging Station receives a SendLocalListRequest with *updateType* = `Differential` AND *versionNumber* is less or equal to the version number of its Local Authorization List	The Charging Station SHALL refuse to update its Local Authorization List and SHALL return a SendLocalListResponse with *status* set to `VersionMismatch`.	

3.70. (Minor) - Page 99, Use case D02, Prerequisite and Remarks changed

The prerequisite in the use case has been removed and added to the requirements instead. A remark was added to further clarify the meaning of *versionNumber* = 0.

Old text	**Prerequisite(s)**	Local Authorization List is enabled with Configuration Variable `LocalAuthListEnabled`.

New text	**Prerequisite(s)**	*<empty>*
Old text	**Remark(s)**	n/a
New text	**Remark(s)**	A *versionNumber* of 0 (zero) is reserved to indicate that no local authorization list exists, either because it is not enabled or because it has not yet received any update from CSMS and thus does not have a version number to return. In contrast, a local authorization list that was emptied, because CSMS sent a SendLocalListRequest with an empty *localAuthorizationList*, does have a *versionNumber* > 0.

3.71. (Minor) - Page 102, Section 1.3.2, Sequence number, 32 bit unsigned integers are not supported by JSON schemas

Changed text:

Old text	When a TransactionEventRequest has to be created, the Charging Station SHALL set the message's seqNo field to the value of a transaction event request counter maintained for the EVSE on which the transaction is occurring. An unsigned 32 bit integer should be used for the counter. Immediately after taking the counter value, the Charging Station SHALL update the counter value as follows: - If the counter's value is smaller than 4294967295, the counter's value is incremented - If the counter's value is 4294967295, the counter's value is set to 0
New text	When a TransactionEventRequest has to be created, the Charging Station SHALL set the message's seqNo field to the value of a transaction event request counter maintained for the EVSE on which the transaction is occurring. Immediately after taking the counter value, the Charging Station SHALL update the counter value as follows: - If the counter's value is smaller than **2147483647**, the counter's value is incremented. - If the counter's value is **2147483647**, the counter's value is set to 0.

3.72. (Major) - Page 106, Use case E01, Start Transaction Options: adding *context* Transaction.Begin/End to MeterValue

Since energy consumption during a transaction is usually calculated as the difference between the value of the measurand `Energy.Active.Import.Register` at the end and start of the transaction, it is important to properly mark the *context* of these sampled values with `Transaction.Begin` and `Transaction.End`. If the start point of a transaction is configured such that the EVSE (and thus the meter) is not yet known at that moment, then Charging Station shall send sampled value(s) of the meter with *context* = `Transaction.Begin` in the TransactionEventRequest(eventType=`Updated`) that occurs when charging starts.

3.72.1. Page 107, E01 - Requirements

Change the following requirement:

E01.FR.09	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.

Add the following requirement:

E01.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.

3.72.2. Page 110, E02 - Requirements

Change the following requirement:

E02.FR.09	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.

Add the following requirement:

E02.FR.20	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.

3.72.3. Page 112, E03 - Requirements

Change the following requirement:

E03.FR.07	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.

Add the following requirement:

E03.FR.11	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.

3.72.4. Page 115, E04 - Requirements

Change the following requirement:

E04.FR.05	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.

Add the following requirement:

E04.FR.11	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.

3.72.5. Page 117, E05 - Requirements

Change the following requirement:

E05.FR.05	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.

Add the following requirement:

E05.FR.08	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.

3.72.6. Page 123, E06 - Requirements

Change the following requirement:

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.

Add the following requirement:

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.

3.72.7. Page 124, E07 - Requirements

Change the following requirement:

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.

Add the following requirement:

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.

3.73. (Medium) - Page 107, E01 - Start Transaction options - Requirements, Missing requirement for at least once sending EVSE.id and EVSE.connectorId during a regular transaction

New requirement:

ID	Precondition	Requirement definition
E01.FR.16	After the EV is connected with the Charging Station.	The next TransactionEventRequest SHALL contain *evse.id* AND *evse.connectorId*.

3.74. (Medium) - Page 110 & 112, Use case E02 & E03, A TransactionEventRequest Message needs to be sent for occurred trigger event

If a TransactionEventRequest messages would be omitted, the CSMS will miss important information.

New requirement:

ID	Precondition	Requirement definition	Note
E02.FR.17	When a transaction-related trigger event occurs, listed in *TriggerReasonEnumType* AND the transaction is ongoing.	The Charging Station SHALL send a TransactionEventRequest with a triggerReason corresponding to the occurred event.	When two trigger reasons overlap, the more specific one should be used. For example, when a cable is plugged in, triggerReason *CablePluggedIn* should be used, not *EVDetected*. When two events occur at the same time, they need transmitted using two separate TransactionEventRequest messages. This is to prevent information loss, when something goes wrong.
E03.FR.12	When a transaction-related trigger event occurs, listed in *TriggerReasonEnumType* AND the transaction is ongoing.	The Charging Station SHALL send a TransactionEventRequest with a triggerReason corresponding to the occurred event.	When two trigger reasons overlap, the more specific one should be used. For example, when a cable is plugged in, triggerReason *CablePluggedIn* should be used, not *EVDetected*. When two events occur at the same time, they need transmitted using two separate TransactionEventRequest messages. This is to prevent information loss, when something goes wrong.

3.75. (Medium) - Page 112, The TxStartPoint and TxStopPoint configuration variables control when transactions start and stop

The TxStopPoint configuration variable controls when a transaction will end. A plugin cable timeout should only cause the transaction to be deauthorized, unless the TxStopPoint configuration variable contains the value *EVConnected*. A few requirements are describing this incorrectly and therefore need be changed.

3.75.1. Page 112, Use case E03

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	E03.FR.05	E03.FR.04	The Charging Station SHALL **end** the transaction and send a TransactionEventRequest (**eventType = Ended, stoppedReason = Timeout**) to the CSMS.
New	E03.FR.05	E03.FR.04	The Charging Station SHALL **deauthorize** the transaction and send a TransactionEventRequest (**triggerReason = EVConnectionTimeout**) to the CSMS.

3.75.2. Page 133, Use case E10

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	E10.FR.02	Cable permanently attached AND Cable not plugged in within timeout	The Charging Station SHALL **stop** the transaction.
New	E10.FR.02	Cable permanently attached AND Cable not plugged in within timeout	The Charging Station SHALL **deauthorize** the transaction.

3.76. (Minor) - Page 114 & 119, Use Case E02 & E03, Missing requirements for the use of numberOfPhasesUsed field

New requirements:

ID	Precondition	Requirement definition
E02.FR.18	When the energy transfer starts AND If the Charging Station is able to report the number of phases used	The Charging Station SHALL provide the number of phases used, using the numberOfPhasesUsed field.
E02.FR.19	E02.FR.18 AND during the transaction the number of phases used changes	The Charging Station SHALL provide the adjusted number of phases used, using the numberOfPhasesUsed field.
E03.FR.13	When the energy transfer starts AND If the Charging Station is able to report the number of phases used	The Charging Station SHALL provide the number of phases used, using the numberOfPhasesUsed field.
E03.FR.14	E03.FR.13 AND during the transaction the number of phases used changes	The Charging Station SHALL provide the adjusted number of phases used, using the numberOfPhasesUsed field.

Incorrect description for the numberOfPhasesUsed field.

Changed description:

Old text	Optional. The actual number of phases, a connected EV uses to draw power. When omitted, the currently used number of phases can be determined by the CSMS according to (the lower number has priority): 1: The last numberOfPhasesUsed sent. 2: The numberPhases in the currently used ChargingSchedule. 3: The number of phases provided via device management. 4: Assume 3 phases as the last fallback.
New text	Optional. **If the Charging Station is able to report the number of phases used, then it SHALL provide it. When omitted the CSMS may be able to determine the number of phases used via device management.**

3.77. (Medium) - Page 123, Stop a transaction when a Fault or Failure occurs

The specification does not describe a way to report a stopped transaction caused by an Abnormal Error or Fault Condition. Use case E06 needs to be extended with an extra Sub-Use Case, which will describe this.

New requirement:

ID	Precondition	Requirement definition
E06.FR.16	A transaction was stopped by an Abnormal Error or Fault Condition.	The Charging Station SHALL send *TransactionEventRequest(eventType = Ended, TriggerReason = AbnormalCondition)* to the CSMS.

The value *AbnormalCondition* needs to be added to *TriggerReasonEnumType*, to support this Sub-Use Case.

Added Enum value:

Page	section	Message/dataType	Value	Description
372	2.80	TriggerReasonEnumType	AbnormalCondition	An Abnormal Error or Fault Condition has occurred.

New Schema files are available.

3.78. (Medium) - Page 128, Incorrect descriptions StopTxOnInvalidId and StopTxOnEVSideDisconnect

In older versions of OCPP there were two configuration keys which caused the transaction to end in certain situations; StopTxOnInvalidId and StopTxOnEVSideDisconnect. These were also added to OCPP 2.0, but with the introduction of the new transaction mechanism and its TxStart and TxStop point configuration variables it is in some cases not needed for the transaction to end. Only if the TxStopPoint contains *Authorized* the transaction should end or in the case of *StopTxOnEVSideDisconnect EVConnected* AND/OR *Authorized*.

3.78.1. Page 129, Use case E09

The following requirement also needs to be adjusted, because of the description change of *StopTxOnEVSideDisconnect*

Changed requirement:

Version	Id	Precondition	Requirement definition	Note
Old	E09.FR.01	If StopTxOnEVSideDisconnect = true .	The transaction SHALL be **stopped** when the cable is disconnected from the EV. If the EV is reconnected, energy transfer is not allowed until the transaction is **stopped and a new transaction is started.**	Setting StopTxOnEVSideDisconnect to true will prevent sabotage acts **to stop the energy flow by** unplugging not locked cables on EV side.
New	E09.FR.01	If StopTxOnEVSideDisconnect = true .	The transaction SHALL be **deauthorized** when the cable is disconnected from the EV. If the EV is reconnected, energy transfer is not allowed until the transaction is **authorized once again**.	Setting StopTxOnEVSideDisconnect to true will prevent sabotage acts **when** unplugging not locked cables on EV side.

3.78.2. Page 389, section 2.6.3, StopTxOnEVSideDisconnect

Changed description:

Old Description	When set to true, the Charging Station SHALL **administratively stop** the transaction when the cable is unplugged from the EV.
New Description	When set to true, the Charging Station SHALL **deauthorize** the transaction when the cable is unplugged from the EV.

3.78.3. Page 390, section 2.6.8, StopTxOnInvalidId

Changed description:

Old Description	whether the Charging Station will **stop** an ongoing transaction when it receives a non- Accepted authorization status in TransactionEventResponse for this transaction.
New Description	whether the Charging Station will **deauthorize** an ongoing transaction when it receives a non- Accepted authorization status in TransactionEventResponse for this transaction.

3.79. (Minor) - Page 144, Use case F01, rephrased Postcondition

Old postcondition	The Charging Station tries to start a transaction, if the value of `AuthorizeRemoteStart` is *true*, the Charging Station will first try to authorize the IdToken, using Local Authorization List, Authorization Cache and/or an AuthorizeRequest.
New postcondition	The Charging Station **offers energy.** If the value of `AuthorizeRemoteStart` is *true*, the Charging Station will **only offer energy when it successfully authorized** the IdToken, using Local Authorization List, Authorization Cache and/or an AuthorizeRequest.

3.80. (Minor) - Page 145 & 148, Use case F01 & F02,

Addition to remark(s) F01 & F02:

Remark	The CSMS is allowed to send a RequestStartTransactionRequest with IdTokenType *NoAuthorization*. The operator should be aware that if the Charging Station supports local stop transaction, this transaction can be stopped by anyone.

3.81. (Minor) - Page 153, Use Case F05, Missing requirement for handling a UnlockConnectorRequest when no EV is connected

New requirement:

ID	Precondition	Requirement definition
F05.FR.06	F05.FR.01 AND No cable is connected to the connector.	The Charging Station SHALL attempt to unlock the connector, even if no cable is detected and SHALL return the result of the unlock attempt.

3.82. (Major) - Page 157, Use case F06: Refuse TriggerMessage for BootNotification after being accepted

Add the following requirement to table 133.

ID	Precondition	Requirement definition	Note
F06.FR.17	If Charging Station receives a TriggerMessageRequest with *requestedMessage* set to: *BootNotification* AND the response it received from CSMS to the last BootNotificationRequest was: *Accepted*	Charging Station SHALL respond with a TriggerMessageResponse with status *Rejected*.	A trigger to request a Charging Station to send a BootNotification is only meant to be used when the BootNotification has not yet been accepted.

3.83. (Minor) - Page 160, Use case G01, No explanation about the deprecation of the StatusNotificationRequest message

Addition to remark(s) G01:

Remark	Notifying a connector status from the Charging Station to the CSMS will be taken over by the new Device Management Monitoring feature, however this mechanism has not been proven in the field yet. So the old StatusNotificationRequest message remains available for use for now.

3.84. (Major) - Page 163, Use case G03 & G04, Missing/incorrect explanation about the impact of the Charging Station, EVSE and Connector level on each other and the persistence of the statuses

If the Charging Station, an EVSE or a connector is not available for charging it is deemed Inoperative.

New requirements:

ID	Precondition	Requirement definition	Note
G03.FR.06	When the availability of an EVSE becomes Inoperative (*Unavailable*, *Faulted*)	All operative connectors (i.e. not *Faulted*) of that EVSE SHALL become *Unavailable*.	
G03.FR.07	When the availability of an EVSE becomes Operative	The Charging Station SHALL revert the status of all connectors of that EVSE to their original status.	See Note 1.

ID	Precondition	Requirement definition	Note
G03.FR.08	When the availability of an EVSE or Connector has been set explicitly via ChangeAvailabilityRequest	The set availability state SHALL be persistent across reboot/power loss.	
G04.FR.07	When the availability of the Charging Station becomes Inoperative (*Unavailable*, *Faulted*)	All operative EVSEs and connectors (i.e. not *Faulted*) SHALL become *Unavailable*.	
G04.FR.08	When the availability of the Charging Station becomes Operative	The Charging Station SHALL revert the status of all EVSEs and connectors to their original status.	See Note 1.
G04.FR.09	When the availability of a Charging Station has been set explicitly via ChangeAvailabilityRequest	The set availability state SHALL be persistent across reboot/power loss.	

NOTE 1. The Charging Station, EVSEs and Connectors have separate / individual states. This means (for example) that when setting a connector to Inoperative, then setting the connected EVSE to Inoperative and thereafter change the EVSE back to operative, the connector will remain Inoperative.

NOTE 2. It is only required to report a status change of a connector. StatusNotificationRequest only supports the reporting of connector statuses.

Changed requirement:

Version	Id	Precondition	Requirement definition	Note
Old	G04.FR.01	In the case ChangeAvailabilityRequest contains evseId = 0	The Charging Station status change SHALL apply to the Charging Station and all EVSEs.	
New	G04.FR.01	In the case the evse field is omitted in ChangeAvailabilityRequest	The Charging Station status change SHALL apply to the whole Charging Station.	

Removed remark, Use case G04:

The remark is not needed, because it is already handled by the requirements.

Old remark	In the case the ChangeAvailabilityRequest contains evseId = 0, the status change applies to the Charging Station and all EVSEs.

3.85. (Major) - Page 163, Use case G03, it is not possible to change the availability of a connector with ChangeAvailabilityRequest

It needs to be possible to change the availability of a connector. For example if the EVSE has 2 connectors, one might be a broken, so cannot be used, but the other might still be usable.

Page	Section	Message/dataType	Field	Type	Card.	Description	Action
299	1.5.1	ChangeAvailabilityRequest	evseId	integer	1..1	Required. The id of the EVSE for which availability needs to change. Id '0' (zero) is used if the availability of the Charging Station and all its EVSEs needs to change.	removed
299	1.5.1	ChangeAvailabilityRequest	EVSE	EVSEType	0..1	Optional. Contains Id's to designate a specific EVSE/connector by index numbers. When omitted, the message refers to the Charging Station as a whole.	added

Rename Use case G03:

Old name	Change Availability EVSE

New name	Change Availability EVSE/Connector

Changed Successful post condition, Use case G03:

Old post condition	When changing the availability of an EVSE to Operative, the status of the EVSE has changed to Available, Occupied or Reserved. When changing the availability of an EVSE to Inoperative , the status of the EVSE has changed to Unavailable or Faulted.
New post condition	When changing the availability of an **EVSE/connector** to Operative, the status of the component has changed to either Available, Occupied or Reserved. When changing the availability of an **EVSE/connector** to Inoperative , the status of the EVSE has changed to **Unavailable.**

Changed Successful post condition, Use case G04:

Old post condition	The CSMS was able to change the Charging Station' availability. When changing the availability of a Charging Station to Operative, the status of the Charging Station has changed to Available. When changing the availability of a Charging Station to Inoperative, the status of the Charging Station has changed to Unavailable or Faulted.
New post condition	The CSMS was able to change the **availability of the Charging Station.** When changing the availability of a Charging Station to Operative, the status of the Charging Station has changed to Available. When changing the availability of a Charging Station to Inoperative, the status of the Charging Station has changed to **Unavailable.**

3.86. (Minor) Page 170, Use case H01

This use case inadvertently used the word 'Connector' where 'EVSE' was meant. It has been changed to refer to 'EVSE'.

3.87. (Major) - Page 172, Requirements use case H01

The following requirements in *Table 145. H01 - Requirements* are changed: (changes marked in **bold**)

ID	Precondition	Requirement definition	Note
H01.FR.04	If the Charging Station receives a ReserveNowRequest **without *evseId*** AND at least one EVSE is *Available* AND H01.FR.18	The Charging Station SHALL accept the reservation AND respond with a ReserveNowResponse with status *Accepted*.	
H01.FR.05			***Requirement deleted***
H01.FR.06	If the Charging Station receives a ReserveNowRequest with a connector type AND at least one EVSE with the specified connector type is *Available* **AND H01.FR.18**	The Charging Station SHALL accept the reservation AND respond with a ReserveNowResponse with status *Accepted*.	
H01.FR.07	When the Charging Station has *Accepted* a ReserveNowRequest **without *evseId***	The Charging Station SHALL make sure that at any time during the validity of the reservation, one EVSE remains available for the reserved IdTokenType.	
H01.FR.08			***Requirement deleted***
H01.FR.10			***Requirement deleted***
H01.FR.11	When receiving a ReserveNowRequest AND (all) targeted **EVSEs** have status *Reserved*	The Charging Station SHALL return *Occupied*.	
H01.FR.12	When receiving a ReserveNowRequest AND (all) targeted **EVSEs** have status *Faulted*	The Charging Station SHALL return *Faulted*.	

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*.	
H01.FR.14	When receiving a ReserveNowRequest AND (all) targeted **EVSEs** have status *Unavailable*	The Charging Station SHALL return *Unavailable*.	
H01.FR.16	When the status of a targeted **EVSE** changes to *Faulted*	The Charging Stations SHALL cancel the reservation AND send a ReservationStatusUpdate with status *Removed*.	
H01.FR.17	When the status of a targeted **EVSE** changes to *Unavailable*	The Charging Stations SHALL cancel the reservation AND send a ReservationStatusUpdate with status *Removed*.	
H01.FR.18	If the Configuration Variable: `ReservationNonEvseSpecific` is set to *true*.	The Charging Station SHALL accept reservations on an unspecified **EVSE**.	
H01.FR.19	If the Configuration Variable: `ReservationNonEvseSpecific` is not set or set to *false*.	The Charging Station SHALL reject reservations on an unspecified **EVSE**.	
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*.	

3.88. (Minor) Page 174, Use case H02

This use case inadvertently used the word 'Connector' where 'EVSE' was meant. It has been changed to refer to 'EVSE'.

3.89. (Minor) Page 174, Use case H02

A remark was add why no reservation update message is sent in this case.

Old text	**Remark(s)**	n/a
New text	**Remark(s)**	**The Charging Station does not send a ReservationStatusUpdate, because it was explicitly cancelled by CSMS, so it is already aware of the event.**

3.90. (Minor) Page 174, Use case H03

This use case inadvertently used the word 'Connector' where 'EVSE' was meant. It has been changed to refer to 'EVSE'.

3.91. (Major) Page 177, Use case H03, all requirements rewritten

The requirements for using a reserved EVSE were inadequate and have been completely rewritten.

The following requirements have **replaced** the requirements from H03:

ID	Precondition	Requirement definition
H03.FR.01	Reservation is pending for *idToken* for a specific *evseId*	The Charging Station SHALL allow charging on that EVSE when IdToken presented for authorization matches *idToken*.
H03.FR.02	Reservation is pending for *idToken* for a specific *connectorType*	The Charging Station SHALL allow charging on an EVSE with a connector of type *connectorType* when IdToken presented for authorization matches *idToken*.
H03.FR.03	Reservation is pending for *idToken* without a specific *evseId* or *connectorType*	The Charging Station SHALL allow charging on an EVSE when IdToken presented for authorization matches *idToken*.
H03.FR.04	H03.FR.01 AND attribute *groupIdToken* in reservation has a value	The Charging Station SHALL allow charging on that EVSE when IdToken presented for authorization matches *idToken* or when the associated *groupIdToken* matches.

ID	Precondition	Requirement definition
H03.FR.05	H03.FR.02 AND attribute *groupIdToken* in reservation has a value	The Charging Station SHALL allow charging on an EVSE with a connector of type *connectorType* when IdToken presented for authorization matches *idToken* or when the associated *groupIdToken* matches.
H03.FR.06	H03.FR.03 AND attribute *groupIdToken* in reservation has a value	The Charging Station SHALL allow charging on an EVSE when IdToken presented for authorization matches *idToken*.
H03.FR.07	If attribute *groupIdToken* in the reservation has a value (it is optional).	In order to determine the *groupIdToken* that is associated with an incoming IdToken, the Charging Station MAY look it up in its Local Authorization List or Authorization Cache.
H03.FR.08	H03.FR.07 AND If it is not found in the Local Authorization List or Authorization Cache.	The Charging Station SHALL send an AuthorizeRequest for the incoming IdToken to the CSMS in order to get its associated *groupIdToken*.

3.92. (Minor) Page 178, Use case H04

This use case inadvertently used the word 'Connector' where 'EVSE' was meant. It has been changed to refer to 'EVSE'.

3.93. (Minor) Page 178, Use case H04, changes to requirements

Rephrased requirements:

Version	ID	Precondition	Requirements
Old	H04.FR.01	The reservation ends (expiryDateTime reached) AND If a specific EVSE was reserved for this reservation	The Charging Station SHALL allow Charging again on this EVSE.
New	H04.FR.01	The reservation ends (*expiryDateTime* reached)	The Charging Station SHALL **send a ReservationStatusUpdateRequest with status *Expired*.**
Old	H04.FR.02	H04.FR.01	The Charging Station SHALL send a StatusNotificationRequest with status *Available* to the CSMS, notifying the CSMS the EVSE is available again for any EV Driver.
New	H04.FR.02	H04.FR.01 AND **If a specific EVSE was reserved for this reservation**	**The Charging Station SHALL allow charging again on this EVSE.**
Old	H04.FR.03	The reservation ends (expiryDate reached)	The Charging Station SHALL send a ReservationStatusUpdateRequest with status *Expired*.
New	H04.FR.03	**H04.FR.02**	The Charging Station SHALL send a **StatusNotificationRequest with status *Available* to the CSMS, notifying the CSMS the EVSE is available again for any EV Driver.**

3.94. (Minor) Page 183, Use case I03: remark about offline stopped transaction improved

Old text	**Remarks**	When a transaction was stopped offline, see: E07 - Offline Stop Transaction and the Charging Station receives a TransactionEventRequest (eventType = Ended) containing the total cost of the transaction, it is advised not to show the cost on a display, the EV driver is probably no longer at the Charging Station. The scenario description and sequence diagram above are based on the Configuration Variable for stop transaction being configured as follows. `TxStopPoint`: ParkingBayOccupancy, EVConnected, Authorized This use-case is also valid for other configurations, but then the transaction might stop at another moment, which might change the sequence in which message are send. For more details see the use case: E06 - Stop Transaction options
New text	**Remarks**	If the Charging Station was offline when the transaction ended and the TransactionEventResponse with *totalCost* is received when the Charging Station comes back online some time after that, then there is no use in displaying the cost, because the user has likely left already. A similar situation applies when `TxStopPoint` is defined as ParkingBayOccupancy, in which case the EV must leave the Charging Station to cause the transaction to end. The scenario description and sequence diagram above are based on the Configuration Variable for stop transaction being configured as follows. `TxStopPoint`: ParkingBayOccupancy, EVConnected, Authorized This use-case is also valid for other configurations, but then the transaction might stop at another moment, which might change the sequence in which message are send. For more details see the use case: E06 - Stop Transaction options

3.95. (Minor) Page 184, Use case I03: change to text of precondition of requirement I03.FR.01

Version	Requirement	Precondition
Old text	I03.FR.01	When the EV Driver locally stops the transaction.
New text	I03.FR.01	When transaction is stopped

3.96. (Minor) Page 184, Use case I03: change to requirements

Change requirement:

Version	ID	Precondition	Requirement
Old	I03.FR.03	I03.FR.02	The Charging Station SHALL display the total cost to the EV Driver.
New	I03.FR.03	I03.FR.02 AND **Charging Station was online when transaction stopped**	The Charging Station SHALL display the total cost to the EV Driver.

New requirement:

ID	Precondition	Requirement
I03.FR.05	I02.FR.02 AND `TxStopPoint` is defined as ParkingBayOccupancy	The Charging Station SHOULD NOT display the total cost to the EV Driver. (Driver has left already).

3.97. (Minor) Page 186, Use case I05: change to text of precondition of requirement I05.FR.02

Version	Requirement	Precondition
Old text	I05.FR.02	When the EV Driver locally stops the transaction AND When the Charging Station cannot retrieve the total cost for the stopped transaction, because the Charging Station is offline.
New text	I05.FR.02	When the Charging Station cannot retrieve the total cost for the stopped transaction, because the Charging Station is offline.

3.98. (Minor) Page 187, Use case I06: added remark about tariff updates during a transaction

Since changing a tariff when a transaction is in progress might not always be allowed, the following remark was added:

8	**Remarks**	There may be a policy or a legal requirement in place, that the tariff communicated at the start of the transaction must be used for the entire transaction, in which case no updated tariff information should be sent during the transaction.

3.99. (Minor) - Page 189, Introduction, Not enough explanation about the deprecation of the MeterValuesRequest message

Old text	The MeterValuesRequest and MeterValuesResponse messages are deprecated in OCPP 2.0. It is advised to start using Device Management Monitoring instead, see N. Diagnostics.
New text	The transfer of the MeterValues from the Charging Station to the CSMS will be taken over by the new Device Management Monitoring feature, however this mechanism has not been proven in the field yet. So the old MeterValuesRequest message remains available for use for now.

3.100. (Minor) - Page 193, Use case J01, Requirement changes

The following requirement has been removed, because it is not essential to the use case. This text has instead been added as a remark to the use case.

Removed requirement:

ID	Precondition	Requirements	Note
J01.FR.12		The CSMS MAY query the `ConnectorPhaseRotation` Configuration Variable on the Charging Station using GetVariablesRequest.	For individual Connector phase rotation information.

The following requirements have been rephrased to make them clearer.

Rephrased requirements:

Version	ID	Precondition	Requirements	Note
Old	J01.FR.13		The Charging Station SHALL report the phase rotation in respect to the grid connection	If known to the Charging Station.
New	J01.FR.13	**When reporting phase rotation of a component**	The Charging Station SHALL report the phase rotation **relative** to the grid connection	
Old	J01.FR.15	J01.FR.14 AND Amount of measurands is too much for 1 MeterValuesRequest	The Charging Station MAY split measurands over multiple MeterValuesRequest messages.	
New	J01.FR.15	J01.FR.14 AND Amount of measurands is too much for 1 MeterValuesRequest	The Charging Station MAY **use multiple MeterValuesRequest messages to send all measurands.**	

3.101. (Minor) - Page 195, Use case J02, Sending transaction-related Meter Values

The following requirement has been removed, because it is not essential to the use case. This text has instead been added as a remark to the use case.

Removed requirement:

ID	Precondition	Requirements	Note
J02.FR.08		The CSMS MAY query the `ConnectorPhaseRotation` Configuration Variable on the Charging Station using GetVariablesRequest.	For individual Connector phase rotation information.

The following requirements have been rephrased to make them clearer.

Rephrased requirements:

Version	ID	Precondition	Requirements	Note
Old	J01.FR.16	SampledDataSignReadings is true	The Charging Station SHALL retrieve signed meter values and put them in the signedMeterValue field.	
New	J01.FR.16	`SampledDataSignReadings` is *true*	The Charging Station SHALL retrieve signed meter values **from components that support data signing** and put them in the *signedMeterValue* field.	**This will usually only apply to energy meters.**
Old	J02.FR.09	If known to the Charging Station	The Charging Station SHALL report the phase rotation in respect to the grid connection	
New	J02.FR.09	**When reporting phase rotation of a component**	The Charging Station SHALL report the phase rotation **relative** to the grid connection	
Old	J02.FR.14	J02.FR.11 AND Amount of measurands is too much for 1 MeterValuesRequest	The Charging Station MAY split measurands over multiple MeterValuesRequest messages.	
New	J02.FR.14	J02.FR.11 AND Amount of measurands is too much for 1 MeterValuesRequest	The Charging Station MAY **use multiple MeterValuesRequest messages to send all measurands.**	
Old	J02.FR.15	AlignedDataSignReadings is true	The Charging Station SHALL retrieve signed meter values and put them in the signedMeterValue field.	
New	J02.FR.15	`AlignedDataSignReadings` is *true*	The Charging Station SHALL retrieve signed meter values **from components that support data signing** and put them in the *signedMeterValue* field.	**This will usually only apply to energy meters.**

New requirements:

ID	Precondition	Requirements	Note
J01.FR.17		The timestamp of a MeterValue SHALL apply to all its SampledValues.	
J01.FR.18	When CSMS receives a MeterValuesRequest	CSMS SHALL respond with MeterValuesResponse.	Failing to respond with MeterValuesResponse might cause the Charging Station to try the same message again.
J02.FR.18		The timestamp of a MeterValue SHALL apply to all its SampledValues.	

ID	Precondition	Requirements	Note
J02.FR.19	When CSMS receives a TransactionEventRequest	CSMS SHALL respond with TransactionEventResponse.	Failing to respond with TransactionEventResponse might cause the Charging Station to try the same message again.

3.102. (Minor) - Page 196, Use case J03 - Charging Loop with metering information exchange

In the **Combined scenario description** step **1** has been changed, because the ISO 15118 MeteringReceiptReq only applies to AC charging:

Old	**1.** The EV sends a ChargingStatusReq (in case of AC charging) or CurrentDemandReq (in case of DC charging) message to the Charging Station.
New	**1.** The EV sends a ChargingStatusReq (in case of AC charging) message to the Charging Station.

Figure 91 has also been adapted accordingly.

3.103. (Major) - Page 197, Use case J03: requirements changed

The following requirements to J03 have been removed, because they are no requirements to OCPP, but to an ISO 15118 implementation on the Charging Station.

Removed requirements:

ID	Precondition	Requirement definition
J03.FR.01		See ISO15118-1, use case Requirements F1, first requirement, page 37.
J03.FR.02		See ISO15118-1, use case Requirements F1, second requirement, page 37.
J03.FR.03		See ISO15118-1, use case Requirements F1, third requirement, page 37.

The following requirement has been changed to be more precise.

Changed requirement:

*Version	ID	Precondition	Requirement definition
Old	J03.FR.04		See ISO15118-1, use case Requirements F1, fourth requirement, page 37. This should be done using the SignedMeterValue element of the SampledValue
New	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.

3.104. (Minor) - Page 201, After 3.2: New section "Charging Profile Kinds"

Add a new section "3.3 Charging Profile Kinds" after section 3.2.

This section explains the different kind of charging schedules that can be use in a charging profile:

Purpose	Description
Absolute	The charging schedule periods are relative to an absolute point in time defined in the schedule. This requires that *startSchedule* is set to a starting point in time. Use this, for example, to define a schedule that reduces charging between 17:00h and 21:00h, regardless of when charging session was started.
Recurring	The charging schedule restarts periodically at the first schedule period. To be most useful, this requires that *startSchedule* is set to a starting point in time. Use this in combination with *recurrencyKind* = `Daily`, for example, to define a schedule that reduces charging between 17:00h and 21:00h every day, regardless of when charging session was started.

Purpose	Description
Relative	Charging schedule periods start when ChargingProfile is activated. In this case no value for *startSchedule* should be supplied.

3.105. (Minor) - Page 203, K. section 3.4, Examples added

An example of using stacked charging profiles has been added to this section.

3.106. (Minor) - Page 204, K. section 3.6, More precise description

Old text	[…] As mentioned before, for each charging profile purpose, at any point in time, the leading charging profile for that purpose is the profile with the highest stackLevel that is valid at that time, as determined by their validFrom and validTo parameters. […]
New text	[…] As mentioned before, for each charging profile purpose, at any point in time, the leading charging schedule for that purpose is the **charging schedule that has a schedule period defined for that time and that belongs to a charging** profile with the highest stack level that is valid at that time, as determined by their *validFrom* and *validTo* parameters. […]

3.107. (Minor) - Page 207, section 5.1, K01 SetChargingProfile, requirement K01.FR.06 changed

3.107.1. It should be allowed to have the same stackLevel - chargingProfilePurpose combination on different EVSEs

The specification describes that it is not allowed to set a ChargingProfile with a stackLevel - chargingProfilePurpose combination that already exists in another ChargingProfile on the Charging Station. However this should be allowed if it is set on a different EVSE.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	K01.FR.06		The CSMS SHALL NOT set a ChargingProfile with a stackLevel - chargingProfilePurpose combination that already exists in another ChargingProfile on the Charging Station.
New	K01.FR.06		The CSMS SHALL NOT set a ChargingProfile with a stackLevel - chargingProfilePurpose - **evseId** combination that already exists in another ChargingProfile on the Charging Station.

3.107.2. Requirement 'K01.FR.06' incomplete

Additional clarifications are added to requirement 'K01.FR.06'. ChargingProfilePurpose 'TxProfile', needs to be excluded from this requirement.

Changed requirement:

Version	Id	Precondition	Requirement definition	Note
Old	K01.FR.06		The CSMS SHALL NOT set a ChargingProfile with a stackLevel - chargingProfilePurpose combination that already exists in another ChargingProfile on the Charging Station.	Charging Profiles can be replaced using their chargingProfileId (K01.FR.05)

Version	Id	Precondition	Requirement definition	Note
New	K01.FR.06	**When chargingProfilePurpose is NOT TxProfile**	The CSMS SHALL NOT send a ChargingProfile with a stackLevel - chargingProfilePurpose - evseId combination that already exists in another ChargingProfile **(with different id)** on the Charging Station **and has an overlapping validity period**.	**This is to ensure that no two charging profiles with same stack level and purpose can be valid at the same time.**

3.108. (Minor) - Page 207, section 5.1, K01 SetChargingProfile, ValidFrom / ValidTo parameters unclear

Changed requirements:

Version	Id	Precondition	Requirement definition
Old	K01.FR.10		The Charging Station SHALL continue to execute the existing ChargingProfile until the new ChargingProfile is installed.
New	K01.FR.10	**When validFrom and validTo of a ChargingProfile are not set**	The Charging Station SHALL **consider the ChargingProfile to be valid indefinitely until it is explicitly replaced.**
Old	K01.FR.17		At any point in time, the prevailing charging profile for a certain charging purpose SHALL be the charging profile with the highest stackLevel among the profiles with the same charging purpose that are valid at that point in time, as determined by their validFrom and validTo parameters.
New	K01.FR.17		**When more than one ChargingProfile with the same chargingProfilePurpose is valid, as determined by their *validFrom* and *validTo* fields, then a ChargingSchedule from a ChargingProfile with a higher *stackLevel* overrules a ChargingSchedule from a ChargingProfile with a lower *stackLevel*.**

New requirements:

ID	Precondition	Requirement definition
K01.FR.36	When validFrom of a ChargingProfile is set	The Charging Station SHALL consider the ChargingProfile to be valid when current time >= validFrom.
K01.FR.37	When validTo of a ChargingProfile is set	The Charging Station SHALL consider the ChargingProfile to be valid when current time < validTo.

3.109. (Minor) - Page 207, Use case K01, Requirements regarding duration and ChargingSchedulePeriods are unclear

Changed requirements:

Version	Id	Precondition	Requirement definition
Old	K01.FR.11	If ChargingSchedulePeriod is longer than duration.	The remainder ChargingSchedulePeriod SHALL NOT be executed.
New	K01.FR.11	**If ChargingSchedule has a *duration* AND ChargingSchedulePeriod.*startPeriod* >= ChargingSchedule.*duration***	**The Charging Station SHALL not execute the ChargingSchedulePeriod, because it is past the duration of the ChargingSchedule.**
Old	K01.FR.12	If duration is longer than the ChargingSchedulePeriod	The Charging Station SHALL keep the value of the last ChargingSchedulePeriod until duration has ended.
New	K01.FR.12		**A ChargingSchedulePeriod remains active until the next ChargingSchedulePeriod in the list starts or until ChargingSchedule.*duration* has elapsed.**

3.110. (Minor) - Page 207, section 5.1, K01 SetChargingProfile, Response to invalid connectorId

The following requirement must be added to table 158:

ID	Precondition	Requirement definition
K01.FR.28	When a SetChargingProfileRequest is received for an evseId that does not exist.	Charging Station SHALL respond with SetChargingProfileResponse with status `Rejected`
K01.FR.29	When Charging Station does not support smart charging.	Charging Station SHALL respond with RPC Framework CALLERROR: NotSupported.

3.111. (Major) - Page 208, Use case K01, SetChargingProfile

Since SetChargingProfileRequest can contain up to three ChargingSchedules and SalesTariffs in order to support ISO 15118 sessions, a requirement needs to be added that only one ChargingSchedule may be provided when ISO 15118 is not being used.

New requirement:

ID	Precondition	Requirements	Note
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.

3.112. (Medium) - Page 208, Use case K01, Not specified that the elements must be ordered by startPeriod value

OCPP has always meant for the elements in the ChargingSchedule to be ordered based on the startPeriod (in chronological order). However the specification does not specifically state that this is the case. A requirement needs to be added for this.

New requirement:

ID	Precondition	Requirement definition
K01.FR.35		The list of ChargingSchedulePeriod elements in a chargingSchedule SHALL be ordered by increasing values of ChargingSchedulePeriod.*startPeriod*.

3.113. (Medium) - Page 208, Use case K01, It is unclear if ChargingProfiles set via SetChargingProfileRequest are persistent across reboots/power cycles

New requirement:

ID	Precondition	Requirement definition
K01.FR.27		ChargingProfiles set via SetChargingProfileRequest SHALL be persistent across reboots/power cycles.

3.114. (Minor) - Page 208, Use case K01, The Charging Station has no way to reject not supported ChargingRateUnits

New requirement:

ID	Precondition	Requirement definition
K01.FR.26	When a SetChargingProfileRequest is received with a value for chargingRateUnit, that is not configured in the configuration variable ChargingScheduleChargingRateUnit.	Charging Station SHALL respond with SetChargingProfileResponse with status Rejected.

3.115. (Minor) - Page 208, Use case K01, a *SetChargingProfileRequest* with 3 phases on a Charging Station which only supports 1 phase should be *Rejected*

Requirement K01.FR.18 is unclear on how the Charging Station should react.

Old requirement definition	For AC charging, the CSMS SHALL NOT set *numberPhases* different from the EVSE capabilities in a *SetChargingProfileRequest*.
New requirement definition	For AC charging, the CSMS SHALL NOT set *numberPhases* different from the EVSE capabilities in a *SetChargingProfileRequest*, **otherwise the Charging Station SHOULD respond with *Rejected***.

Also a note needs to be added to the requirement about the number of phases supported by the EV and/or cable .

Added requirement note 'K01.FR.18':

Note	When a ChargingProfile asks for 3 phases and the charging station is able to charge 3 phases, it is no guaranteed that the EV and/or cable are able to charge 3 phases. Based on *MeterValues* the CSMS can determine the phases used.

3.116. (Minor) - Page 208, Use case K01 and K10, Effect of TxDefaultProfile change on running transactions

The following changes are added to make explicit, that changing or clearing a TxDefaultProfile affects active transactions that are using a TxDefaultProfile.

The following requirement must be added to table 158:

ID	Precondition	Requirement definition
K01.FR.32	(K01.FR.14 OR K01.FR.15) AND a transacton is active on the specified EVSE(s) (evseId = 0 refers to all EVSEs.)	The Charging Station SHALL continue the transaction on the specified EVSE(s), but switch to using the new TxDefaultProfile.

The following requirement must be added to table 176:

ID	Precondition	Requirement definition
K10.FR.07	K10.FR.05 AND the cleared profile has chargingProfilePurpose = TxDefaultProfile	The Charging Station SHALL continue any active transaction, that started with a TxDefaultProfile, as if it was started without a TxDefaultProfile.

3.117. (Minor) - Page 208, Use case K01: missing requirement for chargingShedulePeriod

New requirement:

ID	Precondition	Requirement definition	Note
K01.FR.31		The **startPeriod** of the first chargingSchedulePeriod in a chargingSchedule SHALL always be 0.	

3.118. (Minor) - Page 208, Use case K01, Set Charging Profile

Add the following requirement to table 158

K01.FR.30	*chargingProfile* has a *chargingSchedule* with *startSchedule* set to a time in the future	The Charging Station SHALL only start imposing the limitation of this schedule as of point in time set by *startSchedule*

3.119. (Minor) - Page 208, Use case K01, Unclear what the Charging Station needs to do when receiving a SetChargingProfileRequest with an unknown transactionId

Added requirement:

ID	Precondition	Requirements
K01.FR.33	K01.FR.03 AND the given transactionId is not known	The Charging Station SHALL reject the SetChargingProfileRequest.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	K01.FR.04	K01.FR.03	The Charging Station SHALL apply the sent TxProfile to the transaction with the specified transactionId.
New	K01.FR.04	K01.FR.03 **AND** **the given transactionId is known**	The Charging Station SHALL apply the sent TxProfile to the transaction with the specified transactionId.

3.120. (Medium) - Page 208, Use case K01, Missing requirement for setting a chargingProfilePurpose TxProfile

It needs to be allowed to set a TxProfile with the same stackLevel for a different transaction.

Added requirement:

ID	Precondition	Requirement definition
K01.FR.39	When *chargingProfilePurpose* is `TxProfile`	The CSMS SHALL NOT send a ChargingProfile with a *stackLevel* that already exists in another ChargingProfile (with different *id*) with purpose `TxProfile` and the same *transactionId*.

It should not be allowed to replace a profile based on the stackLevel - chargingProfilePurpose - transactionId combination. Therefore requirement K05.FR.04 needs to be removed.

Removed requirement:

ID	Precondition	Requirement definition
K05.FR.04	When the Charging Station receives a SetChargingProfileRequest, with the transactionId for this transaction, with the same StackLevel as the profile given in the RequestStartTransactionRequest.	the Charging Station SHALL replace the existing charging profile, otherwise it SHALL install/stack the profile next to the already existing profile(s).

3.121. (Minor) - Page 208, Use case K01, The specification does not describe it is not allowed to set a Relative ChargingStationMaxProfile

It is not allowed to use chargingProfileKind *Relative* for a Charging Profile with chargingProfilePurpose *ChargingStationMaxProfile*.

Added requirement:

ID	Precondition	Requirement definition
K01.FR.38	When *chargingProfilePurpose* = `ChargingStationMaxProfile`	*chargingProfileKind* SHALL NOT be `Relative`

3.122. (Minor) - Page 222, Use case K08, Missing requirement about rejecting not supported chargingRateUnit

New requirement:

ID	Precondition	Requirement definition
K08.FR.07	When receiving a GetCompositeScheduleRequest with a chargingRateUnit, which is not configured in the configuration variable `ChargingScheduleChargingRateUnit`	The Charging Station SHALL respond with GetCompositeScheduleResponse(status=Rejected)

3.123. (Minor) - Page 222, Use case K08, It is unclear how to respond to a GetCompositeSchedule request when there is no transaction ongoing

New requirement:

ID	Precondition	Requirement definition
K08.FR.06	When there is no transaction ongoing	The Charging Station SHALL calculate the composite schedule as if there is a transaction ongoing.

3.124. (Minor) - Page 222, Use case K08, unclear what the composite schedule is reporting.

There is a lot of information about how the composite schedule is calculated, but it should also be made clear what the composite schedule is reporting.

Addition to remark(s) K08:

Remark	The composite schedule reports the expected power or current the Charging Station expects to consume from the grid, for the requested EVSE, during the requested time period. When requested for evseid=0, the Charging Station will calculate the total expected consumption for the grid connection.

3.125. (Minor) - Page 222, Use case K08, Requirement K08.FR.02, confusing requirement

The requirement is a bit confusing, so it should be rephrased.

Changed requirement:

Version	Id	Requirement definition
Old	K08.FR.02	The Charging Station SHALL calculate the Composite Charging Schedule intervals, from the moment the request PDU is received: Time X, up to X Duration, and send them in the GetCompositeScheduleResponse PDU to the CSMS.
New	K08.FR.02	The Charging Station SHALL calculate the **scheduled time intervals, from the moment of message receipt up to the Duration (in seconds) and send them** to the CSMS.

3.126. (Minor) - Page 223, Use case K09, Unclear whether the fields in GetChargingProfileRequest are matched as a logical AND or not

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	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 to specify which Charging Profiles need to be reported.
New	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.

3.127. (Minor) - Page 224, Use case K10, criteria fields are incorrect

This needs to be read **after** erratum (Major) - Page 299, ClearChargingProfileRequest, criteria fields incorrectly structured

The text of the following requirements needs to be changed.

Old text	K10.FR.03	Upon receipt of a ClearChargingProfileRequest with a specified chargingProfile.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 **id**	The Charging Station SHALL clear the Charging Profile with the matching id and respond with a ClearChargingProfileResponse message
Old text	K10.FR.04	Upon receipt of a ClearChargingProfileRequest, with values for stackLevel, evseId and/or chargingProfilePurpose	The Charging Station SHALL clear the Charging Profiles that match the values in the request and respond with a ClearChargingProfileResponse message.
New 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.

3.128. (Major) - Page 231, 5.3 ISO 15118 based Smart Charging

For ISO 15118 special use cases K15-K17 have been defined, that use messages and types that differ from the other smart charging use cases. This is confusing and introduces more complexity in both CS and CSMS and also creates a dependency between OCPP and ISO 15118, which is unnecessary.

Instead of using SetChargingProfileRequest with a ChargingSchedule, we use NotifyCentralChargingNeedsRequest with an SASchedule – a different format. It may appear to be easier on the CS to use the ISO 15118 SASchedule directly in this message, but it is not. Because when the EV returns its own calculated schedule, then CS must report this in NotifyEVChargingScheduleRequest in the format of a ChargingSchedule (and not SASchedule). Also, when the CS needs to return its CompositeSchedule in a GetCompositeScheduleResponse, then it needs to take the SASchedule into account and report the CompositeSchedule in a ChargingSchedule format.

It is confusing that in order to change a charging profile, for which a CSMS will normally use SetChargingProfileRequest, it now has to send a Renegotiate15118ScheduleRequest, receive an NotifyEVChargingNeeds (which should not differ from the original, because renegotiation was initiated by CSMS) and then send a NotifyCentralChargingNeedsRequest.

Therefore we changed the use cases K15-K17 to use SetChargingProfileRequest and we updated ChargingScheduleType with an optional element *salesTariff*, so that it can provide the same information as a NotifyCentralChargingNeedsRequest message.

3.128.1. K15 - Charging with load leveling based on High Level Communication

In use case K15 change the description of row 6 in Table 185 as follows:

Table 185. K15 - Charging with load leveling based on High Level Communication

No.	Type	Description
5	**Actors**	**EV**, Charging Station, **CSMS**.
6	**Combined scenario description**	**1.** The EV sends a ChargeParameterDiscoveryReq message to the Charging Station. **2.** The Charging Station sends a NotifyEVChargingNeedsRequest message to the CSMS. **3.** The CSMS sends a NotifyEVChargingNeedsResponse message to the Charging Station. **4. The CSMS sends a SetChargingProfileRequest message to the Charging Station.** **5. The Charging Station sends a SetChargingProfileResponse message to the CSMS.** **6.** The Charging Station responds to the EV with a ChargeParameterDiscoveryRes message to the EV. **7.** The EV sends a PowerDeliveryReq message to the Charging Station with ChargeProgress=Start. This marks the point in time when the EVSE provides voltage to its output power outlet and the EV can start to recharge its battery. **8.** The contactor is closed. **9.** A PowerdeliveryRes message is sent to the EV. **10.** Optionally, the Charging Station sends a NotifyEVChargingScheduleRequest message to the CSMS. **11.** The transaction is updated with a TransactionEventRequest message.

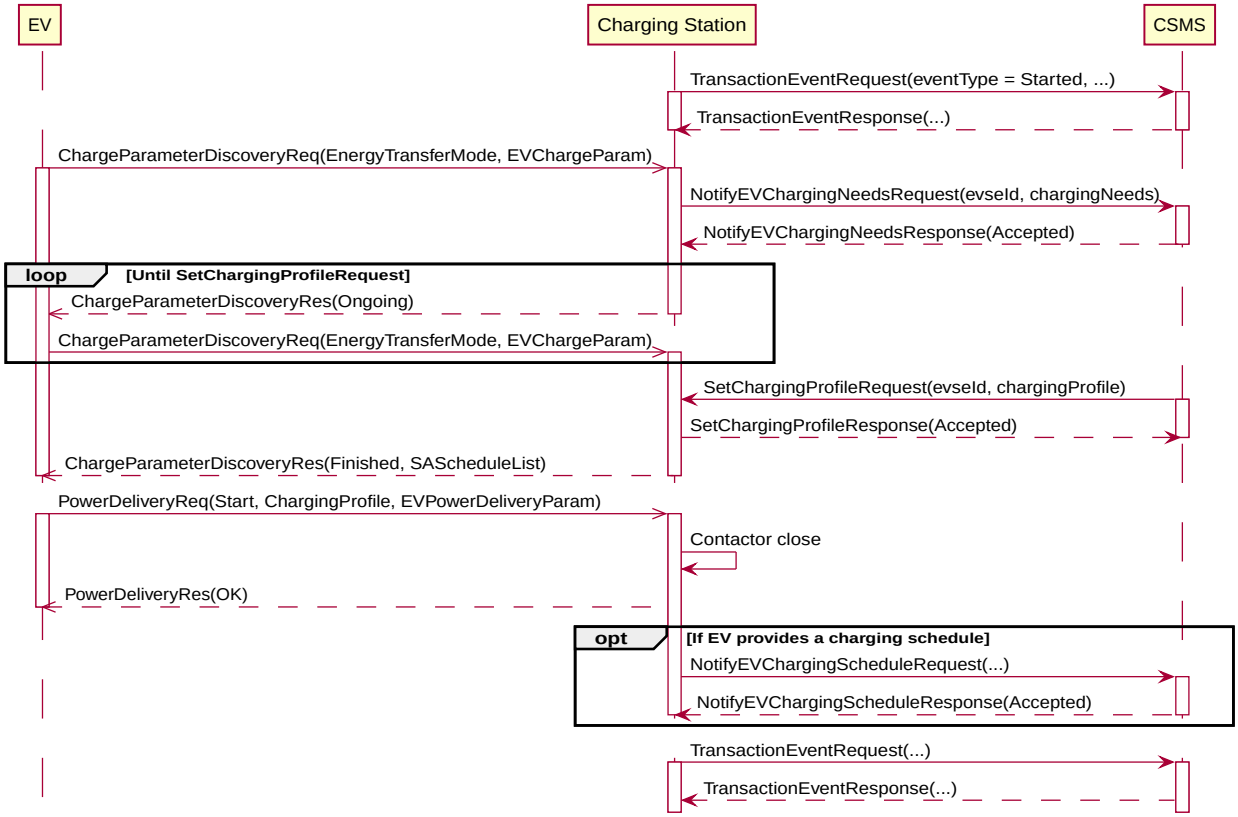


Figure 1. Sequence Diagram: Charging with load leveling based on High Level Communication

The following table **replaces** the requirements of use case K15:

Table 186. K15 - Requirements

ID	Precondition	Requirements	Note
K15.FR.01	When the Charging Station receives charging needs from the EV	The Charging Station SHALL send a NotifyEVChargingNeedsRequest to the CSMS.	
K15.FR.02	K15.FR.01	In response to a NotifyEVChargingNeedsRequest the CSMS SHALL send a NotifyEVChargingNeedsResponse.	

ID	Precondition	Requirements	Note
K15.FR.03	K15.FR.02	If the CSMS is able to provide a charging schedule, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to 'Accepted'.	
K15.FR.04	K15.FR.02	If the CSMS is not able to provide a charging schedule, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to 'Rejected'.	
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'.	
K15.FR.06		A NotifyEVChargingNeedsRequest SHALL contain either ACChargingParameters or DCChargingParameters.	
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`.	
K15.FR.08	K15.FR.01	The CSMS SHOULD send a SetChargingProfileRequest to the Charging Station within 60 seconds.	This is to satisfy the ISO 15118 ChargeParameterDiscoveryReq timeout.
K15.FR.09	K15.FR.07 AND EV returns a charging profile	Charging Station SHALL verify that provided charging profile is within boundaries of the ChargingSchedule from CSMS.	In ISO 15118 EV can sent its charging profile as part of PowerDeliveryReq.
K15.FR.10	K15.FR.09	Charging Station SHALL send the EV charging profile in a NotifyEVChargingScheduleRequest message to CSMS.	
K15.FR.11	K15.FR.10 AND EV charging profile is within limits of CSMS ChargingSchedule	CSMS responds with NotifyEVChargingScheduleResponse with *status* `Accepted` to Charging Station.	Note: Already checked by Charging Station, but CSMS does its own check.
K15.FR.12	K15.FR.10 AND EV charging profile is NOT within limits of CSMS ChargingSchedule	CSMS responds with NotifyEVChargingScheduleResponse with *status* `Rejected` to Charging Station.	
K15.FR.13	K15.FR.12	CSMS starts new renegotiation as per use case K16.	
K15.FR.14	K15.FR.11	The Charging Station SHOULD take the schedule from the NotifyEVChargingScheduleRequest into account when calculating the actual Composite schedule.	
K15.FR.15	K15.FR.03 AND Charging Station is offline	The Charging Station SHALL use the TxDefaultProfile (if present) and generate a charging schedule within the limits of its composite schedule.	

ID	Precondition	Requirements	Note
K15.FR.16	K15.FR.07	It is RECOMMENDED to configure the Charging Station, such that a TransactionEvent with idToken has been sent prior to the NotifyEVChargingNeedsRequest Message, so that CSMS can take the user into account when creating a charging schedule.	
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.

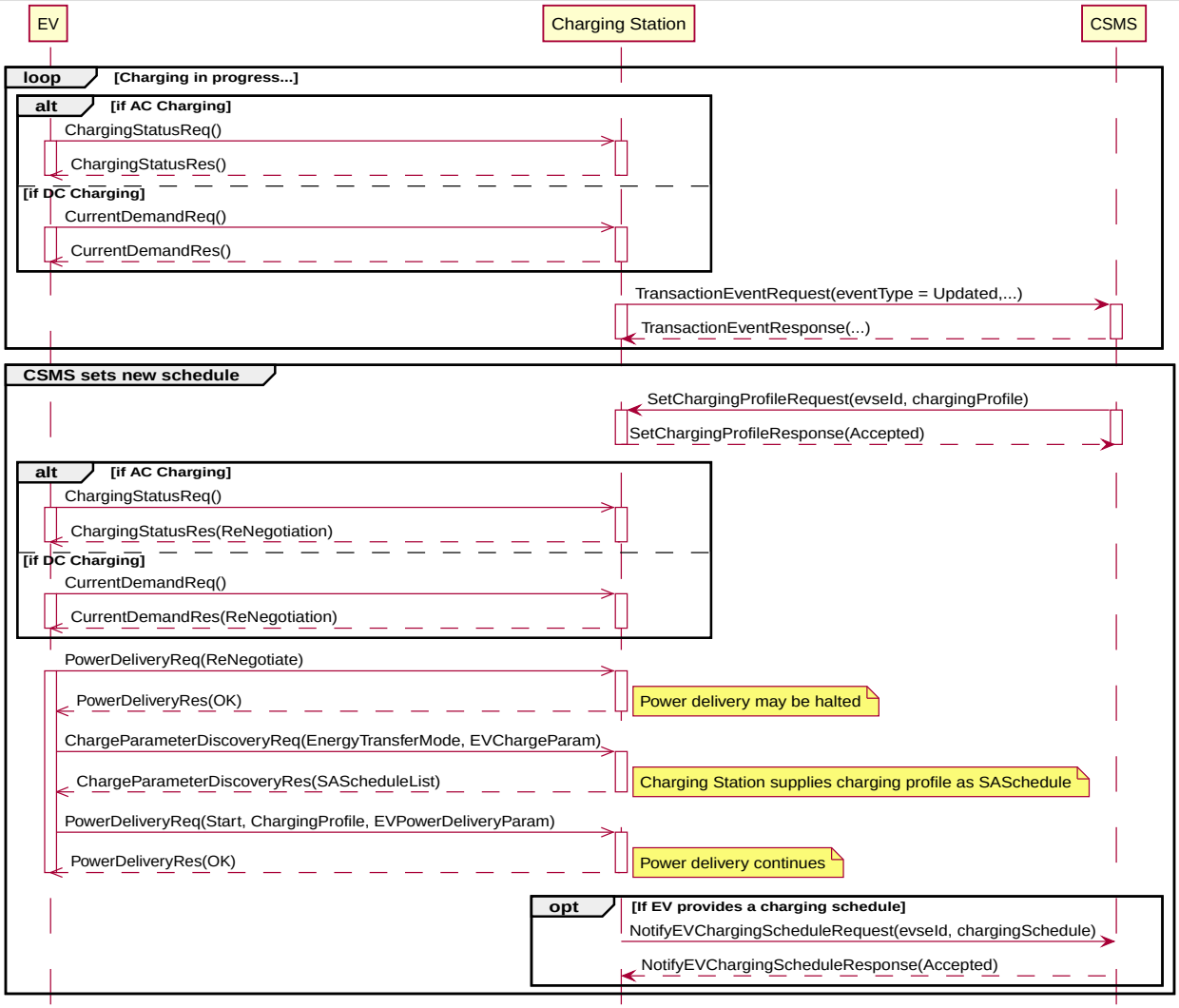
3.128.2. K16 - Renegotiation initiated by CSMS

The original use case was not providing much added value compared to K15. We therefore rewrote it to specify the behavior of the CSMS providing a new charging schedule during the charging session. This is called renegotiation in ISO 15118.

Table 2. K16 - Renegotiation initiated by CSMS

No.	Type	Description
1	**Name**	Renegotiation initiated by CSMS.
2	**ID**	K16
	Functional block	K. Smart Charging
3	**Objectives**	To control the charging power or current of a Charging Station
4	**Description**	The CSMS sends a SetChargingProfileRequest to the Charging Station to influence the power or current drawn by the EV. The CSMS calculates a ChargingSchedule to stay within limits which MAY be imposed by an external system. **Note:** Description of actions between EV and Charging Station is informative only and not mandated by OCPP.
	Actors	EV, Charging Station, CSMS
	Scenario description	**1** CSMS sends a SetChargingProfileRequest to the Charging Station. **2** Charging Station responds with a SetChargingProfileResponse to the CSMS. **3** When EV sends the next CurrentDemandReq (for DC) or ChargingStatusReq (for AC), the Charging Station will respond with *evseNotification* = `ReNegotiation`. **4** EV sends a PowerDeliveryReq with *chargeProgress* = `ReNegotiate` to confirm this. **5** Charging Station responds with a PowerDeliveryRes. **6** EV sends a ChargeParameterDiscoveryReq. **7** Charging Station responds with a ChargeParameterDiscoveryRes with an SAScheduleList that contains the ChargingSchedule data from the SetChargingProfileRequest. **8** EV sends a PowerDeliveryReq with *chargeProgress* = `Start` (with an optional charging profile) to confirm this. **9** Charging Station responds with PowerDeliveryRes and, if charging was suspended at start of the renegotiation, will resume power delivery. **10** If EV provided a charging profile in the previous step, then Charging Station will send a NotifyEVChargingScheduleRequest to the CSMS.
5	**Prerequisites**	Charging session started according to use case K15.
6	**Postcondition(s)**	Charging session uses the new charging profile.

Figure 113. Renegotiation initiated by CSMS



7	**Remark(s)**	Signed SalesTariffs are currently not supported. If these are needed please use P01 - Data Transfer to the Charging Station to send these to the Charging Station.

K16 - Renegotiation initiated by CSMS - Requirements

Since use case K16 is completely rewritten, the requirements for use case K16 are all replaced by the following:

ID	Precondition	Requirements	NOTE
K16.FR.01	CSMS sends a new SetChargingProfileRequest	Charging Station SHALL respond with a SetChargingProfileResponse with *status* = `Accepted`.	
K16.FR.02	K16.FR.01	Charging Station SHALL initiate schedule renegotiation with EV.	In ISO 15118 this is done by replying with EVSENotification=`ReNegotiation` to a CurrentDemandReq (for DC) or ChargingStatusReq (for AC) message.
K16.FR.03	K16.FR.02	Charging Station SHALL provide the ChargingSchedule data to the EV.	In ISO 15118 this is done in the ChargeParameterDiscoverRes message.
K16.FR.04	EV returns a charging profile	Charging Station SHALL verify that provided charging profile is within boundaries of the ChargingSchedule from CSMS.	In ISO 15118 EV may provide this as part of the PowerDeliveryReq message.
K16.FR.05	K16.FR.04	Charging Station SHALL send the EV charging profile in a NotifyEVChargingScheduleRequest message to CSMS.	
K16.FR.06	K16.FR.05 AND EV charging profile is within limits of CSMS ChargingSchedule	CSMS responds with NotifyEVChargingScheduleResponse with *status* `Accepted` to Charging Station.	Note: Already checked by Charging Station, but CSMS does its own check.

ID	Precondition	Requirements	NOTE
K16.FR.07	K16.FR.05 AND EV charging profile is NOT within limits of CSMS ChargingSchedule	CSMS responds with NotifyEVChargingScheduleResponse with *status* `Rejected` to Charging Station.	
K16.FR.08	K16.FR.07	CSMS starts new renegotiation as per use case K16.	
K16.FR.09	When the Charging Station receives charging needs from the EV	The Charging Station SHOULD NOT send a NotifyEVChargingNeedsRequest to the CSMS.	CSMS initiated the renegotiation and has just sent a new charging profile, based on the initial charging needs from EV, energy already consumed by EV and whatever information has caused CSMS to update the charging profile. In ISO 15118 charging needs are sent via ChargeParameter-DiscoveryReq.
K16.FR.10	K16.FR.04	The Charging Station SHOULD take the schedule from the NotifyEVChargingScheduleRequest into account when calculating the actual Composite schedule.	
K16.FR.11	K16.FR.02 AND current or power in new charging schedule is lower than actual current or power	The Charging Station SHALL request EV to lower current or power to a value matching the new charging schedule at the first possible opportunity.	In ISO 15118 this can be communicated in CurrentDemandRes (for DC) or ChargingStatusRes (for AC).
K16.FR.12	K16.FR.09 AND Charging Station sends a NotifyEVChargingScheduleRequest	The CSMS SHALL send a SetChargingProfileRequest.	This situation is not desirable, because charging profile will likely be the same as in K16.FR.01, but this is added for robustness when Charging Station is not adhering to K16.FR.09.

3.128.3. K17 - Renegotiation initiated by EV

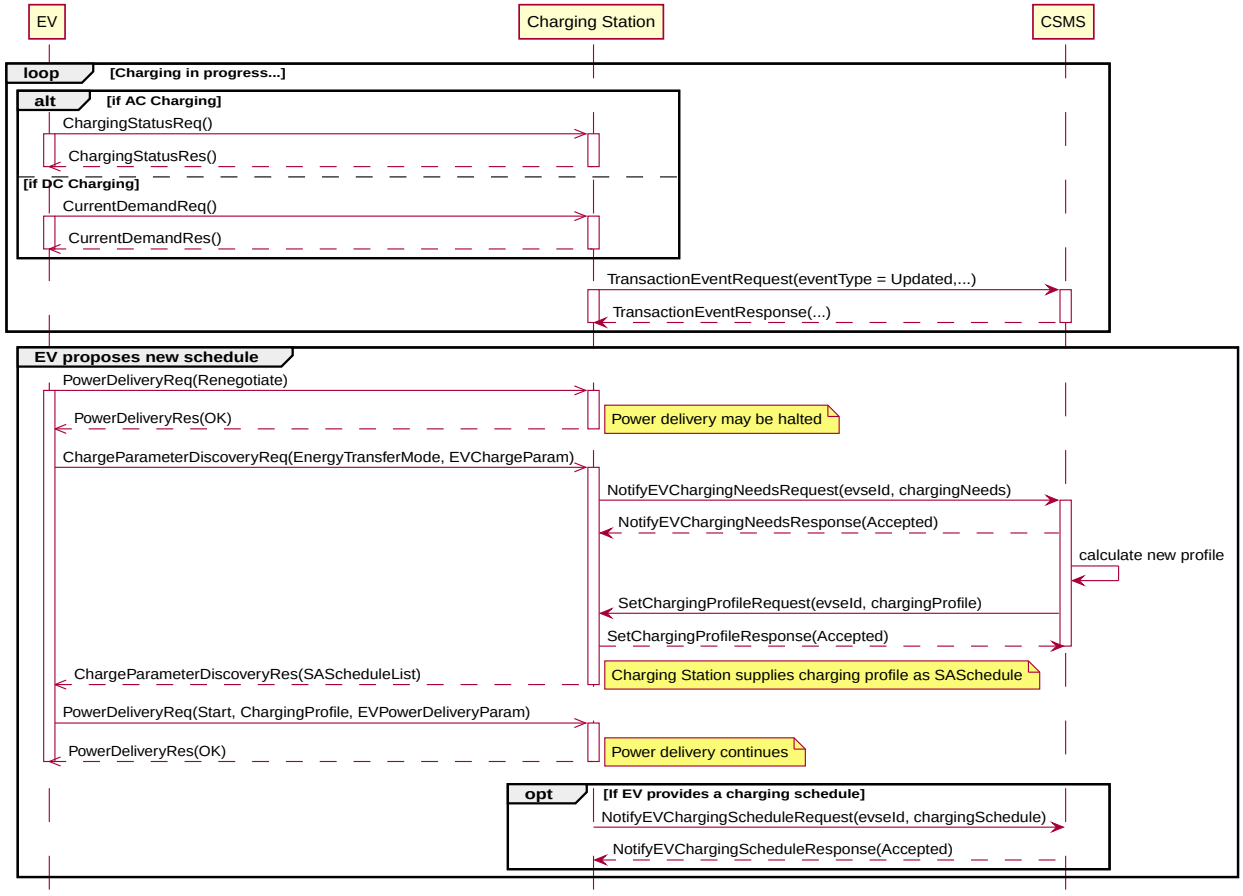
The original use case has been rewritten to cover the renegotiation of a new charging schedule by the EV.

Table 3. K17 - Renegotiation initiated by EV

No.	Type	Description
1	**Name**	Renegotiation initiated by EV.
2	**ID**	K16
	Functional block	K. Smart Charging
3	**Objectives**	To let an EV request a new charging schedule.
4	**Description**	The EV signals the Charging Station that it wants to renegotiate and it provides new charging needs, which the Charging Station sends to the CSMS. Based on this and other parameters, the CSMS calculates a new charging schedule and sends it via SetChargingProfileRequest to Charging Station, which communicates it to the EV. **Note:** Description of actions between EV and Charging Station is informative only and not mandated by OCPP.
	Actors	EV, Charging Station, CSMS

No.	Type	Description
	Scenario description	**1** When EV sends a ChargeParameterDiscoveryReq with with charging needs parameters, then Charging Station sends this information in a NotifyEVChargingNeedsRequest to CSMS. **2** CSMS responds with NotifyEVChargingNeedsResponse to Charging Station. **3** CSMS calculates new charging schedule, that tries to accomodate the EV charging needs and still fits within the schedule boundaries imposed by other parameters. **4** CSMS sends a SetChargingProfileRequest with the new schedule to the Charging Station. **5** Charging Station responds with SetChargingProfileResponse with status `Accepted`. **6** Charging Station sends new charging schedule to EV in a ChargeParameterDiscoveryRes message. **7** EV sends a PowerDeliveryReq with *chargeProgress* = `Start` (with an optional charging profile) to confirm this. **8** Charging Station responds with PowerDeliveryRes and, if charging was suspended at start of the renegotiation, will resume power delivery. **9** If EV provided a charging profile in the previous step, then Charging Station will send a NotifyEVChargingScheduleRequest to the CSMS.
5	**Prerequisites**	Charging session started according to use case K15.
6	**Postcondition(s)**	Charging session uses the new charging profile.

Figure 114. Renegotiation initiated by EV



7	**Remark(s)**	Signed SalesTariffs are currently not supported. If these are needed please use P01 - Data Transfer to the Charging Station to send these to the Charging Station.

K17 - Renegotiation initiated by EV - Requirements

Since use case K17 is completely rewritten, the requirements for use case K17 are all replaced by the following:

ID	Precondition	Requirements	Note
K17.FR.01	EV triggers a renegotiation and sends new charging needs	The Charging Station SHALL send a NotifyEVChargingNeedsRequest to the CSMS.	
K17.FR.02	K17.FR.01	In response to a NotifyEVChargingNeedsRequest the CSMS SHALL send a NotifyEVChargingNeedsResponse.	
K17.FR.03	K17.FR.02	If the CSMS is able to provide a charging schedule, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to 'Accepted'.	
K17.FR.04	K17.FR.02	If the CSMS is not able to provide a charging schedule, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to 'Rejected'.	
K17.FR.05	K17.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'.	
K17.FR.06		A NotifyEVChargingNeedsRequest SHALL contain either ACChargingParameters or DCChargingParameters.	
K17.FR.07	K17.FR.03 or K17.FR.05	The CSMS SHALL send a SetChargingProfileRequest with *chargingProfilePurpose* = `TxProfile` 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.	
K17.FR.08	K17.FR.01	The CSMS SHOULD send a SetChargingProfileRequest to the Charging Station within 60 seconds.	This is to satisfy the ISO 15118 ChargeParameterDiscoveryReq timeout.
K17.FR.09	K17.FR.07 AND EV returns a charging profile	Charging Station SHALL verify that provided charging profile is within boundaries of the ChargingSchedule from CSMS.	In ISO 15118 EV can sent its charging profile as part of PowerDeliveryReq.
K17.FR.10	K17.FR.09	Charging Station SHALL send the EV charging profile in a NotifyEVChargingScheduleRequest message to CSMS.	
K17.FR.11	K17.FR.10 AND EV charging profile is within limits of CSMS ChargingSchedule	CSMS responds with NotifyEVChargingScheduleResponse with *status* `Accepted` to Charging Station.	Note: Already checked by Charging Station, but CSMS does its own check.
K17.FR.12	K17.FR.10 AND EV charging profile is NOT within limits of CSMS ChargingSchedule	CSMS responds with NotifyEVChargingScheduleResponse with *status* `Rejected` to Charging Station.	
K17.FR.13	K17.FR.12	CSMS starts new renegotiation as per use case K16.	

ID	Precondition	Requirements	Note
K17.FR.14	K17.FR.11	The Charging Station SHOULD take the schedule from the NotifyEVChargingScheduleRequest into account when calculating the actual Composite schedule.	
K17.FR.15	K17.FR.01 AND Charging Station is offline	The Charging Station SHALL use the TxDefaultProfile (if present) and generate a charging schedule within the limits of its composite schedule.	

3.128.4. Page 311, Messages 1.33, NotifyCentralChargingNeeds now obsolete

NotifyCentralChargingNeedsRequest and -Response are no longer needed. The revelant information is now provided by SetChargingProfileRequest.

3.128.5. Page 315, Messages 1.44, Renegotiate15118Schedule now obsolete

The specific message for ISO 15118 to trigger renegotiation has become obsolete as this is handled by the Charging Station when a SetChargingProfile is received while an ISO 15118 session is in progress.

3.128.6. Page 331, Datatypes 1.10, ChargingProfileType

The type definition for ChargingProfileType has been adapted to support up to three ChargingSchedules in order to support ISO 15118 schedule negotiation, which supports three schedules.

The attribute primary has been removed.

ChargingProfileType

Field Name	Field Type	Card.	Change
primary	boolean	0..1	Removed
chargingSchedule	ChargingScheduleType	**1..3**	Changed cardinality to 1..3

3.128.7. Page 332, Datatype 1.12, ChargingScheduleType

In order to use a ChargingScheduleType with ISO 15118, the following changes have been made:

- The cardinality of *chargingSchedulePeriod* has been changed from 1..* to 1..1024.
- SalesTariffType from SAScheduleType has been added as an optional attribute *salesTariff*.
- An id has been added, because it is possible to have up to three charging schedules in a charging profile.

ChargingScheduleType

Field Name	Field Type	Card.	Change
chargingSchedulePeriod	ChargingSchedulePeriodType	**0..1024**	Cardinality changed from 1..* to 1..1024. Note: the maximum number of periods, that is supported by the Charging Station is set by device model variable SmartChargingCtrlr.PeriodsPerSchedule.
salesTariff	SalesTariffType	0..1	Added
id	integer	1..1	Added

3.128.8. Page 343, Datatypes 145, SAScheduleType now obsolete

SAScheduleType is no longer used. The relevant information is now in ChargingScheduleType.

3.129. (Minor) - Page 241, Use Cases L01 & L02 & N01, Interoperability issues may occur for firmware update and upload log

Addition to remark(s) L01 & L02 & N01:

Remark	FTP needs to be able to use Passive FTP, to be able to transverse over as much different typologies as possible.

3.130. (Minor) - page 241, Use case L01, Charging Station may be unable to send FirmwareStatusNotification(status=Installing) when it needs to reboot during firmware update

It may be possible the Charging Station is unable to send FirmwareStatusNotification(status=Installing), when it needs to reboot during the firmware update process and the bootloader is not able to send OCPP messages.

New requirement:

ID	Precondition	Requirement definition
L01.FR.24	When the Charging Station needs to reboot during a firmware update AND the bootloader is unable to send OCPP messages	The Charging Station MAY omit the FirmwareStatusNotification(status=Installing) message.
L02.FR.16	When the Charging Station needs to reboot during a firmware update AND the bootloader is unable to send OCPP messages	The Charging Station MAY omit the FirmwareStatusNotification(status=Installing) message.

3.131. (Minor) - page 241, Use case L01, The certificate provided in the UpdateFirmwareRequest can immediately be validated when receiving the message

The Charging Station needs to be able to respond to the UpdateFirmwareRequest with an InvalidCertificate status. This also means that there is no need for the CertificateVerified FirmwareStatusNotification message.

Added Enum values:

Page	section	Message/dataType	Value	Description
374	2.84	UpdateFirmwareStatusEnumType	InvalidCertificate	Failure end state. The Firmware Signing certificate is invalid.
374	2.84	UpdateFirmwareStatusEnumType	RevokedCertificate	Failure end state. The Firmware Signing certificate has been revoked.

Removed Enum values:

Page	section	Message/dataType	Value
360	2.34	FirmwareStatusEnumType	CertificateVerified
360	2.34	FirmwareStatusEnumType	InvalidCertificate
360	2.34	FirmwareStatusEnumType	RevokedCertificate

New Schema files are available.

New requirements:

ID	Precondition	Requirement definition
L01.FR.21	When the Charging Station receives an UpdateFirmwareRequest	The Charging Station SHALL validate the certificate before accepting the message.
L01.FR.22	L01.FR.21 AND the certificate is invalid	The Charging Station SHALL respond with UpdateFirmwareResponse(status=InvalidCertificate).

ID	Precondition	Requirement definition
L01.FR.23	L01.FR.21 AND the certificate is revoked	The Charging Station SHALL respond with UpdateFirmwareResponse(status=RevokedCertificate).

3.132. (Minor) - page 241, Use case L01, There is no way to provide intermediate certificates for the firmware signing certificate to the Charging Station

There is no immediate need to include this functionality. So the use of intermediate certificates for the firmware signing certificate will be prohibited for now. This may be added in a future version, if there is a need for it.

Addition to remark(s) L01:

Remark	The manufacturer SHALL NOT use intermediate certificates for the firmware signing certificate in the Charging Station.

3.133. (Minor) - page 241, Use case L01, PublishFailed is never used in FirmwareStatusNotificationRequest

The enum value PublishFailed belongs to PublishFirmwareStatusNotification, not FirmwareStatusNotificationRequest. So PublishFailed needs to be removed from FirmwareStatusNotificationRequest.

Removed Enum value:

Page	section	Message/dataType	Value
360	2.34	FirmwareStatusEnumType	PublishFailed

New Schema files are available.

3.134. (Minor) - page 243, requirement L01.FR.07: May cause EVSE to be set InOperative for a long time

L01.FR.07: May cause EVSE to be set InOperative for a long time, thereby blocking use of the Charging Station. This may not be a desirable situation for some CSO's. So it is needed to add a configuration variable to control the Charging Station's behavior in this situation.

New configuration variable 'AllowNewSessionsPendingFirmwareUpdate':

Required	no		
Component	**componentName**	ChargingStation	
Variable	**variableName**	AllowNewSessionsPendingFirmwareUpdate	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	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. When a firmware update is waiting to be installed and this variable exists and has the value *true*, then, the Charging Station will not set free EVSEs to Unavailable, pending the update. This means that it may take longer until there is a point in time when all EVSEs of the Charging Station are free and it can perform the firmware update.		

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	L01.FR.07	L01.FR.06	The Charging Station SHALL set all connectors that are not in use to UNAVAILABLE while the Charging Station waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.
New	L01.FR.07	L01.FR.06 **AND** **configuration variable** `AllowNewSessionsPendingFirmwareUpdate` **is *false* or does not exist**	The Charging Station SHALL set all connectors that are not in use to UNAVAILABLE while the Charging Station waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.
Old	L02.FR.04	L02.FR.03	The Charging Station SHALL set all connectors that are not in use to UNAVAILABLE while the Charging Station waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.
New	L02.FR.04	L02.FR.03 **AND** **configuration variable** `AllowNewSessionsPendingFirmwareUpdate` **is *false* or does not exist**	The Charging Station SHALL set all connectors that are not in use to UNAVAILABLE while the Charging Station waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.

3.135. (Minor) - Page 244, Use Case L01 & L02, Missing requirements for status AcceptedCanceled

New requirements:

ID	Precondition	Requirement definition	Note
L01.FR.24	When a Charging Station is installing new Firmware OR is going to install new Firmware, but has received an UpdateFirmware command to install it at a later time AND the Charging Station receives a new UpdateFirmwareRequest	The Charging Station SHOULD cancel the ongoing firmware update AND respond with status *AcceptedCanceled*.	The Charging Station SHOULD NOT first check if the new firmware file exists, this way the CSMS will be able to cancel an ongoing firmware update without starting a new one.
L02.FR.15	When a Charging Station is installing new Firmware OR is going to install new Firmware, but has received an UpdateFirmware command to install it at a later time AND the Charging Station receives a new UpdateFirmwareRequest	The Charging Station SHOULD cancel the ongoing firmware update AND respond with status *AcceptedCanceled*.	The Charging Station SHOULD NOT first check if the new firmware file exists, this way the CSMS will be able to cancel an ongoing firmware update without starting a new one.

3.136. (Minor) - Page 244, Missing requirement stating that the after the installation the new firmware should be activated

New requirement:

ID	Precondition	Requirement definition
L01.FR.28	After Charging Station has sent FirmwareStatusNotificationRequest with *status* = Installed	Charging Station SHOULD have activated the new firmware or do so immediately. This MAY involve an automatic reboot, but not necessarily so.

3.137. (Minor) - Page 254, Use case M, 2.1 ISO15118 Certificate structure

The following sentence needs to be changed in section 2.1 ISO15118 Certificate structure:

Old	The CertificateUpdateRes and CertificateInstallationRes need to be sent from the CSO backend to the charging station as Base64 encoded binary data which is then directly forwarded from the Charging Station to the EV without further processing.
New	The CertificateUpdateRes and CertificateInstallationRes need to be sent from the CSO backend to the charging station as Base64 encoded binary data. The Charging Station removes the Base64 encoding and sends it to the EV as a binary EXI message.

3.138. (Minor) - Page 256, Use Case L02, Missing requirements from Use case L01

A lot of requirements from Use Case L01 are also applicable for L02. So they need to be copied from L01 to L02.

New requirements:

ID	Link ID
L02.FR.02	L01.FR.05
L02.FR.03	L01.FR.06
L02.FR.04	L01.FR.07
L02.FR.05	L01.FR.08
L02.FR.06	L01.FR.10
L02.FR.07	L01.FR.13
L02.FR.08	L01.FR.14
L02.FR.09	L01.FR.15
L02.FR.10	L01.FR.16

3.139. (Major) - Page 258, Use case M01 & M02, Replace Update15118EVCertificate by Get15118EVCertificate with option Install or Update

3.139.1. Page 258, Use Case M02, Scenario description needs to be updated

New Scenario description	15118: See ISO15118-1, use case Objective C1, Scenario Description, first 3 bullets, page 21. OCPP: - The Charging Station sends a Get15118EVCertificateRequest to the CSMS. - The CSMS responds with Get15118EVCertificateResponse to the Charging Station. 15118: See ISO15118-1, use case Description C1, Scenario Description, last 2 bullets, page 21.
New Scenario description	15118: See ISO15118-1, use case Objective C1, Scenario Description, first 3 bullets, page 21. OCPP: - The Charging Station sends a Get15118EVCertificateRequest **message with *action* = `Install`** to the CSMS. - The CSMS responds with Get15118EVCertificateResponse to the Charging Station. 15118: See ISO15118-1, use case Description C1, Scenario Description, last 2 bullets, page 21.

3.139.2. Page 258, M01

Change the following requirement:

Table 201. M01 - Requirements

ID	Precondition	Requirement definition	Note
M01.FR.01	Upon receiving a 15118 CertificateInstallationReq	The Charging Station SHALL forward the request to the CSMS using the Get15118EVCertificateRequest message **with *action* = `Install`**.	The CSMS is responsible for forwarding it to the 15118 contract certificate pool.

3.139.3. Page 259, Use Case M02, Scenario description needs to be updated

Old Scenario description	15118: See ISO15118-1, use case Objective C1, Scenario Description, first 3 bullets, page 21. OCPP: - The Charging Station sends a Update15118EVCertificateRequest to the CSMS. - The CSMS responds with Update15118EVCertificateResponse to the Charging Station. 15118: See ISO15118-1, use case Description C1, Scenario Description, last 2 bullets, page 21.
New Scenario description	15118: See ISO15118-1, use case Objective C1, Scenario Description, first 3 bullets, page 21. OCPP: - The Charging Station sends a **Get15118EVCertificateRequest message with *action* = `Update`** to the CSMS. - The CSMS responds with **Get15118EVCertificateResponse** to the Charging Station. 15118: See ISO15118-1, use case Description C1, Scenario Description, last 2 bullets, page 21.

3.139.4. Page 259, M02

Change the following requirement: *Table 203. M02 - Requirements*

ID	Precondition	Requirement definition	Note
M02.FR.01		Upon receiving a CertificateUpdateReq the Charging Station SHALL forward the request to the CSMS using the Get15118EVCertificateRequest message **with *action* = `Update`**.	The CSMS is responsible for forwarding it to the 15118 contract certificate pool.

3.139.5. Page 303, 1.16.1, Get15118EVCertificateRequest

Add field *action* to the message Get15118EVCertificateRequest as follows:

Field Name	Field Type	Card.	Description
action	CertificateActionEnumType	1..1	Required. Defines whether certificate needs to be installed or updated.

3.139.6. Page 326, 1.66, Remove Update15118EVCertificate message

Remove from section 1.66 the message Update15118EVCertificate. This has been replaced by Get15118EVCertificate with *action* = `Update`.

3.139.7. Page 352, after 2.6, add CertificateActionEnumType

Add a new enumeration: CertificateActionEnumType

Action in Get15118EVCertificateRequest.

Value	Description
Install	Install the requested certificate in the Charging Station.
Update	Update the requested certificate in the Charging Station.

3.140. (Medium) - Page 261, Use Case M05, there is no maximum amount of installed certificates defined

It must be possible to limit the maximum amount of certificates that a CSMS can install in a Charging Station.

New Configuration Variable 'CertificateEntries':

Required	yes		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	CertificateEntries	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	Maximum number of Certificates installed at any time.
Description	Amount of Certificates currently installed on the Charging Station.		

New requirement:

ID	Precondition	Requirement definition
M05.FR.06	When a new certificate gets installed AND the CertificateEntries.maxLimit is going to be exceeded	The Charging Station SHALL respond with status *Rejected*.

3.141. (Medium) - Page 261 & 277 & 278, The CSMS is able to use a different hash algorithm, than used while installing a certificate

When installing a certificate the CSMS uses a certain hash algorithm. The Charging Station will use this hash algorithm to internally calculate the certificate hashes. But when the CSMS would use a different hash algorithm when sending a DeleteCertificateRequest or CustomerInformationRequest, then the Charging Station has to recalculate the certificate hashes.

New requirements:

ID	Precondition	Requirement definition	Note
M04.FR.07	When deleting a certificate	The CSMS SHALL use the *hashAlgorithm*, which was used to install the certificate.	When a new firmware is installed it is RECOMMENDED that the CSMS requests the certificate first using GetInstalledCertificateIdsRequest to be sure of the used *hashAlgorithm*.
N09.FR.08	When requesting user information according to the *customerCertificate*	The CSMS SHALL use the *hashAlgorithm*, which was used to install the certificate.	When a new firmware is installed it is RECOMMENDED that the CSMS requests the certificate first using GetInstalledCertificateIdsRequest to be sure of the used *hashAlgorithm*.
N10.FR.09	When clearing user information according to the *customerCertificate*	The CSMS SHALL use the *hashAlgorithm*, which was used to install the certificate.	When a new firmware is installed it is RECOMMENDED that the CSMS requests the certificate first using GetInstalledCertificateIdsRequest to be sure of the used *hashAlgorithm*.

3.142. (Minor) - Page 261, Use Case M04, DeleteCertificate

There are no remarks or requirements about deleting the last CSMSRootCertificates. When the Charging Station is using security profile 2 or 3 and has no root certificate, it cannot verify the CSMSCertificate for the TLS connection and thus can never connect anymore.

New Remarks:

It is possible to delete the last (every) installed CSMSRootCertificates. When all CSMSRootCertificates are deleted, the Charging Station cannot validate CSMS Certificates, so it will not be able to connect to a CSMS. Before a CSMS would ever send a DeleteCertificateRequest that would delete the last/all CSMSRootCertificates the CSMS is ADVISED to make very sure that this is what is really wanted. It is possible to delete the last (every) installed ManufacturerRootCertificates, when all ManufacturerRootCertificates are deleted, no "Signed Firmware" can be installed in the Charging Station.

3.143. (Minor) - Page 261, Use case M05, Incorrect certificates are mentioned in the description

Old text	The CSMS requests the Charging Station to install a new root CA certificate, Sub-CA certificate for an eMobility Operator, Charging Station operator, or a V2G root certificate.
New text	The CSMS requests the Charging Station to install a new **CSMS root** certificate, **MO root** certificate for an eMobility Operator, **Manufacturer root**, or a V2G root certificate.

3.144. (Medium) - page 262, Use Case M05, Missing/incorrect requirements

New requirement:

ID	Precondition	Requirement definition
M05.FR.07	M05.FR.01 AND The certificate is invalid.	The Charging Station SHALL indicate rejection by setting 'status' to 'Rejected' in the *InstallCertificateResponse*.

Changed requirement M05.FR.02:

Old requirement definition	The Charging Station SHALL indicate success by setting 'status' to **'Success'** in the InstallCertificateResponse.
New requirement definition	The Charging Station SHALL indicate success by setting 'status' to **'Accepted'** in the InstallCertificateResponse.

3.145. (Minor) - Page 262, The specification does not specifically specify that it is allowed to install multiple certificates of the same type

Addition to remark(s) M05:

Remark	It is allowed to have multiple certificates of the same type installed.

3.146. (Minor) - Page 262, Use case M06, Description is unclear about when this Use case is triggered

When the cable get plugged in and a ISO 15118 supported EV get connected to the Charging Station, the Charging Station has to check the validity of the provided ISO 15118 certificate. Because this can take some time it is desirable to cache the OCSP certificate status for its ISO 15118 certificates beforehand. This has to be done at least once a week.

Old description	The Charging Station requests the CSMS to provide the OCSP certificate status for its ISO 15118 certificates.
New description	*When the cable gets plugged in and an ISO 15118 supported EV gets connected to the Charging Station, The EV requests the Charging Station to prove the validity of the (SubCa) certificates by a OCSPResponse. A request needs to be send per subCA. Because the timeout constraint in ISO 15118 is too strict to make the call to an external server, OCPP requires to cache the OCSP certificate status of the certificates beforehand. The Charging Station needs to refresh the cached OCSP data once a week.

Addition to remark(s) M06:

Remark	OCPP allows for only one certificate per *GetCertificateStatusRequest*. Because when multiple answers on a *GetCertificateStatusRequest* are to be expected, it makes handling the request and status more complex. So a *GetCertificateStatusRequest* needs to be sent per subCA.

3.147. (Minor) - Page 262, Use case M06, The Use case uses the term Charging Station certificate, this should be V2G Charging Station certificate

Rename Use case M06:

Old name	Get Charging Station Certificate status
New name	Get V2G Charging Station Certificate status

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	M06.FR.06		The Charging Station SHALL request and cache the OCSP status for its Charging Station intermediate certificates.
New	M06.FR.06		The Charging Station SHALL request and cache the OCSP status for its **V2G certificates**.

Removed requirement:

ID	Precondition	Requirement definition
M06.FR.05		The Charging Station SHALL request and cache the OCSP status for its Charging Station certificate.

3.148. (Minor) - page 266, use case N01, Prerequisite(s): Use Case Prerequisite too strict, also valid for security logs

Use Case N01 is for retrieving log information: diagnostics and security. The Prerequisite is now to strict, it only talks about diagnostics information available.

Old text	**Diagnostics information** is available for upload.
New text	**Requested information (either DiagnosticsLog or SecurityLog)** is available for upload.

3.149. (Minor) - page 266, use case N01, Error handling: sounds like a requirement

The current text of the N01 Error Handling reads like a requirement, but was meant to be a tip for handling hick-ups while uploading a file.

Old text	When the upload fails and the transfer protocol supports "resume" the Charging Station **SHOULD** try resume before aborting the upload.
New text	When the upload fails and the transfer protocol supports "resume" the Charging Station is **RECOMMENDED** to try to resume the upload before aborting it.

3.150. (Minor) - Page 266, Section 2.2, Using ISO 15118 Certificates in OCPP, Incorrect text about OCPP not supporting multiple V2G certificate chains

OCPP currently supports multiple V2G certificate chains, so the following text needs to be adjusted.

Old text	The V2G Charging Station Certificate needs to be derived from a V2G root. If this root is not known by the EV, no connection via 15118 is possible, so charging controlled by 15118 is NOT possible. In the event a Charging Station needs to support more than one V2G root, multiple V2G Charging Station Certificates are needed. **This is NOT supported by OCPP.**
New text	The V2G Charging Station Certificate needs to be derived from a V2G root. If this root is not known by the EV, no connection via 15118 is possible, so charging controlled by 15118 is NOT possible. In the event a Charging Station needs to support more than one V2G root, multiple V2G Charging Station Certificates are needed.

3.151. (Minor) - page 267, requirement N01.FR.06: Is not a requirement of N01

Use Case N01 is for retrieving log information, not about security events, so it is not logical to add a requirement that security events have to be logged to N01.

Removed requirement:

ID	Precondition	Requirement definition	Note
N01.FR.06		The Charging Station SHALL store log events	It is recommended to implement this in a rolling format.

3.152. (Minor) - Page 267, Use Case N01, Requirement N01.FR.11 not in scope of this Use Case

Requirement N01.FR.11 is not in scope of this Use Case. It is already covered by A04.FR.04. It needs to be removed here.

Removed requirement:

ID	Precondition	Requirement definition
N01.FR.11	When a security event happens	The Charging Station SHALL log this event in its security log. See Part 2 Appendices for a list of security events.

3.153. (Minor) - Page 267, Use Case N01, Missing requirement for status AcceptedCanceled

New requirement:

ID	Precondition	Requirement definition
N01.FR.12	When a Charging Station is assembling or uploading the log file AND the Charging Station receives a new GetLogRequest	The Charging Station SHOULD cancel the ongoing log file upload AND respond with status *AcceptedCanceled*.

3.154. (Minor) - Page 267, Use cases N, Section 2.2, Configure Monitoring

To the start of section 2.2, before use case N02 - Get Monitoring Report, add the following note:

NOTE	For managing the monitoring of a Charging Station a basic understanding of Device Model concepts is essential. These concepts are explained in "OCPP 2.0: Part 1 - Architecture & Topology", chapter 4.

3.155. (Medium) - Page 269, N02, B07, B08, Missing explanation about the sequence numbers used in the report messages

New requirements:

ID	Precondition	Requirement definition
N02.FR.09		The sequence number contained in the seqNo field of the NotifyMonitoringReportRequest is incremental per report. So the NotifyMonitoringReportRequest message which contains the first report part, SHALL have a seqNo with value *0*.
B07.FR.10		The sequence number contained in the seqNo field of the NotifyReportRequest is incremental per report. So the NotifyReportRequest message which contains the first report part, SHALL have a seqNo with value *0*.
B08.FR.14		The sequence number contained in the seqNo field of the NotifyReportRequest is incremental per report. So the NotifyReportRequest message which contains the first report part, SHALL have a seqNo with value *0*.

Addition to remark(s) N02, B07, B08:

Remark	It is recommended that a Charging Station sends Report messages in order, but the CSMS should take into account that this is not the case. In order to make it possible to know that all report messages were received, OCPP uses sequence numbers in combination with a tbc (to be continued) indicator. For every report, the Charging Station maintains a counter of the number of report messages generated for that report. With this mechanism, a CSMS can check if it has received all report information by checking that all report messages with seqNo 0 to n are received. n is the seqNo contained in the report message with tbc is *false*.

3.156. (Medium) - Page 269, Missing requirements regarding monitoringCriteria

New requirements:

ID	Precondition	Requirement definition
N02.FR.11	N02.FR.01 AND *monitoringCriteria* is empty AND *componentVariables* is empty.	The set of all existing monitors is reported in one or more NotifyMonitoringReportRequest messages.
N02.FR.12	If *monitoringCriteria* contains `ThresholdMonitoring`	All monitors with *type* = `UpperThreshold` or *type* = `LowerThreshold` are reported.
N02.FR.13	If *monitoringCriteria* contains `DeltaMonitoring`	All monitors with *type* = `Delta` are reported.
N02.FR.14	If *monitoringCriteria* contains `PeriodicMonitoring`	All monitors with *type* = `Periodic` or *type* = `PeriodicClockAligned` are reported.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	N02.FR.06	When the Charging Station receives a getMonitoringReportRequest for supported criteria	The Charging Station SHALL set the status field in the corresponding GetMonitoringReportResponse to: Accepted.
New	N02.FR.06	N02.FR.01 AND *monitoringCriteria* is NOT empty AND *componentVariables* is empty.	The set of monitors reported in one or more NotifyMonitoringReportRequest messages is limited to the set defined by *monitoringCriteria*.

3.157. (Minor) - Page 270, Requirement 'N03.FR.05' does not take hard-wired monitors into account

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	N03.FR.05	N03.FR.01 AND When the Charging Station received a setMonitoringBaseRequest with monitoringBase None	Then the Charging Station SHALL disable all monitoring.
New	N03.FR.05	N03.FR.01 AND When the Charging Station received a setMonitoringBaseRequest with *monitoringBase* **HardWiredOnly**	Then the Charging Station SHALL disable all **custom and pre-configured monitors. Only hard-wired monitors remain active.**

3.158. (Medium) - Page 270, Information missing about setting a delta VariableMonitor to a negative value

It is NOT be possible to set a delta VariableMonitor to a negative value.

New requirement:

ID	Precondition	Requirement definition
N04.FR.14	When the Charging Station receives a SetVariableMonitoringRequest with type *Delta* and value contains a negative value.	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: *OutOfRange*.

3.159. (Minor) - Page 270, It is not specified if a set variableMonitor is persistent across a reboot and after a firmware update

The specification should specify that all variableMonitors are persistent across reboot and also after a firmware update if possible.

Addition to remark(s) N04:

Remark	All variableMonitors are persistent across reboot.
Remark	A variableMonitor is persistent after a firmware update, if the monitored variable still exists and it is still monitor-able. Otherwise the VariableMonitor is removed.

3.160. (Medium) - Page 271, Information missing about replacing a PreconfiguredMonitor

After defining the set of monitor/notification groups, some use cases are missing monitor group specific requirements.

New requirement:

ID	Precondition	Requirement definition
N04.FR.15	N04.FR.12 AND The replaced VariableMonitor belonged to the 'PreconfiguredMonitors'.	The new VariableMonitor shall be classified as a 'CustomMonitor', until reset by a SetMonitoringBaseRequest.

3.161. (Medium) - Page 271, Replacing a VariableMonitor using a different Component/Variable combination

Using a different Component/Variable combination while replacing a VariableMonitor should NOT be allowed.

New requirement:

ID	Precondition	Requirement definition	Note
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.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	N04.FR.12	When the Charging Station receives a SetVariableMonitoringRequest with an Id AND A monitor exists matching the given Id.	The Charging Station SHALL replace the monitor.
New	N04.FR.12	When the Charging Station receives a SetVariableMonitoringRequest with an Id AND A monitor exists matching the given Id **AND The given Component/Variable combination corresponds with the existing VariableMonitor.**	The Charging Station SHALL replace the monitor.

3.162. (Minor) - Page 271, Unclear how to use monitors of type *Delta* for a variable that is NOT of a numeric type

It is not clear how to use monitors of type *Delta* for a variable that is NOT of a numeric type. A requirement needs to be added to explain this.

New requirement:

ID	Precondition	Requirement definition
N04.FR.17	When the CSMS sends a SetVariableMonitoringRequest with type *Delta* for a Variable that is NOT of a numeric type	It is RECOMMENDED to use a *monitorValue* of 1.

3.163. (Medium) - Page 271, Use case N04, Incorrect requirement

The Charging Station only needs to respond with status = OutOfRange, when trying to set a monitor of type *UpperThreshold* or *LowerThreshold* with a monitor value, which is below or above the limit.

Changed requirement:

Version	Id	Precondition	Requirement definition	Note
Old	N04.FR.06	When the Charging Station receives a SetVariableMonitoringRequest with a monitorValue that is lower or higher than the range of the given Variable in the SetMonitoringData	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: OutOfRange.	
New	N04.FR.06	When the Charging Station receives a SetVariableMonitoringRequest with monitor type *UpperThreshold* or *LowerThreshold* AND the monitor value is lower or higher than the range of the given Variable	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: OutOfRange.	

3.164. (Minor) - Page 271, Use case N04, requirement N04.FR.05 too strict

Changed requirement:

Version	Id	Precondition	Requirement definition	Note
Old	N04.FR.05	When the Charging Station receives a SetVariableMonitoringRequest with an MonitorType which is **unknown**	The Charging Station SHALL set the attributeStatus field in the corresponding SetMonitoringResult to: NotSupportedMonitorType.	
New	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.	

3.165. (Medium) - Page 274, Unclear how to handle exceeded *UpperThreshold* or *LowerThreshold* VariableMonitors after a reboot

The Charging Station must perform a check if the *UpperThreshold* or *LowerThreshold* is still exceeded after the reboot and needs to notify the CSMS about this. Requirements must be added to specify this.

New requirements:

ID	Precondition	Requirement definition
N07.FR.13		A VariableMonitoring needs to be stored persistently across reboots.
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.

3.166. (Minor) - Page 274, NotifyEventRequest, monitoring write-only variables is not described

It is allowed to set a VariableMonitor on a write-only variable, but the Charging Station should never report the actual value to the CSMS.

New requirement:

ID	Precondition	Requirement definition
N07.FR.10	When a monitor is triggered AND A VariableMonitor has been set on a write-only variable.	The actualField of the NotifyEventRequest SHALL be empty.

3.167. (Minor) - Page 274, Requirement "N07.FR.02" is unclear and does not cover enough

When a set *UpperThreshold* or *LowerThreshold* VariableMonitor is modified or removed, it is unclear how the Charging Station should handle this.

Changed requirement precondition N07.FR.02:

Old precondition	When a monitored value returns to within the set threshold

New precondition	When a monitored value returns to within the set *UpperThreshold* or *LowerThreshold*.

New requirements:

ID	Precondition	Requirement definition
N07.FR.11	When modifying a set *UpperThreshold* or *LowerThreshold* VariableMonitor.	The Charging Station SHALL check if the new threshold clears the old threshold OR if the new threshold is exceeded by the monitored value.
N07.FR.12	When removing a set *UpperThreshold* or *LowerThreshold* VariableMonitor AND The threshold is exceeded.	The Charging Station SHALL NOT send a notifyEventRequest with an eventData with the attribute cleared is true.

3.168. (Medium) - Page 274, Use Case N07, Missing requirements for all different monitor types

Only the description of the monitor types are explaining how they work, however there are no requirements defining this. Requirements need to be added for all different monitor types.

New requirements:

ID	Precondition	Requirement definition
N07.FR.15	When a monitor is triggered AND The *severity* of the monitor is greater than the monitoring severity level set in a SetMonitoringLevelRequest by the CSMS (see use case N05)	The Charging Station SHALL NOT send a NotifyEventRequest for the triggered monitor.
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.
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.
N07.FR.18	When there is a monitor with type Delta on a Component/Variable combination AND the Variable is of a numeric type AND the Actual value (attributeType Actual) of the Variable has changed more than plus or minus *monitorValue* since the time that this monitor was set or since the last time this event notice was sent, whichever was last	The Charging Station SHALL send a NotifyEventRequest with *trigger* Delta for the triggered monitor.
N07.FR.19	When there is a monitor with type Delta on a Component/Variable combination AND the Variable is NOT of a numeric type AND the Actual value (attributeType Actual) of the Variable has changed since the time that this monitor was set or since the last time this event notice was sent, whichever was last (Note: For variables that are not numeric, like boolean, string or enumerations, a monitor of type Delta will trigger an event notice whenever the variable changes, regardless of the value of *monitorValue*)	The Charging Station SHALL send a NotifyEventRequest with *trigger* Delta for the triggered monitor.
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.

ID	Precondition	Requirement definition
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.

Requirement 'N07.FR.01' is covered by above requirement, making it obsolete.

Removed requirement:

ID	Precondition	Requirement definition
N07.FR.01	When a VariableMonitoring setting is triggered for a Component/Variable combination AND The severity number of the monitor is equal to or lower than the monitoring severity level set in a SetMonitoringLevelRequest by the CSMS (see use case N05 - Set Monitoring Level)	The Charging Station SHALL send a notifyEventRequest.

3.169. (Medium) - Page 278, Use Case N10, Requirements incorrect / unclear / conflicting

Changed requirement:

Version	Id	Precondition	Requirement definition	Note
Old	N10.FR.01	When receiving a CustomerInformationRequest with the clear flag set to *true* and the Charging Station is in a state where it can process this request.	The Charging Station SHALL respond with a status *Accepted* and remove all customer related data for the Customer referred to by the customer identifier from the Charging Station, except from the Local Authorization List.	To prevent problems with Local Authorization List versions, only the CSMS can change the contents of the Local Authorization List.
New	N10.FR.01	When the Charging Station receives a CustomerInformationRequest AND it is in a state where it can process this request.	the Charging Station SHALL respond with a CustomerInformationResponse message with status *Accepted* .	
Old	N10.FR.03	When receiving a CustomerInformationRequest with the clear flag set to *true* AND the report flag set to *true* AND the Charging Station has information stored about the customer referred to by the customer identifier.	N10.FR.01 applies and the Charging Station SHALL send the requested information via one or more NotifyCustomerInformationRequest messages to the CSMS.	
New	N10.FR.03	N10.FR.01 AND receiving a CustomerInformationRequest with the clear flag set to *true* and the report flag set to *true* AND the Charging Station has information stored about the customer referred to by the customer identifier.	The Charging Station SHALL remove all customer related data for the Customer referred to by the customer identifier from the Charging Station, except from the LocalList AND the Charging Station SHALL send the cleared information via one or more NotifyCustomerInformationRequest messages to the CSMS.	To prevent problems with LocalList versions only the CSMS can change the contents of the LocalList.
Old	N10.FR.04	When receiving a CustomerInformationRequest with the clear flag set to *true* AND the report flag set to *true* AND the Charging Station has no information stored about the customer referred to by the customer identifier.	N10.FR.01 applies and the Charging Station SHALL send one NotifyCustomerInformationRequest message to the CSMS indicating that no data was found.	

Version	Id	Precondition	Requirement definition	Note
New	N10.FR.04	N10.FR.01 AND receiving a CustomerInformationRequest with the clear flag set to *true* and the report flag set to *true* AND the Charging Station has no information stored about the customer referred to by the customer identifier.	The Charging Station SHALL send one NotifyCustomerInformationRequest message to the CSMS indicating that no data was found.	
Old	N10.FR.06	When receiving a CustomerInformationRequest with the clear flag set to *true*, the report flag set to *false* and the Charging Station is in a state where it can process this request.	The Charging Station SHALL respond with a status *Accepted* and remove all customer related data for the Customer referred to by the customer identifier from the Charging Station, except from the LocalList. The Charging Station SHALL send one NotifyCustomerInformationRequest message to the CSMS indicating that the data was cleared.	To prevent problems with LocalList versions only the CSMS can change the contents of the LocalList.
New	N10.FR.06	N10.FR.01 AND receiving a CustomerInformationRequest with the clear flag set to *true*, the report flag set to *false*	The Charging Station SHALL remove all customer related data for the Customer referred to by the customer identifier from the Charging Station, except from the LocalList AND the Charging Station SHALL send one NotifyCustomerInformationRequest message to the CSMS indicating that the data was cleared.	To prevent problems with LocalList versions only the CSMS can change the contents of the LocalList.

3.170. (Minor) - Page 284, Requirement O02.FR.11 and O02.FR.13 are duplicates

Removed requirement:

ID	Precondition	Requirement definition
O02.FR.13	When the Charging Station receives a SetDisplayMessageRequest and the total number of messages after having handled this request will exceed NumberOfDisplayMessages.maxLimit.	The Charging Station SHALL respond with status: Rejected.

3.171. (Minor) - Page 284, Requirement 'O03.FR.01', There needs to be at least one display message for the Charging Station to accept the GetDisplayMessagesRequest

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	O03.FR.01	When all fields except requestId in a GetDisplayMessagesRequest are omitted	The Charging Station SHALL always respond with Accepted.
New	O03.FR.01	When all fields except requestId in a GetDisplayMessagesRequest are omitted **AND at least one display message is configured.**	The Charging Station SHALL respond with Accepted.

3.172. (Major) - Page 296, It should not be possible to freely (without any format) extend any message/dataType/enum with additional custom values/fields

All definitions of types in the OCPP 2.0 JSON schemas have "additionalProperties": true, which allows for any message/dataType/enum to be extended with additional fields. In previous OCPP versions this was always: "additionalProperties": false, because a system needs to be able to perform strict schema checks. Otherwise a CSMS or Charging Station can start sending extra information that the other may not be able to understand and this will lead to interoperability problems. Therefore additionalProperties should be *false*. It is recommended to first check if the custom behavior is able to be implemented using the device model, otherwise DataTransfer message(s) and/or CustomData fields can be used. The custom DataType can be included to any other DataType or Message, using the structure below. By having set additionalProperties to true for this DataType only, the implementer is able to create a custom data structure within an enclosed space.

Message/dataType	Field	Type	Card.	Description	Action
<All Messages and DataTypes>	customData	CustomData	0..*	Optional. This field may be used to add vendor-specific data to any DataType or Message. The additional data sent needs to be decided upon by both parties.	added
CustomDataType	vendorId	string[0..255]	1..1	Required. This identifies the Vendor specific implementation.	added

New Schema files are available.

3.172.1. Page 291, P. DataTransfer, Introduction, information needs to be added about the CustomData element

Old text	This Functional Block describes the functionality that enables parties to add custom commands to OCPP, enabling custom extensions to OCPP. The mechanism of vendor-specific data transfer allows for the exchange of data or messages not standardized in OCPP. As such, it offers a framework within OCPP for experimental functionality that may find its way into future OCPP versions. Experimenting can be done without creating new (possibly incompatible) OCPP dialects. Secondly, it offers a possibility to implement additional functionality agreed upon between specific CSMS and Charging Station vendors. Behaviour of this operation is identical for Data Transfer messages initiated either by the CSMS or the Charging Station.
New text	This Functional Block describes the functionality that enables parties to **extend existing commands with custom attributes or add new custom commands to OCPP** **OCPP offers two mechanisms to create vendor-specific custom extension.** 1. **The DataTransferRequest message allows for the exchange** of data or messages not standardized in OCPP. As such, it offers a framework within OCPP for experimental functionality that may find its way into future OCPP versions. Experimenting can be done without creating new (possibly incompatible) OCPP dialects. Secondly, it offers a possibility to implement additional functionality agreed upon between specific CSMS and Charging Station vendors. 2. **A CustomData element exists as an optional element in the JSON schemas of all types. CustomData is the only class in the JSON schema files that allows additional properties. It can thus be used to add additional custom attributes to any type. The CustomData has been deliberately left out of the specification document, because it would introduce a lot of clutter and it is not meant to be used in standard implementations. See also [OCPP2.0-PART4].**

3.173. (Medium) - Page 297, Fields starting with a number cause serialization problems

All field names starting with a number should be changed. The only fields starting with a number are fields starting with 15118.

Page	Section	Message/dataType	Old Field Name	New Field Name
297	1.1.1	AuthorizeRequest	15118CertificateHashData	iso15118CertificateHashData
303	1.16.1	Get15118EVCertificateRequest	15118SchemaVersion	iso15118SchemaVersion
326	1.66.1	Update15118EVCertificateRequest	15118SchemaVersion	iso15118SchemaVersion

There also is one EnumType that starts with 15118, which needs to be changed.

Page	Section	Old Name	New Name
351	2.1	15118EVCertificateStatusEnumType	Iso15118EVCertificateStatusEnumType

New Schema files are available.

3.174. (Major) - Page 298, Certificate length fields too short

The length of the field to send certificate (and CSR) to/from a Charging Station is too short, it was 800. But certificates and a CSR (Certificate Signing Request) can be much larger then 800 characters. The length of a certificate depends a lot on the data inside the certificate, but looking at the use case of EV Charging it is not foreseen that certificate will be longer then 5000. Just to be sure: 5500 is chosen is the correct length which allows for even more optional data in the certificates. this should be more then enough for even the most complex certificate structures.

NOTE The name, length and cardinality of the cert field of the CertificateSignedRequest PDU are changed at the following errata: (Major) - Page 298, section 1.4.1, CertificateSignedRequest, certificate field size of 5500 may cause that a Charging Station has to allocate a lot of memory

Page	Section	Message/dataType	Field	Old Length	New Length
298	1.4.1	CertificateSignedRequest	cert	800	5500
309	1.30.1	InstallCertificateRequest	certificate	800	5500
323	1.60.1	SignCertificateRequest	csr	800	5500
336	1.24	FirmwareType	signingCertificate	800	5500

New Schema files are available.

3.175. (Major) - Page 298, Serial Number length fields too short

The length of the certificate serialNumber of a couple of messages is too short, it was 20, but should have been 40.

Page	Section	Message/dataType	Field	Old Length	New Length
298	1.4.1	OCSPRequestDataType	serialNumber	20	40
298	1.4.1	DeleteCertificateRequest	serialNumber	20	40
298	1.4.1	GetInstalledCertificateIdsResponse	serialNumber	20	40
298	1.4.1	CustomerInformationRequest	serialNumber	20	40

New Schema files are available.

3.176. (Major) - Page 298, section 1.4.1, CertificateSignedRequest, certificate field size of 5500 may cause that a Charging Station has to allocate a lot of memory

The errata (Major) - Page 298, Certificate length fields too short did not take into account that the certificate field of the CertificateSignedRequest can contain a certificate chain. Every element in the array will have a maximum size of 5500, which may cause that a Charging Station has to allocate a lot of memory when it is configured to support receiving a (long) certificate chain. By bundling the whole certificate chain within one data element the total size of the message field can be smaller, because then the calculated margin will be applied only one time, instead of times the chain length. This will also improve the flexibility to allow either multiple smaller certificates part of a longer chain or larger certificates part of a shorter chain. An estimation - including three (RSA) intermediate certificates and the leaf certificate using 4k bit keys - concluded that we need to support a maximum size of 10.000, to

be future proof. However this size will not be necessary for most current implementation. So in addition a configuration variable will be defined, which limits the maximum size of the certificate chain.

Version	Message/dataType	Field	Card.	length
Old	CertificateSignedRequest	cert	1..*	5500
New	CertificateSignedRequest	certificateChain	1..1	10000

New Schema files are available.

Configuration variable 'MaxCertificateChainSize':

Required	no		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	MaxCertificateChainSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	10000
Description	This configuration variable can be used to limit the size of the 'certificateChain' field from the CertificateSignedRequest PDU.		

New requirements:

ID	Precondition	Requirement definition
A02.FR.16	If the configuration variable MaxCertificateChainSize is implemented AND The Charging Station receives a CertificateSignedRequest message with a certificate (chain) with with a size that exceeds the set value configured at MaxCertificateChainSize	The Charging Station SHALL respond with a CertificateSignedResponse message with status *Rejected* .
A03.FR.16	If the configuration variable MaxCertificateChainSize is implemented AND The Charging Station receives a CertificateSignedRequest message with a certificate (chain) with with a size that exceeds the set value configured at MaxCertificateChainSize	The Charging Station SHALL respond with a CertificateSignedResponse message with status *Rejected* .

3.177. (Medium) - Page 298, There is a discrepancy in the specification about the certificate encoding

Requirement A00.FR.507 states the following: "The certificates SHALL be stored and transmitted in the X.509 format encoded in Distinguished Encoding Rules (DER) format, followed by Base64 encoding." However some certificates in the specification are described to be hex encoded. To be consistent it is decided to encode all certificates with Base64.

Changed descriptions:

Page	Section	Message/dataType	Field Name	Old description	New description
298	1.4.1	CertificateSignedRequest	cert	Required. The signed X.509 certificate, first DER encoded into binary, and then hex encoded into a case insensitive string. This can also contain the necessary sub CA certificates. In that case, the order should follow the certificate chain, starting from the leaf certificate.	Required. The signed **PEM encoded** X.509 certificate. This can also contain the necessary sub CA certificates. In that case, the order should follow the certificate chain, starting from the leaf certificate.

Page	Section	Message/dataType	Field Name	Old description	New description
310	1.30.1	InstallCertificateRequest	certificate	Required. An X.509 certificate, first DER encoded into binary, and then hex encoded into a case insensitive string.	Required. A **PEM encoded** X.509 certificate.
336	1.24	FirmwareType	signingCertificate	Optional. Certificate with which the firmware was signed. X.509 certificate, first DER encoded into binary, and then hex encoded into a case insensitive string.	Optional. Certificate with which the firmware was signed. **PEM encoded** X.509 certificate.

3.178. (Major) - Page 299, ClearChargingProfileRequest, criteria fields incorrectly structured

Basically the ClearChargingProfileRequest is like a logical AND, the Charging Station is asked to look for charging profiles that match all fields. The only field that is a bit different from the four is: (chargingProfile)id. A CSMS either provides the id, or 1 or more of the 3 other fields. Now with id inside: ClearChargingProfileType and evseId at the top layer the intended way of filtering is not possible.

Page	Section	Message/dataType	Field	Type	Action
299	1.7.1	ClearChargingProfileRequest	evseId	integer	removed
299	1.7.1	ClearChargingProfileRequest	chargingProfileId	integer	added
333	1.14	ClearChargingProfileType	id	integer	removed
333	1.14	ClearChargingProfileType	evseId	integer	added

The previous change causes the field name "chargingProfile" to be incorrect.

Page	Section	Message/dataType	Old Field Name	New Field Name
299	1.7.1	ClearChargingProfileRequest	chargingProfile	chargingProfileCriteria

New Schema files are available.

3.179. (Minor) - Page 299, Messages, section 1.7.1: Unclear how to match fields in a ClearChargingProfile request

This needs to be read **after** erratum "3.3 ClearChargingProfileRequest, criteria fields incorrectly structured".

Old text	This contains the field definition of the ClearChargingProfileRequest PDU sent by the CSMS to the Charging Station. The CSMS can use this message to clear (remove) either a specific charging profile (denoted by id) or a selection of charging profiles that match with the values of the optional evse, stackLevel and ChargingProfilePurpose fields.
New text	This contains the field definition of the ClearChargingProfileRequest PDU sent by the CSMS to the Charging Station. The CSMS can use this message to clear (remove) either 1. all charging profiles (when no parameters are provided), 2. or a specific charging profile (when **id** is provided) 3. or all charging profiles that match (as logical AND) with the optional **evseId**, **chargingProfilePurpose** and **stackLevel** in **chargingProfileCriteria** (when no **id** is provided).

3.180. (Major) - Page 303, exiRequest and exiResponse maximum size too small for ISO 15118 certificate messages

The ISO 15118 CertificateUpdate message can have a size of up to 5588 bytes and the CertificateInstallation is only insignificantly smaller with 5585 bytes. This means that the current exiRequest and exiResponse maximum size of 5500 is too small.

Page	Section	Message/dataType	Field	Old Field Type	New Field Type
303	1.16.1	Get15118EVCertificateRequest	exiRequest	string[0..5500]	string[0..5600]
304	1.16.2	Get15118EVCertificateResponse	exiResponse	string[0..5500]	string[0..5600]

New Schema files are available.

3.181. (Medium) - Page 303, Section 1.15.1, FirmwareStatusNotificationRequest, No requestId can be given when triggered by TriggerMessageRequest AND no log upload is ongoing

Changed Cardinality:

Page	Section	Message/dataType	Field	Old Card.	New Card.	Old Description	New Description
303	1.15.1	FirmwareStatusNotificationRequest	requestId	1..1	0..1	Required. The request id that was provided in the UpdateFirmwareRequest that started this firmware update.	Optional. The request id that was provided in the UpdateFirmwareRequest that started this firmware update. This field can only be omitted, when *status* = `Idle`_.

New Schema files are available.

New requirements:

ID	Precondition	Requirement definition
L01.FR.20		The field *requestId* in FirmwareStatusNotificationRequest is mandatory, unless *status* = `Idle`.
L01.FR.25	Charging Station receives a TriggerMessageRequest for `FirmwareStatusNotification` AND last sent FirmwareStatusNotificationRequest had *status* = `Installed`	Charging Station SHALL return a FirmwareStatusNotificationRequest with *status* = `Idle`.
L01.FR.26	Charging Station receives a TriggerMessageRequest for `FirmwareStatusNotification` AND last sent FirmwareStatusNotificationRequest had NOT *status* `Installed`	Charging Station SHALL return a FirmwareStatusNotificationRequest with the last sent *status*.
L01.FR.27	L01.FR.24 AND the Charging Station is unable to cancel the firmware installation	The Charging Station MAY respond with *status* = `Rejected`.
L02.FR.14		The field *requestId* in FirmwareStatusNotificationRequest is mandatory, unless *status* = `Idle`.
L02.FR.16	Charging Station receives a TriggerMessageRequest for `FirmwareStatusNotification` AND last sent FirmwareStatusNotificationRequest had *status* = `Installed`	Charging Station SHALL return a FirmwareStatusNotificationRequest with *status* = `Idle`.

ID	Precondition	Requirement definition
L02.FR.17	Charging Station receives a TriggerMessageRequest for `FirmwareStatusNotification` AND last sent FirmwareStatusNotificationRequest had NOT *status* `Installed`	Charging Station SHALL return a FirmwareStatusNotificationRequest with the last sent *status*.
L02.FR.18	L02.FR.15 AND the Charging Station is unable to cancel the firmware installation	The Charging Station MAY respond with *status* = `Rejected`.
L03.FR.10	Charging Station receives a TriggerMessageRequest for `PublishFirmwareStatusNotification` AND last sent PublishFirmwareStatusNotificationRequest had *status* = `Published`	Charging Station SHALL return a PublishFirmwareStatusNotificationRequest with *status* = `Idle`.
L03.FR.11	Charging Station receives a TriggerMessageRequest for `PublishFirmwareStatusNotification` AND last sent PublishFirmwareStatusNotificationRequest had NOT *status* `Published`	Charging Station SHALL return a PublishFirmwareStatusNotificationRequest with the last sent *status*.

3.182. (Medium) - Page 304, Duplicate fields in *Get15118EVCertificateResponse*

The fields *contractSignatureCertificateChain* and *saProvisioningCertificateChain* are already included in the *exiResponse* field.

Page	Section	Message/dataType	Field	Action
304	1.16.2	Get15118EVCertificateResponse	contractSignatureCertificateChain	removed
304	1.16.2	Get15118EVCertificateResponse	saProvisioningCertificateChain	removed

New Schema files are available.

3.183. (Minor) - Page 305, GetCompositeScheduleStatusEnumType contains the same arguments and the same description as GenericStatusEnumType

GetCompositeScheduleStatusEnumType is not needed and can be removed and replaced by GenericStatusEnumType.

Changed field type:

Page	Section	Message/dataType	Field	Old field type	New field type
305	1.20.2	GetCompositeScheduleResponse	status	GetCompositeScheduleStatusEnumType	GenericStatusEnumType

Removed EnumType:

Page	Section	Message/dataType	Action
361	2.38	GetCompositeScheduleStatusEnumType	removed

3.184. (Minor) - Page 305, Section 1.20.2, GetCompositeScheduleResponse, Unclear how to implement composite schedule

There is much unclarity about the structure and fields of the GetCompositeScheduleResponse message.

- It is unclear how/when to use which schedule start field.

- The message contains fields, which can/should never be used.

Therefore the compositeScheduleType will be given all necessary fields and will replace the ChargingScheduleType in GetCompositeScheduleResponse.

Page	Section	Message/data Type	Field	Type	Card.	Description	Action
305	1.20.2	GetCompositeScheduleResponse	evseId	integer	1..1	Required. The charging schedule contained in this notification applies to an EVSE.	removed
334	1.18	CompositeScheduleType	evseId	integer	1..1	Required. The **ID of the EVSE for which the schedule is requested. When evseid=0, the Charging Station calculated the expected consumption for the grid connection.**	added
334	1.18	CompositeScheduleType	chargingSchedule	ChargingScheduleType	0..1	Optional. Charging schedule structure defines a list of charging periods.	removed
334	1.18	CompositeScheduleType	chargingRateUnit	ChargingRateUnitEnumType	1..1	Required. The unit of measure Limit is expressed in.	added
334	1.18	CompositeScheduleType	duration	integer	1..1	Required. Duration of the schedule in seconds.	added
334	1.18	CompositeScheduleType	chargingSchedulePeriod	ChargingSchedulePeriodType	1..*	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time.	added

Changed field name and cardinality:

Page	Section	Message/dataType	Old Field Name	New Field Name	Old Card.	New Card.
334	1.18	CompositeScheduleType	startDateTime	scheduleStart	0..1	1..1

Changed description:

Page	Section	Message/dataType	Field Name	Old description	New description
334	1.18	CompositeScheduleType	schedule	Optional. The requested schedule.	Optional. **This field contains the calculated composite schedule. It may only be omitted when this message contains status *Rejected*.**

3.185. (Minor) - Page 307, Messages, section 1.24, Description mentions 'try' rather than 'retry'

In the description of the field **retries** the word "try" should be "retry".

Old text	Optional. This specifies how many times Charging Station must try to upload the diagnostics before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry.
New text	Optional. This specifies how many times Charging Station must retry to upload the diagnostics before giving up. If this field is not present, it is left to Charging Station to decide how many times it wants to retry.

3.186. (Medium) - Page 310, Section 1.31.1, LogStatusNotificationRequest, No requestId can be given when triggered by TriggerMessageRequest AND no log upload is ongoing

Changed Cardinality:

Page	Section	Message/dataType	Field	Old Card.	New Card.	Old Description	New Description
310	1.31.1	LogStatusNotification Request	requestId	1..1	0..1	Required. The request id that was provided in the GetLogRequest that started this log upload.	Optional. The request id that was provided in GetLogRequest that started this log upload. This field is mandatory, unless the message was triggered by a TriggerMessageRequest AND there is no log upload ongoing.

New Schema files are available.

New requirement:

ID	Precondition	Requirement definition
N01.FR.13		The field requestId in LogStatusNotificationRequest is mandatory, unless the message was triggered by a TriggerMessageRequest AND there is no log upload ongoing.

3.187. (Medium) - Page 312, Delivery of empty reports not possible

The use cases Get Custom Report, Get Monitoring Report and Get DisplayMessages may select an empty report set. When this happens the Charging Station has no report information to send, but the messages do not allow empty arrays of report elements.

Page	Section	Message/dataType	Field	Old Card.	New Card.
312	1.36.1	NotifyDisplayMessagesRequest	messageInfo	1..*	0..*
314	1.40.1	NotifyMonitoringReportRequest	monitor	1..*	0..*
314	1.41.1	NotifyReportRequest	reportData	1..*	0..*

3.188. (Minor) - Page 312, The tbc field in all messages should have been optional with false as a default when omitted

By keeping the tbc field required, implementers are forced to always include the field. Even Though in most cases the value would just be *false*.

Page	Section	Message/dataType	Field	Old Card.	New Card.
312	1.35.1	NotifyCustomerInformationRequest	tbc	1..1	0..1
312	1.36.1	NotifyDisplayMessagesRequest	tbc	1..1	0..1
313	1.39.1	NotifyEventRequest	tbc	1..1	0..1
314	1.40.1	NotifyMonitoringReportRequest	tbc	1..1	0..1
314	1.41.1	NotifyReportRequest	tbc	1..1	0..1

3.189. (Medium) - Page 314, Publish firmware mechanism not inline with update firmware mechanism

```
The requestId fields are missing.
```

Page	Section	Message/dataType	Field	Type	Card.	Description	Action
314	1.42.1	PublishFirmwareRequest	requestId	integer	1..1	Required. The Id of the request.	Added
315	1.43.1	PublishFirmwareStatus-NotificationRequest	requestId	integer	0..1	Optional. The request id that was provided in the PublishFirmwareRequest which triggered this action. Mandatory, unless *status* = `Idle`.	Added

The retryInterval field is missing in the PublishFirmwareRequest.

Page	Section	Message/dataType	Field	Type	Card.	Description	Action
314	1.42.1	PublishFirmwareRequest	retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charging Station to decide how long to wait between attempts.	Added

3.189.1. The enumeration PublishFirmwareEnumType must be extended with `Idle` status.

On page 268, section 2.62 PublishFirmwareEnumType, add the following status:

Value	Description
Idle	Successful end state. No publishing of firmware is active. (Only applicable in response to a TriggerMessageRequest)

3.190. (Major) - Page 316, Messages 1.46, RequestStartTransaction

In order to allow a remote start on an EVSE which holds a reservation for a group ID, we have added an attribute *groupIdToken* to RequestStartTransactionRequest, as follows:

Field Name	Field Type	Card.	Description
groupIdToken	GroupIdTokenType	0..1	Optional. The group identifier that the Charging Station must use to start a transaction.

3.191. (Major) - Page 317, 1.49 Flattening of ReserveNow message structure

In order to make the message more clear, the attributes in ReservationType are moved to top level. The structure of message is now as follows:

Field Name	Field Type	Card.	Description
id	integer	1..1	Required. Id of reservation.
expiryDateTime	dateTime	1..1	Required. Date and time at which the reservation expires.
connectorType	ConnectorEnumType	0..1	Optional. This field specifies the connector type.
evseId	integer	0..1	Optional. This contains ID of the evse to be reserved.

Field Name	Field Type	Card.	Description
idToken	IdTokenType	1..1	Required. The identifier for which the reservation is made.
groupIdToken	IdTokenType	0..1	Optional. The group identifier for which the reservation is made.

3.192. (Medium) - Page 318, SecurityEventNotificationRequest has a 'type' field that is incorrectly been linked to type: SecurityEventEnumType

The idea of OCPP Appendices document was: to have a enums that can be extended. The protocol would then contain only a string field. But the Appendices document would contain a list of allowed values. We can then easily update this Appendices document and release it without having impact on existing implementations.

Page	Section	Message/dataType	Field	Old Type	New Type
318	1.51	SecurityEventNotificationRequest	type	SecurityEventEnumType	string[0..50]

New Schema files are available.

3.193. (Major) - Page 318, 1.50.1 ResetRequest has optional parameter for EVSE ID

The ResetRequest has been extended with an optional parameter *evseId* to request the Charging Station to only reset a specific EVSE. New message definition:

Field Name	Field Type	Card.	Description
evseId	integer	0..1	Optional. This contains the ID of a specific EVSE that needs to be reset, instead of the entire Charging Station.
type	ResetEnumType	1..1	Required. This contains the type of reset that the Charging Station or EVSE should perform.

3.194. (Minor) - Page 318, Section 1.51.1, SecurityEventNotificationRequest, Missing human-readable information about the occurred security event

This functionality is far less useful when the Charging Station is unable to provide some extra human-readable information about the occurred security event.

Page	Section	Message/dataType	Field	Type	Card.	Description	Action
318	1.51.1	SecurityEventNotificationRequest	techInfo	string[0..255]	0..1	Additional information about the occurred security event.	added

New Schema files are available.

3.195. (Minor) - Page 321, Inconsistent field names for certificates and certificate types

Changed field names:

Page	Section	Message/dataType	Old Field Name	New Field Name
321	1.22.1	GetInstalledCertificateIdsRequest	typeOfCertificate	certificateType
333	1.4.1	CertificateSignedRequest	typeOfCertificate	certificateType
333	1.4.1	CertificateSignedRequest	cert	certificate

Page	Section	Message/dataType	Old Field Name	New Field Name
339	1.60.1	SignCertificateRequest	typeOfCertificate	certificateType

3.196. (Major) - Page 323, *requestId* needs to be mandatory in almost all messages in which it is being used

requestId needs to be mandatory in all messages, except FirmwareStatusNotificationRequest, PublishFirmwareStatusNotificationRequest and LogStatusNotificationRequest. Because these messages could be triggered by TriggerMessageRequest.

Page	Section	Message/DataType	Field Name	Old Card.	New Card.
323	1.26.1	GetReportRequest	requestId	0..1	1..1
320	1.19.1	GetChargingProfilesRequest	requestId	0..1	1..1
323	1.25.1	GetMonitoringReportRequest	requestId	0..1	1..1
327	1.35.1	NotifyCustomerInformationRequest	requestId	0..1	1..1
331	1.45.1	ReportChargingProfilesRequest	requestId	0..1	1..1
330	1.41.1	NotifyReportRequest	requestId	0..1	1..1
329	1.40.1	NotifyMonitoringReportRequest	requestId	0..1	1..1

3.197. (Minor) - Page 324, Unclear naming of transaction related information in *TransactionEventRequest*

The naming of the *transactionData* field caused confusion. The following fields have been renamed to make it more clear.

Page	Section	Message/dataType	Old Field Name	New Field Name
324	1.62.1	TransactionEventRequest	transactionData	transactionInfo
346	1.51	TransactionType	id	transactionId

New Schema files are available.

3.198. (Medium) - Page 332, *chargingSchedulePeriod* in ChargingScheduleType is optional, but should be required

The chargingSchedulePeriod field should never be empty. If no ChargingProfile is installed, then it should report one period with startPeriod = 0 and limit = maximum charging rate of the Charging Station.

Page	Section	Message/dataType	Field	Old Card.	New Card.
332	1.12	ChargingScheduleType	chargingSchedulePeriod	0..*	1..*

New Schema files are available.

3.199. (Minor) - Page 332, section 1.11, ChargingSchedulePeriodType

Unclear why the field numberOfPhases is needed when charging mode is DC.

Changed description:

Old description	Optional. The number of phases that can be used for charging. If a number of phases is needed, numberPhases=3 will be assumed unless another number is given.
New description	Optional. The number of phases that can be used for charging. If a number of phases is needed, numberPhases=3 will be assumed unless another number is given. **This field is also relevant for charging mode DC, because the phases are not specifically meant for the EV. They are meant for the DC charger connected to the rectifier.**

3.200. (Minor) - Page 332, section 1.10, ChargingProfileType, unclear when allowed to use recurrencyKind

It is unclear when it is allowed to use the field recurrencyKind. The use of the field should only be allowed when *chargingProfileKind* is *Recurring*.

Changed description:

Old	Optional. Indicates the start point of a recurrence.
New	Optional. Indicates the start point of a recurrence. **This field is only allowed to be used when *chargingProfileKind* is *Recurring*.**

3.201. (Minor) - Page 333, Section 1.13, ChargingStationType, SerialNumber length to short

In OCPP 1.6 the serialNumber in the BootNotification has a length of 25, while OCPP 2.0 describes a length of 20. This may cause issues when upgrading existing Charging Stations.

Changed field type:

Message/dataType	Field	Old field type	New field type
ChargingStationType	serialNumber	string[0..20]	string[0..25]

3.202. (Medium) - Page 334, componentVariable.variable used in GetMonitoringReport and GetReport is 1..1, but should be 0..1

Now it is not possible to request all variables of a specific component, without duplicating the componentVariable multiple times and specifying every variable individually.

Page	Section	Message/dataType	Field	Old Card.	New Card.
334	1.17	componentVariable	variable	1..1	0..1

New Schema files are available.

3.203. (Minor) - Page 335, Section 1.20, CostType, The field amount has field type decimal but should be integer

CostType has 'decimal' amount and 'integer' amountMultiplier, mixing floating point and fixed point semantic. This field is fixed point in ISO 15118 having integer amount and integer multiplier.

Changed field type:

Page	Section	Message/dataType	Field	Old field type	New field type
335	1.20	CostType	amount	decimal	integer

3.204. (Medium) - Page 336, It is not possible to specify to which monitor/notification group an EventNotification belongs to.

A new enum is created and added to NotifyEventRequest to specify the monitor/notification group of the EventNotification.

Page	Section	Message/dataType	Field	Type	Action
336	1.22	NotifyEventRequest	eventNotificationType	EventNotificationEnumType	added

New EnumType 'EventNotificationEnumType':

Value	Description
HardWiredNotification	The software implemented by the manufacturer triggered a hardwired notification. Using this mechanism it is not possible to provide a severity. If this is needed it is recommended to use 'HardWiredMonitor' mechanism instead.
HardWiredMonitor	Triggered by a monitor, which is hardwired by the manufacturer.
PreconfiguredMonitor	Triggered by a monitor, which is preconfigured by the manufacturer.
CustomMonitor	Triggered by a monitor, which is set with the SetVariableMonitoringRequest message by the Charging Station Operator.

New Schema files are available.

3.205. (Medium) - Page 336, Unable to link an EventNotification to a specific transaction

Adding a transactionId field to the NotifyEventRequest message will give the Charging Station the ability to link an EventNotification to a specific transaction.

Page	Section	Message/dataType	Field	Field Type	Card.	Description	Action
336	1.22	EventDataType	transactionId	identifierString[0..36]	0..1	If an event notification is linked to a specific transaction, this field can be used to specify its transactionId.	added

New Schema files are available.

3.206. (Medium) - Page 336, NotifyEventRequest contains unneeded fields with information about the VariableMonitor which triggered the event

When sending a NotifyEventRequest it MAY be needed to specify which VariableMonitor triggered the event. To achieve this only the id of the VariableMonitor is needed, not the whole VariableMonitoringType.

Page	Section	Message/dataType	Field	Type	Card.	Description	Action
336	1.22	EventDataType	variableMonitoringEvent	VariableMonitoringType	0..1	Optional. The (list of) variable monitoring settings that triggered the event, can be empty if the event was triggered by a hard-wired monitor in the Charging Station.	removed
336	1.22	EventDataType	VariableMonitoringId	integer	0..1	Optional. Identifies the VariableMonitoring which triggered the event.	added

New Schema files are available.

3.207. (Medium) - Page 336, NotifyEventRequest, cleared field is not always required.

The cleared field is only needed when a monitored value returns within the configured threshold. Always using the cleared field with the value set to 'false' causes a lot of unneeded information to be sent. It also gives the impression that the monitor is able to be cleared when used with monitor type 'Periodic' for example, which is not possible.

Page	Section	Message/dataType	Field	Old Card.	New Card.
336	1.22	EventDataType	cleared	1..1	0..1

New Schema files are available.

3.208. (Medium) - Page 337, Signed meter data cannot be handled by Device Model value sizes

All Device model value fields have a maximum of 1000, which can be limited by the *ValueSize* configuration variable. However the maximum size of a signed meter value is 2500. So it is not possible to send signed meter values using the Device model. To make this possible it is needed to increase the size of a few values used in the Device model, not all.

Page	Section	Message/dataType	Field	Old Length	New Length
337	1.26	GetVariableResult	attributeValue	1000	2500
347	1.53	VariableAttribute	value	1000	2500
336	1.22	EventData	actualValue	1000	2500

New Schema files are available.

Because of previous change it is also needed to split the Configuration variable *ValueSize* in two.

Configuration variable 'ConfigurationValueSize':

Required	no		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ConfigurationValueSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	1000
Description	This Configuration Variable can be used to limit the following fields: SetVariableData.attributeValue and VariableCharacteristics.valueList. The max size of these values will always remain equal.		

Configuration variable 'ReportingValueSize':

Required	no		
Component	**componentName**	DeviceDataCtrlr	
Variable	**variableName**	ReportingValueSize	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
		maxLimit	2500
Description	This Configuration Variable can be used to limit the following fields: GetVariableResult.attributeValue, VariableAttribute.value and EventData.actualValue. The max size of these values will always remain equal.		

3.209. (Medium) - Page 340, *responderURL* is optional, but should be required

It is not possible to check the revocation status without a responderURL, so it should be required.

Page	Section	Message/dataType	Field	Old Card.	New Card.
340	1.37	OCSPRequestDataType	responderURL	0..1	1..1

New Schema files are available.

Addition to remark(s) M06:

Remark	responderURL is required in OCPP, while it is optional in ISO 15118. Without a responderURL in a certificate it cannot work, so a responderURL is required for any certificate for which a GetCertificateStatusRequest can be expected.

3.210. (Major) - Page 341, Messages 1.42, SalesTariffEntryType

The attribute *relativeTimeInterval* should be mandatory.

Field Name	Field Type	Card	Description
relativeTimeInterval	RelativeTimeIntervalType	**1..1**	**Required.** Defines the time interval the SalesTariffEntry is valid for, based upon relative times

3.211. (Minor) - Page 341, The specification has a discrepancy in the naming of the type of a connector

ConnectorCode and ConnectorType are both used for the same thing.

Page	Section	Message/dataType	Old Field Name	New Field Name
341	1.41	ReservationType	connectorCode	connectorType

3.212. (Minor) - page 341, Section 1.63.1, TriggerMessageRequest, Missing enum value for PublishFirmwareStatusNotificationRequest

There are already values for FirmwareStatusNotificationRequest and LogStatusNotificationRequest, but not for PublishFirmwareStatusNotificationRequest. This needs to be added.

Added Enum value:

Page	section	Message/dataType	Value	Description
341	1.63.1	TriggerMessageRequest	PublishFirmwareStatusNotification	To trigger PublishFirmwareStatusNotification.

3.213. (Medium) - Page 343, SetVariableMonitor, it is not clear which monitorTypes can be set on certain dataTypes

To make more clear what combinations are possible, a matrix and explanation needs to added to part 1 chapter 4 of the specification. This is where the Device Model is explained. But the explanation of what the value in SetMonitoringDataType represents based on the used dataType, also needs to be included to the field description. *Changed description:*

Page	Section	Message/dataType	Field	Old Description	New Description
343	1.46	SetMonitoringDataType	value	Required. Value for threshold or delta monitoring. For Periodic or PeriodicClockAligned this is the interval in seconds.	Required. For *UpperThreshold* and *LowerThreshold* this value represents the to be exceeded value by the actual value of the variable. For *Delta* this value represents the change in value comparing with the actual value from the moment the monitor was set. - When the dataType of the variable is integer or decimal, this value represents the difference to be reached to trigger the monitor. - When the dataType of the variable is dateTime the unit of measure will be in seconds. - When the dataType of the variable is string, boolean, OptionList, SequenceList or MemberList, this value is ignored. The monitor will be triggered by every change in the actual value. When a delta monitor is triggered OR when the Charging Station has rebooted, the Charging Station shall set a new momentary value. For *Periodic* and *PeriodicClockAligned* this value represents the interval in seconds.

3.214. (Minor) - Page 343, ocspResult must always be sent when available

The ocspResult was not made optional to allow for omitting the full result when available. It was introduced to make it possible for the CSMS to indicate whether or not the OCSP data was retrieved successfully. In case the OCSP server is down/not found etc, it is not possible to send an OCSP response.

Changed description:

Page	Section	Message/dataType	Field	Old Description	New Description
343	1.46	GetCertificateStatusResponse	ocspResult	Optional. OCSPResponse class as defined in IETF RFC 6960. DER encoded (as defined in IETF RFC 6960), and then base64 encoded.	Optional. OCSPResponse class as defined in IETF RFC 6960. DER encoded (as defined in IETF RFC 6960), and then base64 encoded. **MAY only be omitted when status is not Accepted.**

3.215. (Minor) - Page 344, section 1.47, SetMonitoringResultType, the specification does not specify if the VariableMonitorId's should be unique

The description of the *id* field in the *SetMonitoringResultType* needs to specify that the id's should be unique.

Old Description	Optional. Id given to the Monitor by the Charging Station. The Id is only returned when status is accepted.
New Description	Optional. Id given to the **VariableMonitor** by the Charging Station. The Id is only returned when status is accepted. **Installed VariableMonitors should have unique id's but the id's of removed monitors MAY be reused.**

3.216. (Major) - Page 346, section 1.50, SignedMeterValueType, unclear how to use this DataType and what data to put in the fields

The naming of the fields are unclear. MeterValueSignature for example needs to contain the signed meter data, but the name suggests it only contains the signature. Also a new field needs to be included to provide a public key. When the public key will be included this needs to be configured with a new configuration variable. To make it easier to comprehend the changes, the old fields will be removed and replaced by the correct ones.

Message/dataType	Field	Type	Card.	Description	Action
SignedMeterValueType	meterValueSignature	string[0..2500]	1..1	Required. Digital signature of the meter value.	removed
SignedMeterValueType	signatureMethod	SignatureMethodEnumType	1..1	Required. Method used to create the digital signature.	removed
SignedMeterValueType	encodingMethod	EncodingMethodEnumType	1..1	Required. Method used to encode the meter values before applying the digital signature algorithm.	removed
SignedMeterValueType	encodedMeterValue	string[0..512]	1..1	Required. Meter values as they were encoded before applying the digital signature algorithm.	removed
SignedMeterValueType	SignedMeterData	string[0..2500]	1..1	Required. Base64 encoded, contains the signed data which might contain more then just the meter value. It can contain information like timestamps, reference to a customer etc.	added
SignedMeterValueType	signingMethod	string[0..50]	1..1	Required. Method used to create the digital signature.	added
SignedMeterValueType	encodingMethod	string[0..50]	1..1	Required. Method used to encode the meter values before applying the digital signature algorithm.	added
SignedMeterValueType	publicKey	string[0..2500]	0..1	Optional. Base64 encoded, sending depends on configuration variable *PublicKeyWithSignedMeterValue*.	added

New Schema files are available.

Configuration variable 'PublicKeyWithSignedMeterValue':

Required	no		
Component	**componentName**	OCPPCommCtrlr	
Variable	**variableName**	PublicKeyWithSignedMeterValue	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	OptionList
		valueList	Never,OncePerTransaction,EveryMeterValue
Description	This Configuration Variable can be used to configure whether a public key needs to be send with a signed meter value.		

3.217. (Minor) - Page 347, The fields *persistence* and *constant* in *VariableAttributeType* should be optional

Most variables will be "NOT Constant" and "NOT persistent", so making them optional with default when omitted *false* will safe a lot of data.

Page	Section	Message/dataType	Field	Old Card.	New Card.
347	1.53	VariableAttributeType	persistence	1..1	0..1
347	1.53	VariableAttributeType	constant	1..1	0..1

The name *persistence* is not a correct naming for a boolean. It should be called *persistent*.

Page	Section	Message/dataType	Old Field Name	New Field Name
347	1.53	VariableAttributeType	persistence	persistent

New Schema files are available.

3.218. (Minor) - Page 347, Section 1.53, value is required but should be optional

When the mutability is set to 'WriteOnly', the Charging Station is not able to return a value. So the field needs to be optional.

Changed cardinality AND Description:

Message/dataType	Field	Old Card.	New Card.	Old Description	New Description
VariableAttributeType	value	1..1	0..1	Required. Value of the attribute. The Configuration Variable ValueSize can be used to limit the VariableCharacteristicsType.ValueList and VariableAttributeType.Value. The max size of these values will always remain equal. The default max size is set to 1000.	**Optional. Value of the attribute. May only be omitted when mutability is set to 'WriteOnly'.** The Configuration Variable ValueSize can be used to limit the VariableCharacteristicsType.ValueList and VariableAttributeType.Value. The max size of these values will always remain equal. The default max size is set to 1000.

3.219. (Medium) - Page 349, The use of the dataType 'string' by the *name* and *instance* fields of the *ComponentType* and the *VariableType* allows for a too wide range of characters

The allowed values for the *name* and *instance* fields of the *ComponentType* and the *VariableType* are defined in the appendix. So the limited character set specified by the identifierString will suffice for these fields.

Page	Section	Message/dataType	Field	Old Type	New Type
349	1.56	VariableType	name	string[0..50]	identifierString[0..50]
349	1.56	VariableType	instance	string[0..50]	identifierString[0..50]
334	1.16	ComponentType	name	string[0..50]	identifierString[0..50]
334	1.16	ComponentType	instance	string[0..50]	identifierString[0..50]

3.220. (Major) - Page 353, InstallCertificateResponse incorrectly linked to CertificateStatusEnumType

A new EnumType needs to be used by InstallCertificateResponse instead of the current one. The current EnumType will only be used by *AuthorizeResponse*, so the name 'CertificateStatusEnumType' is not correct anymore.

New EnumType 'InstallCertificateStatusEnumType':

Value	Description
Accepted	The installation of the certificate succeeded.
Failed	The certificate is valid and correct, but there is another reason the installation did not succeed.
Rejected	The certificate is invalid and/or incorrect OR the CSO tries to install more certificates than allowed.

Renamed EnumType:

Page	Section	Old Name	New Name
353	2.9	CertificateStatusEnumType	AuthorizeCertificateStatusEnumType

New Schema files are available.

3.221. (Minor) - Page 353, Typo in one of the Enum values of CertificateUseEnumType

Changed Enum value:

Page	Section	Message/dataType	Old Enum value	New Enum value
353	2.10	CertificateUseEnumType	V2GRootCertficate	V2GRootCertificate

New Schema files are available.

3.222. (Minor) - Page 354, Section 2.14, ChargingProfilePurposeEnumType, Incorrect description 'ChargingStationExternalConstraints'

It needs to be possible to set external constraints on the EVSE level.

Changed description:

Old Description	Additional constraints that will be incorporated into a local power schedule. Only valid for a Charging Station. Therefore evse.Id MUST be 0 in the SetChargingProfileRequest message.
New Description	Additional constraints that will be incorporated into a local power schedule.

3.223. (Major) - Page 355, section 2.17, ChargingStateEnumType, Missing charging state

When a transaction is ongoing and driver unplugs the cable, no new *chargingState* can be reported in the next transactionEvent. A new *chargingState* is needed for when there is no communication between EV and EVSE.

Added Enum value:

Enum value	Description
Idle	There is no connection between EV and EVSE.

3.223.1. Page 346, Section 1.51. TransactionType, The field 'chargingState' has an incorrect description

The chargingState 'Idle' has been added. This caused the description of the chargingState field in the TransactionEventRequest message to become incorrect.

Changed description:

Old Description	Optional. Current charging state, is required when state has changed. Omitted when there is no communication between EVSE and EV, because no cable is plugged in.
New Description	Optional. **This field contains the** current charging state. **This field is only required when the chargingState has changed.**

New Schema files are available.

3.224. (Minor) - Page 355, ChargingStateEnumType has an incorrect description

This is the state of the charging process, not the trigger reason.

Changed description:

Page	Section	Message/dataType	Old description	New description
355	2.17	ChargingStateEnumType	Reason that triggered a transactionEventRequest(eventType=Updated) to be sent.	The state of the charging process.

3.225. (Minor) - Page 355, section 2.17, ChargingStateEnumType, description *Charging* unclear

The description is missing the fact that the energy is flowing. Closing the contactor of the Connector is not enough.

Changed description:

Old	When the contactor of a Connector closes, allowing the vehicle to charge.
New	The contactor of the Connector is closed and energy is flowing to between EVSE and EV.

3.226. (Minor) - Page 355, section 2.17, ChargingStateEnumType, Enum value *EVDetected* is not the correct naming for its use

The Enum value EVDetected is also used to indicate the state after the energy transfer has stopped. So EVDetected does not cover all its uses. The description also needs to be clearer.

Changed Enum value:

Version	Enum value	Description
Old	EVDetected	EV is detected. Cable is plugged in and there is communication between EV and EVSE.
New	EVConnected	There is a connection between EV and EVSE, in case the protocol used between EV and the Charging Station can detect a connection, the protocol needs to detect this for the state to become active. The connection can either be wired or wireless.

New Schema files are available.

3.227. (Major) - Page 359, 2.32 EnergyTransferModeEnumType

The EnergyTransferModeEnumType contained values that are not relevant for a CSMS. In addition, it was missing an option for two phase AC charging.

New definition of EnergyTransferModeEnumType is as follows:

Value	Description
AC_single_phase	AC single phase charging according to IEC 62196.
AC_two_phase	AC two phase charging according to IEC 62196.
AC_three_phase	AC three phase charging according to IEC 62196.
DC	DC charging.

3.228. (Minor) - Page 360, Section 2.35, GenericDeviceModelStatusEnumType, Missing status for an empty result set

The use cases Get Custom Report and Get Monitoring Report may select an empty report set. This is unlikely to happen with Get Base Report, but it should also be considered there. When this happens the Charging Station will not send report information in one or more NotifyReportRequest, NotifyMonitoringRequest messages, because they do not allow empty results.

Added Enum value 'GenericDeviceModelStatusEnumType':

Enum value	Description
EmptyResultSet	If the combination of received criteria result in an empty result set.

New requirements:

ID	Precondition	Requirement definition
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).
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).

3.229. (Medium) - Page 365, SetMonitoringBase, the definition of monitoringBase 'None' is not correct anymore

After defining the set of monitor/notification groups, the definition of 'None' should be changed. Instead of disabling all monitoring, the Charging Station will clear all custom monitors and disable all pre-configured monitors, leaving only the hard-wired monitors active. Because of this a rename was also in order.

Page	Section	Message/dataType	Value	Action	Description
365	2.53	MonitoringBaseEnumType	None	removed	Disable all monitoring.
365	2.53	MonitoringBaseEnumType	HardWiredOnly	added	Clears all custom monitors and disables all pre-configured monitors, leaving only the hard-wired monitors active.

New Schema files are available.

3.230. (Medium) - Page 365, Section 2.51, MessageTriggerEnumType, Missing enum value for combined certificate

As having a certificate that is possible to be used for both roles; ChargingStationCertificate and V2GCertificate. It needs to be possible to trigger the update of such a combined certificate. An enum value needs to be added to MessageTriggerEnumType for the combined certificate.

Added Enum value:

Page	section	Message/dataType	Value	Description
365	2.51	MessageTriggerEnumType	SignCombinedCertificate	To trigger a SignCertificate with typeOfCertificate: ChargingStationCertificate AND V2GCertificate

New Schema files are available.

3.231. (Minor) - Page 365, SetMonitoringBase, the definition of monitoringBase 'All' is not correct anymore

After defining the set of monitor/notification groups, the definition of 'All' should be changed. Instead of just enabling all (preconfigured) monitoring, the Charging Station will also clear all custom monitors.

Page	Section	Message/dataType	Value	Old description	New description
365	2.53	MonitoringBaseEnumType	All	Activate all preconfigured monitoring.	Clears all custom monitors and enables all pre-configured monitors.

New Schema files are available.

3.232. (Medium) - Page 368, Incorrect enum value *UnlockCommand* in *ReasonEnumType*

It is not possible/allowed to stop a transaction using the UnlockConnectorRequest. RequestStopTransactionRequest must be used for this. So the enum value *UnlockCommand* needs to be removed.

Page	Section	Message/dataType	Value	Action
368	2.64	ReasonEnumType	UnlockCommand	removed

New Schema files are available.

3.233. (Minor) - Page 369, section 2.66, RegistrationStatusEnumType, unclear when to use *Rejected*

The description of the *Rejected* status describes the CSMS will respond with *Rejected* when it does not know the Charging Station id. However the OCPP-J specification states that when an unknown Charging Station tries to setup a WebSocket connection it SHOULD abort the connection. To avoid confusion it was decided to change "SHOULD abort the connection" to "MAY abort the connection". Then the implementer may decide whether to abort the connection, but when receiving a BootNotificationRequest from a Charging Station with an unknown id, then the CSMS MUST respond with status *Rejected*.

Changed description:

Old description	Charging Station is not accepted by CSMS. This may happen when the Charging Station id is not known by CSMS.
New description	Charging Station is not accepted by CSMS. This may happen when the Charging Station id is not known by CSMS **and the connection was not aborted during the connection setup.**

3.234. (Minor) - Page 369, Datatypes 2.66, RegistrationStatusEnumType: confusing description

An unknown Charging Station will not receive a BootNotification Rejected, but an HTTP 404, as specified in part 4 (the OCPP-J specification).

Therefore another example in the description for the value `Rejected` is better.

Old text	Charging Station is not accepted by CSMS. This may happen when the Charging Station id is not known by CSMS.
New text	Charging Station is not accepted by CSMS. This may happen when the **imsi** is not known by CSMS.

3.235. (Medium) - Page 373, section 2.80, TriggerReasonEnumType

The ResetRequest message is able to cause a transaction to end and while this is included in the stoppedReason, it is not in the triggerReason. Like for the message UnlockConnectorRequest, which causes a TransactionEventRequest with the triggerReason *UnlockCommand*, we need a triggerReason for ResetRequest.

Added Enum value:

Page	section	Message/dataType	Value	Description
372	2.80	TriggerReasonEnumType	ResetCommand	CSMS sent a Reset Charging Station command.

The description of the following enumeration value has changed:

Old	**ChargingRateChanged**	Rate of charging changed by more than MaxLimitChangedSkipPercentage, or next period in charging schedule
New	**ChargingRateChanged**	Rate of charging changed by more than *LimitChangeSignificance*.

New Schema files are available.

3.236. (Medium) - Page 373, section 2.81, UnlockStatusEnumType, missing enum values

The specification describes that it is not possible to unlock a connector on which there is still an ongoing transaction after receiving a UnlockConnectorRequest from the CSMS. But the Charging Station should only reject the message when the ongoing transaction is still authorized. Also the enum values *OngoingTransaction* and *UnknownConnector* are missing. But *OngoingTransaction* needs to be renamed to *OngoingAuthorizedTransaction*, to make it more clear.

New EnumType values:

Message/dataType	Value	Description
UnlockStatusEnumType	OngoingAuthorizedTransaction	There is still an authorized transaction ongoing, so it is not allowed to unlock the connector.
UnlockStatusEnumType	UnknownConnector	The specified connector is not known by the Charging Station.

New Schema files are available.

Changed requirement:

Version	Id	Precondition	Requirement definition
Old	F05.FR.02	F05.FR.01 AND There is a transaction ongoing on the specified connector.	The Charging Station SHALL NOT try to unlock the connector (or stop the transaction) but use the status: OngoingTransaction in the UnlockConnectorResponse.
New	F05.FR.02	F05.FR.01 AND There is **an authorized** transaction ongoing on the specified connector.	The Charging Station SHALL NOT try to unlock the connector (or stop the transaction) but use the status: **OngoingAuthorizedTransaction** in the UnlockConnectorResponse.

3.237. (Medium) - Page 373, section 2.80, TriggerReasonEnumType, missing trigger reason for signed data

There is a TxStartStopPoint value named *DataSigned*, which indicates to start a transaction when some signed data has been received. But no trigger reason exists for this situation, so one needs to be added.

New EnumType value:

Message/dataType	Value	Description
TriggerReasonEnumType	SignedDataReceived	Signed data is received from the energy meter.

New Schema files are available.

3.238. (Minor) - Page 373, TriggerReasonEnumType has an incorrect description

The trigger reason must always be given, not only with TransactionEventRequest(eventType=Updated).

Changed description:

Page	Section	Message/dataType	Old description	New description
373	2.80	TriggerReasonEnumType	Reason that triggered a transactionEventRequest(eventType=Updated) to be sent.	Reason that triggered a transactionEventRequest.

3.239. (Medium) - Page 377, ISO 15118 plug and charge when the Charging Station is offline

OCPP did not specify anything regarding ISO 15118 plug and charge when the Charging Station is offline. The following changes will be made to manage the plug and charge behavior of the Charging Station.

Message/dataType	Field	Type	Card.	Description	Action
AuthorizeRequest	certificate	string[0..5500]	0..1	Optional. The X.509 certificated presented by EV and encoded in PEM format.	added

The *ISO15118Ctrlr* needs to be added to the standardized component list.

Added Component 'ISO15118Ctrlr':

Two standardized configuration variables need to be added, which allows the CSO to configure the Charging Stations behavior.

Added Configuration variable 'CentralContractValidationAllowed':

Required	no		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	CentralContractValidationAllowed	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	Boolean
Description	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.		

Added Configuration variable 'ContractValidationOffline':

Required	yes		
Component	**componentName**	ISO15118Ctrlr	
Variable	**variableName**	ContractValidationOffline	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	Boolean
Description	If this variable is *true*, then Charging Station will try to validate a contract certificate when it is offline.		

Changed requirements:

Version	Id	Precondition	Requirement definition
Old	C07.FR.01		The Charging Station SHALL send the identification to the CSMS for validation.
New	C07.FR.01	**When Charging Station is online**	The Charging Station SHALL send **an AuthorizeRequest** to the CSMS for validation.
Old	C07.FR.02	When authorization with certificates is used	The AuthorizeRequest SHALL contain the eMAID and data needed for an OCSP request with regards to the contract certificate and certificate chain.

Version	Id	Precondition	Requirement definition
New	C07.FR.02	**C07.FR.01**	The AuthorizeRequest SHALL contain the eMAID and data needed for an OCSP request with regards to the contract certificate and certificate chain.

New requirements:

ID	Precondition	Requirement definition
C07.FR.06	C07.FR.01 AND If Charging Station is not able to validate a contract certificate, because it does not have the associated root certificate AND `CentralContractValidationAllowed` is *true*	The Charging Station SHALL pass the contract certificate to the CSMS in *certificate* attribute (in PEM format) of AuthorizeRequest for validation by CSMS.
C07.FR.07	When Charging Station is offline AND `ContractValidationOffline` is *false*	The Charging Station SHALL NOT allow charging.
C07.FR.08	When Charging Station is offline AND `ContractValidationOffline` is *true*	The Charging Station SHALL try to validate the contract certificate locally.
C07.FR.09	C07.FR.08 AND Contract certificate is valid AND `LocalAuthorizeOffline` is *true*	The Charging Station SHALL lookup the eMAID in Local Authorization List or Authorization Cache.
C07.FR.10	C07.FR.09 AND eMAID found in Local Authorization List	The Charging Station SHALL behave according to use case C13.
C07.FR.11	C07.FR.09 AND eMAID found in Authorization Cache	The Charging Station SHALL behave according to use case C12.
C07.FR.12	C07.FR.09 AND eMAID is not found AND `OfflineTxForUnknownIdEnabled` = *true*	The Charging Station SHALL allow charging according to use case C14.

3.240. (Medium) - Page 377, Charging Station unable to set the subject field in the client certificate

The subject field in the client certificate should contain the organization name of the CSO or an organization trusted by the CSO, but the Charging Station has no way of knowing this information. To solve this the organization name should be made configurable over OCPP by creating a new Configuration Variable.

New configuration variable 'OrganizationName':

Required	yes		
Component	**componentName**	SecurityCtrlr	
Variable	**variableName**	OrganizationName	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	string
Description	This configuration variable is used to set the organization name of the CSO or an organization trusted by the CSO. This organization name is used to specify the subject field in the client certificate.		

3.241. (Minor) - Page 378, Referenced Components and Variables, Section 2

To the end of this paragraph, before start of section 2.1, add the following note:

NOTE See *'OCPP 2.0 Part 4 - JSON over Websockets implementation guide'* for a number of Configuration Variables that are specific to controlling the JSON/Websocket behavior.

3.242. (Minor) - Page 378, Section 2.1.1, ConnectorPhaseRotation

There are several issue regarding the current way in which the ConnectorPhaseRotation configuration variable is defined.

1. PhaseRotation is currently defined as a variable for the Connector component, however this variable should also be applicable for other physical components.
2. PhaseRotation is currently unable to indicate how a single or two phase(s) are connected.

Changed componentName and dataType configuration variable 'PhaseRotation':

Old componentName	New componentName	Old dataType	New dataType
Connector	*	OptionList	String

Changed description configuration variable 'PhaseRotation':

Old description	**The phase rotation of a Component relative to its parent Component.** `NotApplicable` **(for Single phase or DC Charging Stations)** `Unknown` **(not (yet) known)** `RST` **(Standard Reference Phasing)** `RTS` **(Reversed Reference Phasing)** `SRT` **(Reversed 240 degree rotation)** `STR` **(Standard 120 degree rotation)** `TRS` **(Standard 240 degree rotation)** `TSR` **(Reversed 120 degree rotation)** **R can be identified as phase 1 (L1), S as phase 2 (L2), T as phase 3 (L3).**
New description	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. Certain measurands, like voltage and current, are reported with a phase relative to the grid connection. In order to support this, all components in the chain from Connector to ElectricalFeed need to have a value for PhaseRotation. Some examples: "" (unknown) "RST" (Standard Reference Phasing) "RTS" (Reversed Reference Phasing) "SRT" (Reversed 240 degree rotation) "STR" (Standard 120 degree rotation) "TRS" (Standard 240 degree rotation) "TSR" (Reversed 120 degree rotation) "RSx" (Two phases connected) "Rxx" (One phase connected)

3.243. (Minor) - Page 380, section 2.1.11, UnlockOnEVSideDisconnect is read-only for fixed cables

Change the **mutability** for UnlockOnEVSideDisconnect to "ReadWrite/ReadOnly".

Add the following sentence to the **Description**: "For an EVSE with a fixed cable the mutability SHALL be ReadOnly and the actual

value SHALL be false."

3.244. (Minor) - page 384, section 2.2.1: Description of BasicAuthPassword incorrect

The description of BasicAuthPassword incorrectly states that this option does not have to be present when certificates are used. But that is incorrect: It does not have to be present when client side certificates are used.

Old text	If certificates are used, this option does not have to be present.
New text	This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented.

3.245. (Minor) - Page 385, section 2.62, PublishFirmwareStatusEnumType, incorrect description *PublishFailed*

The current description of *PublishFailed* is a copy of the description of *InstallationFailed*.

Old description	Failure end state. Installation of new firmware has failed.
New description	**Publishing the new firmware has failed.**

The description of *Published* is also a bit unclear.

Old description	Successful end state. Firmware is being published.
New description	**The firmware has been successfully published.**

3.246. (Minor) - Page 385, 2.3.1, Meaning of Enabled variable on AuthCtrlr

The description of the variable `Enabled` does not sufficiently explain the purpose of it. The following changes shall be made to this paragraph to improve it.

Old **Description**	If this variable reports a value of true, Authorization is enabled.
New **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`.

3.247. (Major) - Page 388, SendLocalList configuration variables: instance removed

VariableInstances where accidentally added to the variables below. No instance should be used.

Variable `ItemsPerMessageSendLocalList`	Remove **variableInstance** SendLocalList
Variable `BytesPerMessageSendLocalList`	Remove **variableInstance** SendLocalList

3.248. (Medium) - Page 388/389, Referenced Components and Variables, duplicate variable

The Variables; ChargingBeforeAcceptedEnabled and TxBeforeAcceptedEnabled are identical, so one needs to be removed. TxBeforeAcceptedEnabled is referenced in the specification, so this is the one that will remain. However the description from ChargingBeforeAcceptedEnabled is more clear, so that will be used instead.

Removed Standard configuration variable 'ChargingBeforeAcceptedEnabled'

Changed description 'TxBeforeAcceptedEnabled':

Old Description	If this Charging Station is allowed to start transaction before being accepted by a CSMS. See: Transactions before being accepted by a CSMS.
New Description	**With this configuration variable the Charging Station can be configured to allow charging before having received a BootNotificationResponse with RegistrationStatus: Accepted.** See: Transactions before being accepted by a CSMS.

3.249. (Major) - Page 389, TxStartPoint and TxStopPoint should be a *MemberList*, not an *OptionList*

This is an error in the specification. The TxStartPoint and TxStopPoint were designed to be a *MemberList*.

Page	Section	ComponentName	VariableName	Old DataType	New DataType
389	2.6.5	TxCtrlr	TxStartPoint	OptionList	MemberList
390	2.6.6	TxCtrlr	TxStopPoint	OptionList	MemberList

3.250. (Medium) - Page 389, The specification is unclear about whether changing the TxStartPoints/TxStopPoints has influence on ongoing transactions

Changed descriptions:

Page	Section	Message/dataType	Old Description	New Description
389	2.6.5	TxStartPoint	Defines when the Charging Station starts a new transaction: first TransactionEventRequest: eventType = Started. When any event in the given list occurs, the Charging Station SHALL start a transaction. The Charging Station SHALL only send the Started event once for every transaction. It is advised to put all events that should be part of a transaction in the list, in case the start event never occurs. Because the possible events don't always have to come in the same order it is possible to provide a list of events. Which ever comes first will then cause a transaction to be started. For example: EVConnected, Authorized would mean that a transaction is started when an EV is detected (Cable is connected), or when an EV Driver swipes his RFID card en the CSMS successfully authorizes the ID for charging.	Defines when the Charging Station starts a new transaction: first TransactionEventRequest: eventType = Started. When any event in the given list occurs, the Charging Station SHALL start a transaction. The Charging Station SHALL only send the Started event once for every transaction. It is advised to put all events that should be part of a transaction in the list, in case the start event never occurs. Because the possible events don't always have to come in the same order it is possible to provide a list of events. Which ever comes first will then cause a transaction to be started. For example: EVConnected, Authorized would mean that a transaction is started when an EV is detected (Cable is connected), or when an EV Driver swipes his RFID card **and** the CSMS successfully authorizes the ID for charging. **Changing the TxStartPoint does not influence ongoing transactions, only new transactions.**
390	2.6.6	TxStopPoint	Defines when the Charging Station ends a transaction: last TransactionEventRequest: eventType = Ended. When any event in the given list is no longer valid, the Charging Station SHALL end the transaction. The Charging Station SHALL only send the Ended event once for every transaction.	Defines when the Charging Station ends a transaction: last TransactionEventRequest: eventType = Ended. When any event in the given list is no longer valid, the Charging Station SHALL end the transaction. The Charging Station SHALL only send the Ended event once for every transaction. **Adding a value to TxStopPoint that would have already stopped the transaction, causes the transaction to end immediately. If the new value set does not cause the transaction to stop, then the Charging Station will take the new value into account. So when the transaction reaches the new next TxStopPoint the transaction will end.**

3.251. (Minor) - Page 389, Reference Components and Variables, 2.6.3 StopTxOnEVSideDisconnect can be ReadOnly

There may be situations where it is not allowed for the CSMS to set `StopTxOnEVSideDisconnect` to true.

In the table for `StopTxOnEVSideDisconnect` change the mutability as follows:

variableAttributes	**Mutability**	ReadWrite **or ReadOnly, depending on Charging Station implementation.**

3.252. (Minor) - Page 391, The descriptions of the configuration variables *SampledDataSignReadings* and *AlignedDataSignReadings* are reversed

SampledDataSignReadings:

Old description	If set to true, the Charging Station SHALL include signed meter values in the SampledValueType in the MeterValuesRequest to the CSMS. When a Charging Station does not support signed meter values it SHALL NOT report this variable.
New description	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.

AlignedDataSignReadings:

Old description	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 description	If set to true, the Charging Station SHALL include signed meter values in the SampledValueType in the MeterValuesRequest to the CSMS. When a Charging Station does not support signed meter values it SHALL NOT report this variable.

3.253. (Medium) Page 391, 2.7.4: `SampledDataTxEndedMeasurands`: changed description

Old description	[..] The Charging Station might report a limited set of these Measurands via VariableCharacteristicsType.valuesList. [..]
New description	[..] The Charging Station **reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable.** [..]

3.254. (Medium) Page 391, 2.7.6: `SampledDataTxStartedMeasurands`: changed description

Old description	[..] The Charging Station might report a limited set of these Measurands via VariableCharacteristicsType.valuesList. [..]
New description	[..] The Charging Station **reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable.** [..]

3.255. (Medium) Page 391, 2.7.67 `SampledDataTxUpdatedMeasurands`: changed description

Old description	[..] The Charging Station might report a limited set of these Measurands via VariableCharacteristicsType.valuesList. [..]
New description	[..] The Charging Station **reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable.** [..]

3.256. (Medium) Page 393, 2.7.11: `AlignedDataMeasurands`: changed description

Old description	[..] The Charging Station might report a limited set of these Measurands via VariableCharacteristicsType.valuesList. [..]
New description	[..] The Charging Station **reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable.** [..]

3.257. (Medium) Page 394, 2.7.15: `AlignedDataTxEndedMeasurands`: changed desription

Old description	[..] The Charging Station might report a limited set of these Measurands via VariableCharacteristicsType.valuesList. [..]
New description	[..] The Charging Station **reports the list of supported Measurands in VariableCharacteristicsType.valuesList of this variable.** [..]

3.258. (Major) Page 396, Controller Components, Add CustomizationCtrlr component

To manage the custom implementations a controller component needs to be added.

Controller Component	**Description**
CustomizationCtrlr	Responsible for configuration relating to custom vendor-specific implementations, using the DataTransfer message and CustomData extensions.

To enable/disable custom vendor-specific implementations a variable needs to be added to the CustomizationCtrlr component.

Configuration variable 'CustomImplementation':

Required	no	
Component	**componentName**	CustomizationCtrlr

Variable	**variableName**	CustomImplementationEnabled	
	variableInstance	<VendorId>	
	variableAttributes	**mutability**	ReadWrite
	variableCharacteristics	**dataType**	boolean
Description	This standard configuration variable can be used to enable/disable custom implementations that the Charging Station supports. It is recommended to first check if the custom behavior is able to be implemented using the device model, otherwise DataTransfer message(s) and/or CustomData fields can be used. The custom DataType can be included to any other DataType or Message. By having set additionalProperties to true for this DataType only, the implementer is able to create a custom data structure within an enclosed space.		

3.259. (Major) - Page 400, 2.13 Charging Infrastructure related variables

This erratum describes the additions to the specification to specify the minimally required components and variables that shall be reported in response to a GetBaseReport(*id*, FullInventory) command.

Add a new section 2.13 to chapter 2 of "Referenced Components and Variables", as follows:

3.259.1. Section 2.13 Charging Infrastructure related

Available

Required	yes		
Components	**componentName**	ChargingStation	
		EVSE	
		Connector	
	evse	* (*for EVSE and Connector*)	
Variable	**variableName**	Available	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	boolean
Description	When *true* the Component exists and is locally configured/wired for use, but may not be (remotely) Enabled. This variable is required on any Component that can be reported by the Charging Station. As a minimum it shall exist on ChargingStation, EVSE and Connector.		
Note	If any other variables are reported for a Component, then reporting **Available** does not add much value and may be omitted. However, the variable needs to exist, because it can be queried for by a GetCustomReport request for all Components that are 'available'.		

AvailabilityState

Required	yes		
Components	**componentName**	ChargingStation	
		EVSE	
	evse	* (*for EVSE*)	
Variable	**variableName**	AvailabilityState	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	optionList
		valuesList	Available, Occupied, Reserved, Unavailable, Faulted
Description	This variable reports current availability state for the ChargingStation and EVSE. If a Connector has its own availability state independent of the EVSE, then this variable may be used to report the Connector's availability state. As such it replicates ConnectorStatus values reported in StatusNotification messages.		

ConnectorType

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".		

SupplyPhases

Required	yes		
Components	**componentName**	ChargingStation	
		EVSE	
		Connector	
	evse	* (*for EVSE and Connector*)	
Variable	**variableName**	SupplyPhases	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	integer
Description	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.		

Power

Required	yes (*maxLimit only*)		
Component	**componentName**	EVSE	
	evse	*	
Variable	**variableName**	Power	
	variableAttributes	**mutability**	ReadOnly
	variableCharacteristics	**dataType**	decimal
		maxLimit	decimal
Description	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.		