ID	PRECONDITION	REQUIREMENT DEFINITION
A05.FR.09	A05.FR.07 AND The Charge Point was able to successfully connect to the Central System	The Central System SHALL NOT allow the Charge Point to connect with a lower security profile anymore.

M03 - Retrieve list of available certificates from a Charge Point

Table 24. M03 - Retrieve list of available certificates from a Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Retrieve list of available certificates from a Charge Point
2	**ID**	M03 *(OCPP 2.0.1)*
3	**Objective(s)**	To enable the Central System to retrieve a list of available certificates from a Charge Point.
4	**Description**	To facilitate the management of the Charge Point's installed certificates, a method of retrieving the installed certificates is provided. The Central System requests the Charge Point to send a list of installed certificates
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System requests the Charge Point to send a list of installed certificates by sending a GetInstalledCertificateIds.req **2.** The Charge Point responds with a GetInstalledCertificateIds.conf
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The Central System received a list of installed certificates

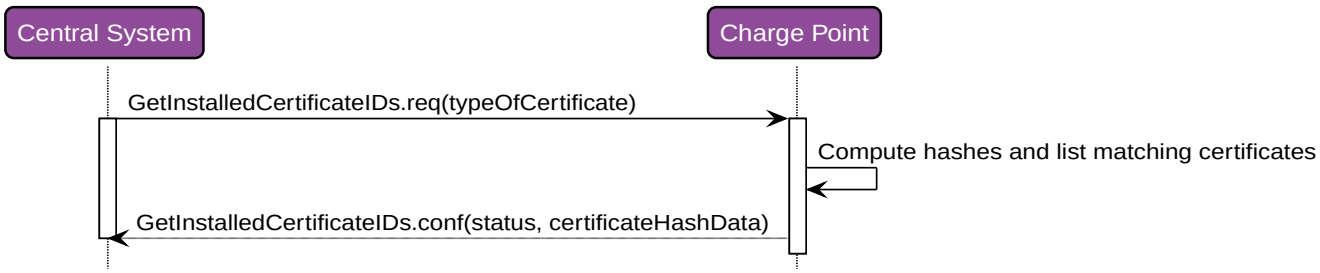

Figure 8. Retrieve list of available certificates from a Charge Point

7	**Error handling**	n/a
8	**Remark(s)**	For installing the Charge Point Certificate, see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point.

M03 - Retrieve list of available certificates from a Charge Point - Requirements

Table 25. M03 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
M03.FR.01	After receiving a GetInstalledCertificateIds.req	The Charge Point SHALL respond with a GetInstalledCertificateIds.conf.
M03.FR.02	M03.FR.01 AND No certificate matching *certificateType* was found	The Charge Point SHALL indicate this by setting *status* in the GetInstalledCertificateIds.conf to *NotFound*.
M03.FR.03	M03.FR.01 AND A certificate matching *certificateType* was found	The Charge Point SHALL indicate this by setting *status* in the GetInstalledCertificateIds.conf to *Accepted*.
M03.FR.04	M03.FR.03	The Charge Point SHALL include the hash data for each matching installed certificate in the GetInstalledCertificateIds.conf.

M04 - Delete a specific certificate from a Charge Point

Table 26. M04 - Delete a specific certificate from a Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Delete a specific certificate from a Charge Point
2	**ID**	M04 *(OCPP 2.0.1)*
3	**Objective(s)**	To enable the Central System to request the Charge Point to delete an installed certificate.
4	**Description**	To facilitate the management of the Charge Point's installed certificates, a method of deleting an installed certificate is provided. The Central System requests the Charge Point to delete a specific certificate.
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System requests the Charge Point to delete an installed certificate by sending a DeleteCertificate.req. **2.** The Charge Point responds with a DeleteCertificate.conf.
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The requested certificate was deleted from the Charge Point.

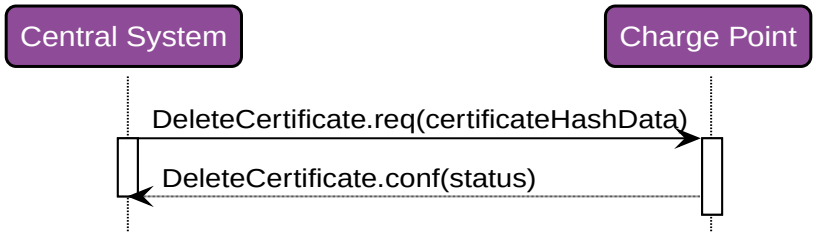

Figure 9. Delete Installed Certificate

7	**Error handling**	n/a
8	**Remark(s)**	For installing the Charge Point Certificate, see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point. It is possible to delete the last (every) installed CentralSystemRootCertificates, when all CentralSystemRootCertificates are deleted, the Charge Point cannot validate Central System Certificates, so it will not be able to connect to a Central System. Before a Central System would ever send a DeleteCertificate.req that would delete the last/all CentralSystemRootCertificates the Central System is ADVICED 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 Charge Point.

M04 - Delete a specific certificate from a Charge Point - Requirements

Table 27. M04 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
M04.FR.01	After receiving a DeleteCertificate.req	The Charge Point SHALL respond with a DeleteCertificate.conf.
M04.FR.02	M04.FR.01 AND The requested certificate was found	The Charge Point SHALL delete it, and indicate success by setting 'status' to 'Success' in the DeleteCertificate.conf.
M04.FR.03	M04.FR.01 AND The deletion fails	The Charge Point SHALL indicate failure by setting 'status' to 'Failed' in the DeleteCertificate.conf.
M04.FR.04	M04.FR.01 AND The requested certificate was not found	The Charge Point SHALL indicate failure by setting 'status' to 'NotFound' in the DeleteCertificate.conf.
M04.FR.05		Deletion of the *Charge Point Certificate* SHALL NOT be possible via a DeleteCertificate.req.
M04.FR.06	M04.FR.01 AND Certificate to delete is a CentralSystemRootCertificate AND This CentralSystemRootCertificate is currently in use for validation of the connection the the Central System	The Charge Point SHALL reject the request by setting 'status' to 'Failed' in the DeleteCertificate.conf.

ID	PRECONDITION	REQUIREMENT DEFINITION
M04.FR.07	When deleting a certificate	The Central System SHALL use the *hashAlgorithm*, which was used to install the certificate.

M05 - Install CA certificate in a Charge Point

Table 28. M05 - Install CA certificate in a Charge Point

NO.	TYPE	DESCRIPTION
1	**Name**	Install CA certificate in a Charge Point
2	**ID**	M05 *(OCPP 2.0.1)*
3	**Objective(s)**	To facilitate the management of the Charge Point's installed certificates, a method to install a new CA certificate.
4	**Description**	The Central System requests the Charge Point to install a new Central System root certificate or Manufacturer root certificate.
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System requests the Charge Point to install a new certificate by sending an InstallCertificate.req. **2.** The Charge Point responds with an InstallCertificate.conf.
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The new certificate was installed in the Charge Point trust store.

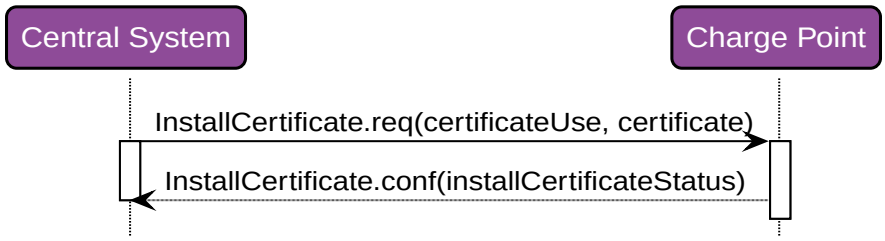

Figure 10. Install CA certificate in a Charge Point

7	**Error handling**	n/a

8	**Remark(s)**	Even though the messages CertificateSigned.req (see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point) and InstallCertificate.req (use case M05) are both used to send certificates, their purposes are different. CertificateSigned.req is used to return the the Charge Points *own* public certificate signed by a Certificate Authority. InstallCertificate.req is used to install Root certificates. For installing the Charge Point Certificate, see use cases A02 - Update Charge Point Certificate by request of Central System and A03 - Update Charge Point Certificate initiated by the Charge Point. It is allowed to have multiple certificates of the same type installed.

M05 - Install CA certificate in a Charge Point - Requirements

Table 29. M05 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION
M05.FR.01	After receiving an InstallCertificate.req	The Charge Point SHALL attempt to install the certificate and respond with an InstallCertificate.conf.
M05.FR.02	M05.FR.01 AND The installation was successful	The Charge Point SHALL indicate success by setting 'status' to 'Accepted' in the InstallCertificate.conf.
M05.FR.03	M05.FR.01 AND Current amount of install certificates >= `CertificateStoreMaxLength`	The Charge Point SHALL indicate failure (no more space to install more certificates) by setting 'status' to 'Rejected' in the InstallCertificate.conf
M05.FR.04	M05.FR.01 AND The installation failed	The Charge Point SHALL indicate failure by setting 'status' to 'Failed' in the InstallCertificate.conf.
M05.FR.06	M05.FR.01 AND The certificate is invalid and/or incorrect.	The Charge Point SHALL indicate rejection by setting 'status' to 'Rejected' in the InstallCertificate.conf.
M05.FR.08	When `AdditionalRootCertificateCheck` is true	Only one certificate (plus a temporarily fallback certificate) of certificateType CentralSystemRootCertificate is allowed to be installed at a time.
M05.FR.09	When `AdditionalRootCertificateCheck` is true AND installing a new certificate of certificateType CentralSystemRootCertificate	The new Central System Root certificate SHALL replace the old Central System Root certificate AND the new Root Certificate MUST be signed by the old Root Certificate it is replacing
M05.FR.10	M05.FR.09 AND the new Central System Root certificate is NOT signed by the old Central System Root certificate	The Charge Point SHALL NOT install the new Central System Root Certificate and respond with status *Rejected*.

ID	PRECONDITION	REQUIREMENT DEFINITION
M05.FR.11	M05.FR.09 AND the new Central System Root certificate is signed by the old Central System Root certificate	The Charge Point SHALL install the new Central System Root Certificate AND temporarily keep the old Central System Root certificate as a fallback certificate AND respond with status *Accepted*
M05.FR.12	M05.FR.11 AND the Charge Point successfully connected to the Central System using the new Central System Root certificate	The Charge Point SHALL remove the old Central System Root (fallback) certificate.
M05.FR.13	M05.FR.11 AND The Charge Point is attempting to reconnect to the Central System, but determines that the server certificate provided by the Central System is invalid when using the new Central System Root certificate to verify it	The Charge Point SHALL try to use the old Central System Root (fallback) certificate to verify the server certificate.

3. Security events/logging

A04 - Security Event Notification

Table 30. A04 - Security Event Notification

NO.	TYPE	DESCRIPTION
1	**Name**	Security Event Notification
2	**ID**	A04 *(OCPP 2.0.1)*
3	**Objective(s)**	To inform the Central System of critical security events.
4	**Description**	This use case allows the Charge Point to immediately inform the Central System of changes in the system security.
	Actors	Central System, Charge Point
	Scenario description	**1.** A *critical* security event happens. **2.** The Charge Point sends a SecurityEventNotification.req to the Central System. **3.** The Central System responds with SecurityEventNotification.conf to the Charge Point.
5	**Prerequisite(s)**	n/a
6	**Postcondition(s)**	The Charge Point *successfully* informed the Central System of critical security events by sending a SecurityEventNotification.req to the Central System.

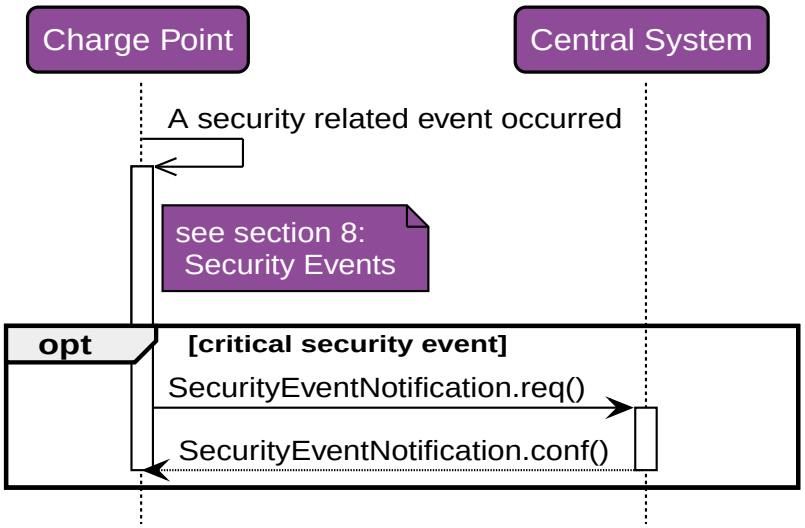

Figure 11. Security Event Notification

7	**Error handling**	n/a

8	**Remark(s)**	A list of security related events and their 'criticality' is provided at Security Events

A04 - Security Event Notification - Requirements

Table 31. A04 - Security Event Notification - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
A04.FR.01	When a *critical* security event happens	The Charge Point SHALL inform the Central System of the security events by sending a SecurityEventNotification.req, to the Central System.	
A04.FR.02	A04.FR.01 AND the Charge Point is disconnected.	Security event notifications MUST be queued with a guaranteed delivery at the Central System.	
A04.FR.03	A04.FR.01	The Central System SHALL confirm the receipt of the notification using the SecurityEventNotification.conf message.	
A04.FR.04	When a security event happens (also none-critical)	The Charge Point SHALL store the security event in a security log.	It is recommended to implement this in a rolling format.

N01 - Retrieve Log Information

Table 32. N01 - Retrieve Log Information

NO.	TYPE	DESCRIPTION
1	**Name**	Retrieve Log
2	**ID**	N01 *(OCPP 2.0.1)*
3	**Objective(s)**	To enable the Central System retrieving of log information from a Charge Point.
4	**Description**	This use case covers the functionality of getting log information from a Charge Point. The Central System can request a Charge Point to upload a file with log information to a given location (URL). The format of this log file is not prescribed. The Charge Point uploads a log file and gives information about the status of the upload by sending status notifications to the Central System.
	Actors	Charge Point, Central System

NO.	TYPE	DESCRIPTION
	Scenario description	**1.** The Central System sends a GetLog.req to the Charge Point. **2.** The Charge Point responds with a GetLog.conf. **3.** The Charge Point sends a LogStatusNotification.req with the status Uploading **4.** The Central System responds with a LogStatusNotification.conf acknowledging the status update request. **5.** Uploading of the diagnostics files. **6.** The Charge Point sends LogStatusNotification.req with the status *Uploaded*. **7.** The Central System responds with LogStatusNotification.conf, acknowledging the status update request. **8.** The Charge Point returns to *Idle* status.
5	**Prerequisite(s)**	- Requested information (either DiagnosticsLog or SecurityLog) is available for upload. - URL to upload file to is reachable and exists.
6	**Postcondition(s)**	**Successful postcondition:** Log file *Successfully* uploaded. **Failure postcondition:** Log file *not Successfully* uploaded and *Failed*.

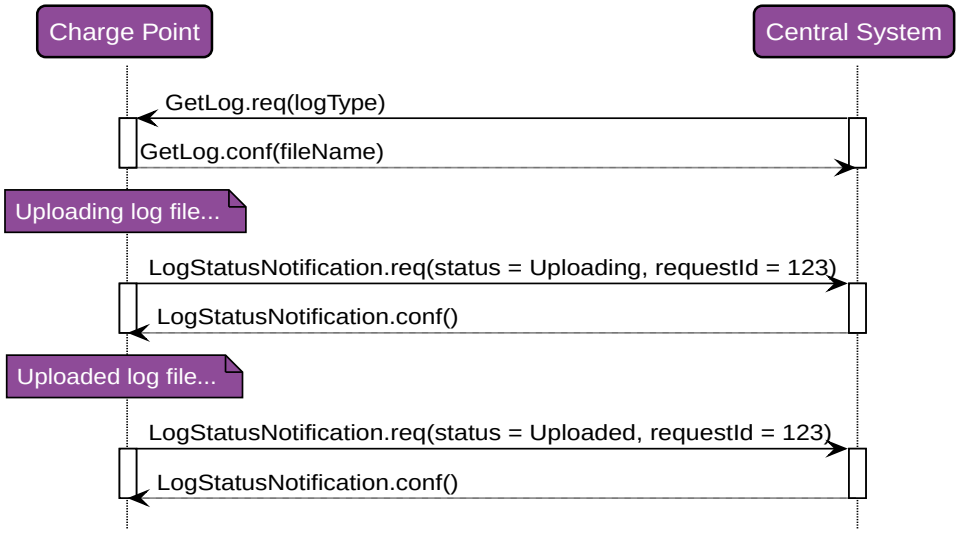

Figure 12. Sequence Diagram: Get Security Log

7	**Error handling**	When the upload fails, and the transfer protocol supports "resume", it is recommended that the Charge Point tries "resume" before aborting the upload.

8	**Remark(s)**	When a Charge Point is requested to upload a log file, the Central System supplies in the request an URL where the Charge Point SHALL upload the file. The URL also contains the protocol which must be used to upload the file. It is recommended that the log file is uploaded via FTP or FTPS. FTP(S) is better optimized for large binary data than HTTP. Also FTP(S) has the ability to resume uploads. In case an upload is interrupted, the Charge Point can resume uploading after the part it already has uploaded. The FTP URL is of format: *ftp://User:password@host:port /path* in which the parts *User*:*password*@, :*password* or :*port* may be excluded. The Charge Point has an optional Configuration Key that reports which file transfer protocols it supports: `SupportedFileTransferProtocols`. The format of the log file is not prescribed. FTP needs to be able to use Passive FTP, to be able to transverse over as much different typologies as possible.

N01 - Retrieve Log Information - Requirements

Table 33. N01 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
N01.FR.01	Upon receipt of a GetLog.req AND if the requested log information is available	The Charge Point SHALL respond with a GetLog.conf stating the name of the file and status *Accepted*.	
N01.FR.02	N01.FR.01	The Charge Point SHALL start uploading a single log file to the specified location	
N01.FR.03	N01.FR.02 AND The GetLog.req contained logType *SecurityLog*	The Charge Point SHALL upload its security log	
N01.FR.04	N01.FR.02 AND The GetLog.req contained logType *DiagnosticsLog*	The Charge Point SHALL upload its diagnostics.	
N01.FR.05	When a security event happens	The Charge Point SHALL log this event in its security log. See Section 8. Security Events for a list of security events.	
N01.FR.07		Every LogStatusNotification.req that is sent for the upload of a specific log SHALL contain the same requestId as the GetLog.req that started this log upload.	
N01.FR.08	When uploading a log document is started	The Charge Point SHALL send a LogStatusNotification.req with status *Uploading*.	
N01.FR.09	When a log document is uploaded successfully	The Charge Point SHALL send a LogStatusNotification.req with status *Uploaded*.	

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
N01.FR.10	When uploading a log document failed	The Charge Point SHALL send a LogStatusNotification.req with status *UploadFailed*, *BadMessage*, *PermissionDenied* OR *NotSupportedOperation*.	It is RECOMMENDED to send a status that describes the reason of failure as precise as possible.
N01.FR.11	When a Charge Point is uploading a log file AND the Charge Point receives a new GetLog.req	The Charge Point SHOULD cancel the ongoing log file upload AND respond with status *AcceptedCanceled*.	
N01.FR.12		The field *requestId* in LogStatusNotification.req is mandatory, unless the message was triggered by an ExtendedTriggerMessage.req AND there is no log upload ongoing.	

4. Secure firmware update

L01 - Secure Firmware Update

Table 34. L01 - Secure Firmware Update

NO.	TYPE	DESCRIPTION
1	**Name**	Secure Firmware Update
2	**ID**	L01
3	**Objective(s)**	Download and install a Secure firmware update.
4	**Description**	Illustrate how a Charge Point processes a Secure firmware update.
	Actors	Central System, Charge Point
	Scenario description	**1.** The Central System sends a SignedUpdateFirmware.req message that contains the location of the firmware, the time after which it should be retrieved, and information on how many times the Charge Point should retry downloading the firmware. **2.** The Charge Point verifies the validity of the certificate against the Manufacturer root certificate. **3.** If the certificate is not valid or could not be verified, the Charge Point aborts the firmware update process and sends a SignedUpdateFirmware.conf with status InvalidCertificate (or status RevokedCertificate when the certificate has been revoked) and a SecurityEventNotification.req with the security event InvalidFirmwareSigningCertificate. If the certificate is valid, the Charge Point starts downloading the firmware, and sends a SignedFirmwareStatusNotification.req with status Downloading. **4.** If the Firmware successfully downloaded, the Charge Point sends a SignedFirmwareStatusNotification.req with status Downloaded. Otherwise, it sends a SignedFirmwareStatusNotification.req with status DownloadFailed. **5.** If the verification is successful, the Charge Point sends a SignedFirmwareStatusNotification.req with status Installing. If the verification of the firmware fails or if a signature is missing entirely, the Charge Point sends a SignedFirmwareStatusNotification.req with status InvalidSignature and a SecurityEventNotification.req with the security event InvalidFirmwareSignature. **6.** If the installation is successful, the Charge Point sends a SignedFirmwareStatusNotification.req with status Installed. Otherwise, it sends a SignedFirmwareStatusNotification.req with status InstallationFailed.
5	**Prerequisite(s)**	The Charge Point Manufacturer provided a firmware update, signing certificate and signature.
6	**Postcondition(s)**	**Successful postcondition:** The firmware is updated and the Charge Point is in *Installed* status. **Failure postconditions:** The certificate is not valid or could not be verified and the Charge Point is in *InvalidCertificate* status. Downloading the firmware failed and the Charge Point is in *DownloadFailed* status. The verification of the firmware's digital signature failed and the Charge Point is in *InvalidSignature* status. The installation of the firmware is not successful and the Charge Point is in *InstallationFailed* status.

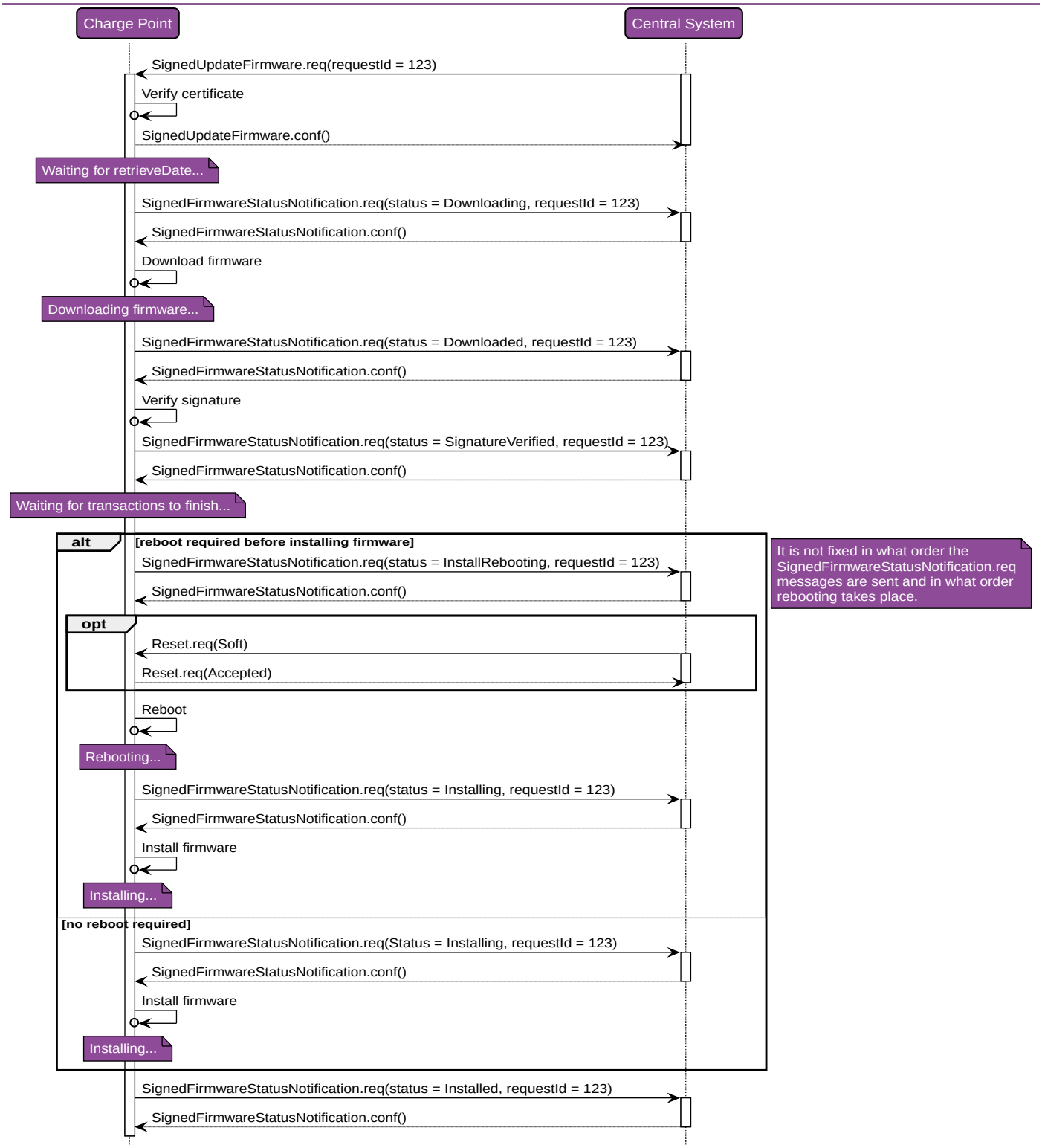

Figure 13. Sequence diagram secure firmware upgrade (happy flow)

7	**Error handling**	n/a

8	**Remark(s)**	Measures SHOULD be taken to secure the firmware when it is stored on a server or workstation. The Charge Point has a required Configuration Key that reports which file transfer protocols it supports: `SupportedFileTransferProtocols` The requirements for the Firmware Signing Certificate are described in the: Certificate Properties section. The manufacturer SHALL NOT use intermediate certificates for the firmware signing certificate in the Charge Point. FTP needs to be able to use Passive FTP, to be able to transverse over as much different typologies as possible.

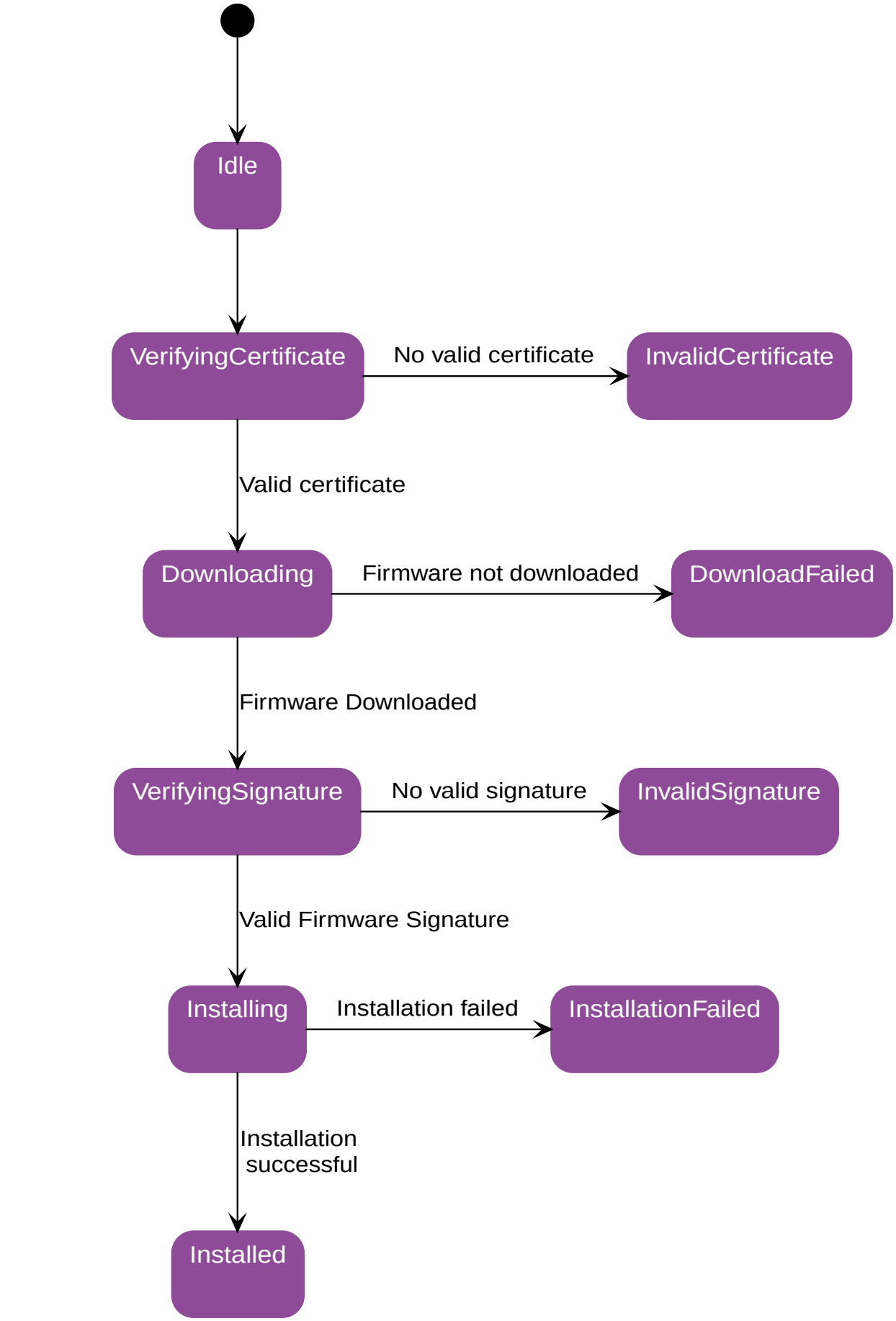

Figure 14. Firmware update process

L01 - Secure Firmware Update - Requirements

Table 35. L01 - Requirements

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.01	Whenever the Charge Point enters a new state in the firmware update process.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req message to the Central System with this new status. What reason to use is described in the description of FirmwareStatusEnumType.	
L01.FR.02	When the Charge Point enters the Invalid Certificate state in the firmware process.	The Charge Point SHALL send a SecurityEventNotification.req message to the Central System with the security event InvalidFirmwareSigningCertificate.	
L01.FR.03	When the Charge Point enters the Invalid Signature state.	The Charge Point SHALL send a SecurityEventNotification.req message to the Central System with the security event InvalidFirmwareSignature.	
L01.FR.04	When the Charge Point has successfully downloaded the new firmware	The signature SHALL be validated, by calculating the signature over the entire firmware file using the RSA-PSS or ECSchnorr algorithm for signing, and the SHA256 algorithm for calculating hash values.	
L01.FR.05	L01.FR.04 AND installDateTime is not set	The Charge Point SHALL install the new firmware as soon as it is able to.	
L01.FR.06	L01.FR.05 AND The Charge Point has ongoing transactions AND When it is not possible to continue charging during installation of firmware	The Charge Point SHALL wait until all transactions have ended, before commencing installation.	
L01.FR.07	L01.FR.06	The Charge Point SHALL set all connectors that are not in use to UNAVAILABLE while the Charge Point waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.	
L01.FR.08		It is RECOMMENDED that the firmware is sent encrypted to the Charge Point. This can either be done by using a secure protocol (such as HTTPS, SFTP, or FTPS) to send the firmware, or by encrypting the firmware itself before sending it.	

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.09		Firmware updates SHALL be digitally protected to ensure authenticity and to provide proof of origin.	This protection is achieved by applying a digital signature over the hash value of the firmware image. Ideally, this signature is already computed by the manufacturer. This way proof of origin of the firmware image can be tracked back to the original author of the firmware.
L01.FR.10		Every SignedFirmwareStatusNotification.req that is sent for a specific firmware update SHALL contain the same requestId as the SignedUpdateFirmware.req that started this firmware update.	
L01.FR.11		For security purposes the Central System SHALL include the Firmware Signing certificate (see Keys used in OCPP) in the SignedUpdateFirmware.req.	
L01.FR.12		For verifying the certificate (see Certificate Hierarchy) use the rules for X.509 certificates [9]. The Charge Point MUST verify the file's digital signature using the Firmware Signing certificate.	
L01.FR.13	When the Charge Point enters the Download Scheduled state.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status DownloadScheduled.	For example when it is busy with installing another firmware or it is busy Charging.
L01.FR.14	When the Charge Point enters the Download Paused state.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status DownloadPaused.	For example when the Charge Point has tasks with higher priorities.
L01.FR.15	When a Charge Point needs to reboot before installing the downloaded firmware.	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status InstallRebooting, before rebooting.	
L01.FR.16	L01.FR.04 AND When installDateTime is set to a future date-time	The Charge Point SHALL send a SignedFirmwareStatusNotification.req with status InstallScheduled and install the firmware at the specified installation time.	

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.17	L01.FR.16 AND current DateTime >= InstallDateTime	The Charge Point SHALL install the new firmware as soon as it is able to.	
L01.FR.18	L01.FR.17 AND The Charge Point has ongoing transactions AND It is not possible to continue charging during installation of firmware	The Charge Point SHALL wait until all transactions have ended, before commencing installation.	
L01.FR.19	L01.FR.18	The Charge Point SHALL set all connectors that are not in use to UNAVAILABLE while the Charge Point waits for the ongoing transactions to end. Until the firmware is installed, any connector that becomes available SHALL be set to UNAVAILABLE.	
L01.FR.20	When the Charge Point receives a UpdateFirmware.req (the original OCPP 1.6 message)	The Charge Point SHALL respond with a WebSocket RPC CALLERROR NotSupported, and the Charge Point SHALL NOT start the Firmware Update process.	
L01.FR.21		The field requestId in SignedFirmwareStatusNotification.req is mandatory, unless *status* = `Idle`.	
L01.FR.22	When the Charge Point needs to reboot during a firmware update AND the bootloader is unable to send OCPP messages	The Charge Point MAY omit the SignedFirmwareStatusNotification.req (status=Installing) message.	
L01.FR.23	When the Charge Point receives an SignedUpdateFirmware.req	The Charge Point SHALL validate the certificate before accepting the message.	
L01.FR.24	L01.FR.23 AND the certificate is invalid	The Charge Point SHALL respond with SignedUpdateFirmware.conf (status=InvalidCertificate).	
L01.FR.25	L01.FR.23 AND the certificate is revoked	The Charge Point SHALL respond with SignedUpdateFirmware.conf (status=RevokedCertificate).	
L01.FR.26	When a Charge Point is installing new Firmware OR is going to install new Firmware, but has received an SignedUpdateFirmware.req command to install it at a later time AND the Charge Point receives a new SignedUpdateFirmware.req	The Charge Point SHOULD cancel the ongoing firmware update AND respond with status *AcceptedCanceled*.	The Charge Point SHOULD NOT first check if the new firmware file exists, this way the Central System will be able to cancel an ongoing firmware update without starting a new one.

ID	PRECONDITION	REQUIREMENT DEFINITION	NOTE
L01.FR.27	L01.FR.26 AND the Charge Point is unabled to cancel the installation	The Charge Point MAY respond with status *Rejected*.	
L01.FR.28	Charge Point receives a ExtendedTriggerMessage.req for `FirmwareStatusNotification` AND last sent SignedFirmwareStatusNotification.req had *status* = `Installed`	Charge Point SHALL return a SignedFirmwareStatusNotification.req with *status* = `Idle`.	
L01.FR.29	Charge Point receives a ExtendedTriggerMessage.req for `FirmwareStatusNotification` AND last sent SignedFirmwareStatusNotification.req had *status* <> `Installed`	Charge Point SHALL return a SignedFirmwareStatusNotification.req with the last sent *status*.	

5. Messages

To add the functionality needed for this WhitePaper, a couple of messages have been added from OCPP 2.0.1. Most have their original name from OCPP 2.0.1. Others have a modified name, because they have been modified between 1.6 and 2.0.1. The messages that have been renamed, are marked as such.

5.1. CertificateSigned.req

This contains the field definition of the CertificateSigned.req PDU sent by the Central System to the Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateChain	string[0..10000]	1..1	Required. The signed PEM encoded X.509 certificates. This can also contain the necessary sub CA certificates. The maximum size of this field is be limited by the configuration key: `CertificateSignedMaxSize`.

5.2. CertificateSigned.conf

This contains the field definition of the CertificateSigned.conf PDU sent by the Charge Point to the Central System in response to a CertificateSigned.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	CertificateSignedStatusEnumType	1..1	Required. Returns whether certificate signing has been accepted, otherwise rejected.

5.3. DeleteCertificate.req

Used by the Central System to request deletion of an installed certificate on a Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateHashData	CertificateHashDataType	1..1	Required. Indicates the certificate of which deletion is requested.

5.4. DeleteCertificate.conf

Response to a DeleteCertificate.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	DeleteCertificateStatusEnumType	1..1	Required. Charge Point indicates if it can process the request.

5.5. ExtendedTriggerMessage.req

This contains the field definition of the ExtendedTriggerMessage.req PDU sent by the Central System to the Charge Point.

This message is based on the OCPP 2.0.1 TriggerMessageRequest, it has been renamed to: ExtendedTriggerMessage.req, because the original name conflicts with the TriggerMessage.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
requestedMessage	MessageTriggerEnumType	1..1	Required. Type of the message to be triggered.
connectorId	integer connectorId > 0	0..1	Optional. Only filled in when request applies to a specific connector.

5.6. ExtendedTriggerMessage.conf

This contains the field definition of the ExtendedTriggerMessage.conf PDU sent by the Charge Point to the Central System in response to ExtendedTriggerMessage.req.

This message is based on the OCPP 2.0.1 TriggerMessageResponse, it has been renamed to: ExtendedTriggerMessage.conf, because the original name conflicts with the TriggerMessage.conf from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	TriggerMessageStatusEnumType	1..1	Required. Indicates whether the Charge Point will send the requested notification or not.

5.7. GetInstalledCertificateIds.req

Used by the Central System to request an overview of the installed certificates on a Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateType	CertificateUseEnumType	1..1	Required. Indicates the type of certificates requested.

5.8. GetInstalledCertificateIds.conf

Response to a GetInstalledCertificateIDs.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	GetInstalledCertificateStatusEnumType	1..1	Required. Charge Point indicates if it can process the request.
certificateHashData	CertificateHashDataType	0..*	Optional. The Charge Point includes the Certificate information for each available certificate.

5.9. GetLog.req

This contains the field definition of the GetLog.req PDU sent by the Central System to the Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
logType	LogEnumType	1..1	Required. This contains the type of log file that the Charge Point should send.
requestId	integer	1..1	Required. The Id of this request
retries	integer	0..1	Optional. This specifies how many times the Charge Point must try to upload the log before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
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 Charge Point to decide how long to wait between attempts.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
log	LogParametersType	1..1	Required. This field specifies the requested log and the location to which the log should be sent.

5.10. GetLog.conf

This contains the field definition of the GetLog.conf PDU sent by the Charge Point to the Central System in response to a GetLog.req.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	LogStatusEnumType	1..1	Required. This field indicates whether the Charge Point was able to accept the request.
filename	string[0..255]	0..1	Optional. This contains the name of the log file that will be uploaded. This field is not present when no logging information is available.

5.11. InstallCertificate.req

Used by the Central System to request installation of a certificate on a Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
certificateType	CertificateUseEnumType	1..1	Required. Indicates the certificate type that is sent.
certificate	string[0..5500]	1..1	Required. An PEM encoded X.509 certificate.

5.12. InstallCertificate.conf

The response to a InstallCertificate.req, sent by the Charge Point to the Central System.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	CertificateStatusEnumType	1..1	Required. Charge Point indicates if installation was successful.

5.13. LogStatusNotification.req

This contains the field definition of the LogStatusNotification.req PDU sent by the Charge Point to the Central System.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	UploadLogStatusEnumType	1..1	Required. This contains the status of the log upload.
requestId	integer	0..1	Optional. The request id that was provided in the GetLog.req that started this log upload.

5.14. LogStatusNotification.conf

This contains the field definition of the LogStatusNotification.conf PDU sent by the Central System to the Charge Point in response to LogStatusNotification.req.

No fields are defined.

5.15. SecurityEventNotification.req

Sent by the Charge Point to the Central System in case of a security event.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
type	string[50]	1..1	Required. Type of the security event (See list of currently known security events)
timestamp	dateTime	1..1	Required. Date and time at which the event occurred.
techInfo	string[0..255]	0..1	Additional information about the occurred security event.

5.16. SecurityEventNotification.conf

Sent by the Central System to the Charge Point to confirm the receipt of a SecurityEventNotification.req message.

No fields are defined.

5.17. SignCertificate.req

Sent by the Charge Point to the Central System to request that the Certificate Authority signs the public key into a certificate.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
csr	string[0..5500]	1..1	Required. The Charge Point SHALL send the public key in form of a Certificate Signing Request (CSR) as described in RFC 2986 [14] and then PEM encoded, using the SignCertificate.req message.

5.18. SignCertificate.conf

Sent by the Central System to the Charge Point in response to the SignCertificate.req message.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	GenericStatusEnumType	1..1	Required. Specifies whether the Central System can process the request.

5.19. SignedFirmwareStatusNotification.req

This contains the field definition of the SignedFirmwareStatusNotification.req PDU sent by the Charge Point to the Central System.

This is the OCPP 2.0.1 FirmwareStatusNotificationRequest, it has been renamed to SignedFirmwareStatusNotification.req, because the original name conflicts with the FirmwareStatusNotification.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	FirmwareStatusEnumType	1..1	Required. This contains the progress status of the firmware installation.
requestId	integer	0..1	Optional. The request id that was provided in the SignedUpdateFirmware.req that started this firmware update. This field is mandatory, unless the message was triggered by a TriggerMessage.req or the ExtendedTriggerMessage.req AND there is no firmware update ongoing.

5.20. SignedFirmwareStatusNotification.conf

This contains the field definition of the SignedFirmwareStatusNotification.conf PDU sent by the Central System to the Charge Point in response to a SignedFirmwareStatusNotification.req.

This is the OCPP 2.0.1 FirmwareStatusNotificationResponse, it is renamed to: SignedFirmwareStatusNotification.conf, because the original name conflicts with the FirmwareStatusNotification.conf from OCPP 1.6.

No fields are defined.

5.21. SignedUpdateFirmware.req

This contains the field definition of the SignedUpdateFirmware.req PDU sent by the Central System to the Charge Point.

This is the OCPP 2.0.1 UpdateFirmwareRequest, it is renamed to SignedUpdateFirmware.req, it is renamed because the original name conflicts with the UpdateFirmware.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
retries	integer	0..1	Optional. This specifies how many times Charge Point must try to download the firmware before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
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 Charge Point to decide how long to wait between attempts.
requestId	integer	1..1	Required. The Id of this request
firmware	FirmwareType	1..1	Required. Specifies the firmware to be updated on the Charge Point.

5.22. SignedUpdateFirmware.conf

This contains the field definition of the SignedUpdateFirmware.conf PDU sent by the Charge Point to the Central System in response to an SignedUpdateFirmware.req.

This is the OCPP 2.0.1 UpdateFirmwareResponse, it is renamed to SignedUpdateFirmware.conf, it is renamed because the original name conflicts with the UpdateFirmware.req from OCPP 1.6.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	UpdateFirmwareStatusEnumType	1..1	Required. This field indicates whether the Charge Point was able to accept the request.

6. Datatypes

6.1. CertificateHashDataType

Class

CertificateHashDataType is used by: DeleteCertificate.req, GetInstalledCertificateIds.conf

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
hashAlgorithm	HashAlgorithmEnumType	1..1	Required. Used algorithms for the hashes provided.
issuerNameHash	CiString128Type	1..1	Required. The hash of the issuer's distinguished name (DN), that must be calculated over the DER encoding of the issuer's name field in the certificate being checked. The hash is represented in hexbinary format (i.e. each byte is represented by 2 hexadecimal digits). Please refer to the OCSP specification: RFC6960 [15].

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
issuerKeyHash	CiString128Type	1..1	Required. The hash of the DER encoded public key: the value (excluding tag and length) of the subject public key field in the issuer's certificate. The hash is represented in hexbinary format (i.e. each byte is represented by 2 hexadecimal digits). Please refer to the OCSP specification: RFC6960 [15].
serialNumber	CiString40Type	1..1	Required. The serial number as a hexadecimal string without leading zeroes (and *without* the prefix *0x*). For example: the serial number with decimal value 4095 will be represented as "FFF". Please note: The serial number of a certificate is a non-negative integer of at most 20 bytes. Since this is too large to be handled as a number in many system, it is represented as a string that contains the hexadecimal representation of this number. The string shall not have any leading zeroes.

6.2. CertificateSignedStatusEnumType

Enumeration

CertificateSignedStatusEnumType is used by: CertificateSigned.conf

VALUE	DESCRIPTION
Accepted	Signed certificate is valid.
Rejected	Signed certificate is invalid.

6.3. CertificateStatusEnumType

Enumeration

Status of the certificate.

CertificateStatusEnumType is used by: InstallCertificate.conf

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 CPO tries to install more certificates than allowed.

6.4. CertificateUseEnumType

Enumeration

CertificateUseEnumType is used by: GetInstalledCertificateIds.req, InstallCertificate.req

VALUE	DESCRIPTION
CentralSystemRootCertificate	Root certificate, used by the CA to sign the Central System and Charge Point certificate.
ManufacturerRootCertificate	Root certificate for verification of the Manufacturer certificate.

6.5. CiString40Type

Type

Generic case insensitive string of 40 characters.

FIELD TYPE	DESCRIPTION
CiString[40]	String is case insensitive.

6.6. CiString128Type

Type

Generic case insensitive string of 128 characters.

FIELD TYPE	DESCRIPTION
CiString[128]	String is case insensitive.

6.7. DeleteCertificateStatusEnumType

Enumeration

DeleteCertificateStatusEnumType is used by: DeleteCertificate.conf

VALUE	DESCRIPTION
Accepted	Normal successful completion (no errors).
Failed	Processing failure.

VALUE	DESCRIPTION
NotFound	Requested resource not found.

6.8. FirmwareStatusEnumType

Enumeration

Status of a firmware download.

A value with "Intermediate state" in the description, is an intermediate state, update process is not finished.

A value with "Failure end state" in the description, is an end state, update process has stopped, update failed.

A value with "Successful end state" in the description, is an end state, update process has stopped, update successful.

FirmwareStatusEnumType is used by: SignedFirmwareStatusNotification.req

VALUE	DESCRIPTION
Downloaded	Intermediate state. New firmware has been downloaded by Charge Point.
DownloadFailed	Failure end state. Charge Point failed to download firmware.
Downloading	Intermediate state. Firmware is being downloaded.
DownloadScheduled	Intermediate state. Downloading of new firmware has been scheduled.
DownloadPaused	Intermediate state. Downloading has been paused.
Idle	Charge Point is not performing firmware update related tasks. Status Idle SHALL only be used as in a SignedFirmwareStatusNotification.req that was triggered by ExtendedTriggerMessage.req.
InstallationFailed	Failure end state. Installation of new firmware has failed.
Installing	Intermediate state. Firmware is being installed.
Installed	Successful end state. New firmware has successfully been installed in Charge Point.
InstallRebooting	Intermediate state. Charge Point is about to reboot to activate new firmware. This status MAY be omitted if a reboot is an integral part of the installation and cannot be reported separately.

VALUE	DESCRIPTION
InstallScheduled	Intermediate state. Installation of the downloaded firmware is scheduled to take place on installDateTime given in SignedUpdateFirmware.req.
InstallVerificationFailed	Failure end state. Verification of the new firmware (e.g. using a checksum or some other means) has failed and installation will not proceed. (Final failure state)
InvalidSignature	Failure end state. The firmware signature is not valid.
SignatureVerified	Intermediate state. Provide signature successfully verified.

6.9. FirmwareType

Class

Represents a copy of the firmware that can be loaded/updated on the Charge Point.

FirmwareType is used by: SignedUpdateFirmware.req

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
location	string[0..512]	1..1	Required. URI defining the origin of the firmware.
retrieveDateTime	dateTime	1..1	Required. Date and time at which the firmware shall be retrieved.
installDateTime	dateTime	0..1	Optional. Date and time at which the firmware shall be installed.
signingCertificate	string[0..5500]	1..1	Required. Certificate with which the firmware was signed. PEM encoded X.509 certificate.
signature	string[0..800]	1..1	Required. Base64 encoded firmware signature.

6.10. GenericStatusEnumType

Enumeration

Generic message response status

VALUE	DESCRIPTION
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.

6.11. GetInstalledCertificateStatusEnumType

Enumeration

GetInstalledCertificateStatusEnumType is used by: GetInstalledCertificateIds.conf

VALUE	DESCRIPTION
Accepted	Normal successful completion (no errors).
NotFound	Requested certificate not found.

6.12. HashAlgorithmEnumType

Enumeration

HashAlgorithmEnumType is used by: CertificateHashDataType

VALUE	DESCRIPTION
SHA256	SHA-256 hash algorithm.
SHA384	SHA-384 hash algorithm.
SHA512	SHA-512 hash algorithm.

6.13. LogEnumType

Enumeration

LogEnumType is used by: GetLog.req

VALUE	DESCRIPTION
DiagnosticsLog	This contains the field definition of a diagnostics log file
SecurityLog	Sent by the Central System to the Charge Point to request that the Charge Point uploads the security log.

6.14. LogParametersType

Class

Class for detailed information the retrieval of logging entries.

LogParametersType is used by: GetLog.req

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
remoteLocation	string[0..512]	1..1	Required. The URL of the location at the remote system where the log should be stored.
oldestTimestamp	dateTime	0..1	Optional. This contains the date and time of the oldest logging information to include in the diagnostics.
latestTimestamp	dateTime	0..1	Optional. This contains the date and time of the latest logging information to include in the diagnostics.

6.15. LogStatusEnumType

Enumeration

LogStatusEnumType is used by: GetLog.conf

VALUE	DESCRIPTION
Accepted	Accepted this log upload. This does not mean the log file is uploaded is successfully, the Charge Point will now start the log file upload.
Rejected	Log update request rejected.
AcceptedCanceled	Accepted this log upload, but in doing this has canceled an ongoing log file upload.

6.16. MessageTriggerEnumType

Enumeration

Type of request to be triggered by trigger messages.

MessageTriggerEnumType is used by: ExtendedTriggerMessage.req

VALUE	DESCRIPTION
BootNotification	To trigger BootNotification.req.
LogStatusNotification	To trigger LogStatusNotification.req.
FirmwareStatusNotification	To trigger SignedFirmwareStatusNotification.req (So the status of the secure firmware update introduced in this document).
Heartbeat	To trigger Heartbeat.req.

VALUE	DESCRIPTION
MeterValues	To trigger MeterValues.req.
SignChargePointCertificate	To trigger a SignCertificate.req with certificateType: ChargePointCertificate.
StatusNotification	To trigger SatusNotification.req.

6.17. TriggerMessageStatusEnumType

Enumeration

TriggerMessageStatusEnumType is used by: ExtendedTriggerMessage.conf

VALUE	DESCRIPTION
Accepted	Requested message will be sent.
Rejected	Requested message will not be sent.
NotImplemented	Requested message cannot be sent because it is either not implemented or unknown.

6.18. UpdateFirmwareStatusEnumType

Enumeration

UpdateFirmwareStatusEnumType is used by: SignedUpdateFirmware.conf

VALUE	DESCRIPTION
Accepted	Accepted this firmware update request. This does not mean the firmware update is successful, the Charge Point will now start the firmware update process.
Rejected	Firmware update request rejected.
AcceptedCanceled	Accepted this firmware update request, but in doing this has canceled an ongoing firmware update.
InvalidCertificate	The certificate is invalid.
RevokedCertificate	Failure end state. The Firmware Signing certificate has been revoked.

6.19. UploadLogStatusEnumType

Enumeration

UploadLogStatusEnumType is used by: LogStatusNotification.req

VALUE	DESCRIPTION
BadMessage	A badly formatted packet or other protocol incompatibility was detected.
Idle	The Charge Point is not uploading a log file. Idle SHALL only be used when the message was triggered by a ExtendedTriggerMessage.req.
NotSupportedOperation	The server does not support the operation
PermissionDenied	Insufficient permissions to perform the operation.
Uploaded	File has been uploaded successfully.
UploadFailure	Failed to upload the requested file.
Uploading	File is being uploaded.

7. Configuration Keys

7.1. `AdditionalRootCertificateCheck`

Required/optional	optional
Accessibility	R
Type	boolean
Description	When set to true, only one certificate (plus a temporarily fallback certificate) of certificateType CentralSystemRootCertificate is allowed to be installed at a time. When installing a new Central System Root certificate, the new certificate SHALL replace the old one AND the new Central System Root Certificate MUST be signed by the old Central System Root Certificate it is replacing. This configuration key 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 Charge Point needs to perform a full certificate chain verification when the new Central System Root certificate is being installed. However, once the old Central System Root certificate is set as the fallback certificate, the Charge Point 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 Central System Root (fallback) certificate is either expired or removed.*

7.2. AuthorizationKey

Required/optional	optional
Accessibility	W
Type	String
Description	The basic authentication password is used for HTTP Basic Authentication, minimal length: 16 bytes. It is strongly advised to be randomly generated binary to get maximal entropy. Hexadecimal represented (20 bytes maximum, represented as a string of up to 40 hexadecimal digits). This configuration key is write-only, so that it cannot be accidentally stored in plaintext by the Central System when it reads out all configuration keys. This configuration key is required unless only "security profile 3 - TLS with client side certificates" is implemented.

7.3. CertificateSignedMaxChainSize

Required/optional	optional
Accessibility	R
Type	integer
Description	This configuration key can be used to limit the size of the 'certificateChain' field from the CertificateSigned.req PDU. The value of this configuration key has a maximum limit of 10.000 characters.

7.4. CertificateStoreMaxLength

Required/optional	optional
Accessibility	R
Type	integer
Description	Maximum number of Root/CA certificates that can be installed in the Charge Point.

7.5. CpoName

Required/optional	optional
Accessibility	RW
Type	String

Description	This configuration key contains CPO name (or an organization trusted by the CPO) as used in the Charge Point Certificate. This is the CPO name that is to be used in a CSR send via: SignCertificate.req

7.6. SecurityProfile

Required/optional	optional
Accessibility	RW
Type	integer
Description	This configuration key is used to set the security profile used by the Charge Point. The value of this configuration key can only be increased to a higher level, not decreased to a lower level, if the Charge Point receives a lower value then currently configured,the Charge Point SHALL Rejected the ChangeConfiguration.req Before accepting the new value, the Charge Point SHALL check if all the prerequisites for the new Security Profile are met, if not, the Charge Point SHALL Rejected the ChangeConfiguration.req. After the security profile was successfully changed, the Charge Point disconnects from the Central System and SHALL reconnect using the new configured Security Profile. Default, when no security profile is yet configured: 0.

8. Security Events

The table below provides a list of security events. Security events that are critical should be pushed to the Central System.

SECURITY EVENT	DESCRIPTION	CRITICAL
FirmwareUpdated	The Charge Point firmware is updated	Yes
FailedToAuthenticateAtCentralSystem	The authentication credentials provided by the Charge Point were rejected by the Central System	No
CentralSystemFailedToAuthenticate	The authentication credentials provided by the Central System were rejected by the Charge Point	No
SettingSystemTime	The system time on the Charge Point was changed	Yes
StartupOfTheDevice	The Charge Point has booted	Yes
ResetOrReboot	The Charge Point was rebooted or reset	Yes
SecurityLogWasCleared	The security log was cleared	Yes

SECURITY EVENT	DESCRIPTION	CRITICAL
ReconfigurationOfSecurityParameters	Security parameters, such as keys or the security profile used, were changed	No
MemoryExhaustion	The Flash or RAM memory of the Charge Point is getting full	Yes
InvalidMessages	The Charge Point has received messages that are not valid OCPP messages, if signed messages, signage invalid/incorrect	No
AttemptedReplayAttacks	The Charge Point has received a replayed message (other than the Central System trying to resend a message because it there was for example a network problem)	No
TamperDetectionActivated	The physical tamper detection sensor was triggered	Yes
InvalidFirmwareSignature	The firmware signature is not valid	No
InvalidFirmwareSigningCertificate	The certificate used to verify the firmware signature is not valid	No
InvalidCentralSystemCertificate	The certificate that the Central System uses was not valid or could not be verified	No
InvalidChargePointCertificate	The certificate sent to the Charge Point using the SignCertificate.conf message is not a valid certificate	No
InvalidTLSVersion	The TLS version used by the Central System is lower than 1.2 and is not allowed by the security specification	No
InvalidTLSCipherSuite	The Central System did only allow connections using TLS cipher suites that are not allowed by the security specification	No

9. Changelog Edition 2

SECTION / USE CASE	CHANGE
2.3. Unsecured Transport with Basic Authentication Profile	Basic auth example added to remarks.
2.4.1. TLS with Basic Authentication Profile	A00.FR.308 changed. "URL or IP address" changed to "FQDN".
2.4.1. TLS with Basic Authentication Profile	A00.FR.317 changed. Added a note.
2.5.1. TLS with Client Side Certificates Profile	A00.FR.405 changed. "unique identifier" changed to "unique serial number".

SECTION / USE CASE	CHANGE
2.5.1. TLS with Client Side Certificates Profile	A00.FR.412 changed. "URL" changed to "FQDN".
2.5.1. TLS with Client Side Certificates Profile	A00.FR.429 added.
2.6.1. Certificate Properties	A00.FR.507 changed. Encoding changed from DER, followed by Base64 encoding to PEM.
2.6.1. Certificate Properties	A00.FR.510 changed. "full URL of the endpoint" changed to "FQDN".
2.6.2. Certificate Hierarchy	A00.FR.604, A00.FR.605 removed.
A02/A03	Prerequisite added. "The configuration variable `CpoName` MUST be set."
A02/A03	Error handling added.
A02/A03	A02.FR.03/A03.FR.03 changed. PEM encoding included.
A02/A03	A02.FR.04/A03.FR.04 changed. The dedicated authority server MAY be operated by the CPO.
A05	Error handling and requirements; A05.FR.08, A05.FR.09 added.
L01	Added requirements; L01.FR.21, L01.FR.22, L01.FR.23, L01.FR.24, L01.FR.25, L01.FR.26, L01.FR.27, L01.FR.28, L01.FR.29.
M04	M04.FR.07 added.
M05	M05.FR.05, M05.FR.06, M05.FR.07, M05.FR.08, M05.FR.09 added in v1.1. M05.FR.05, M05.FR.07 removed in v1.2 M05.FR.08, M05.FR.09 changed in v1.2 M05.FR.10, M05.FR.11, M05.FR.12, M05.FR.13 added in v1.2
N01	N01.FR.11, N01.FR.12 added.
5.1. CertificateSigned.req	Changes in 'cert' field. Field name changed from 'cert' to 'certificateChain'. Field type changed from string[0..5500] to string[0..10000]. Cardinality changed from 1..* to 1..1. Encoding changed from DER, then Hex encoded into a case insensitive string to PEM.
5.7. GetInstalledCertificateIds.req	'typeOfCertificate' field renamed to 'certificateType'.
5.11. InstallCertificate.req	'certificate' field encoding changed from DER, then Hex encoded into a case insensitive string to PEM.
5.17. SignCertificate.req	'csr' field encoding changed from DER to PEM.

SECTION / USE CASE	CHANGE
5.13. LogStatusNotification.req	'requestId' field cardinality changed from 1..1 to 0..1
5.15. SecurityEventNotification.req	'techInfo' field added.
5.19. SignedFirmwareStatusNotification.req	'requestId' field cardinality changed from 1..1 to 0..1
6.1. CertificateHashDataType	'issuerKeyHash' field type changed from string[0..128] to identifierString[0..128].
6.1. CertificateHashDataType	'serialNumber' field type changed from string[0..20] to string[0..40].
6.6. FirmwareStatusEnumType	Enum values 'InvalidCertificate', 'RevokedCertificate', 'CertificateVerified' removed.
6.7. FirmwareType	'signingCertificate' field encoding changed from DER, then Hex encoded into a case insensitive string to PEM.
6.16. UpdateFirmwareStatusEnumType	Enum values 'InvalidCertificate', 'RevokedCertificate' added.
7. Configuration Keys	Configuration key 'CertificateSignedMaxChain' removed.
7. Configuration Keys	Configuration key 'CertificateSignedMaxChainSize' added.
7. Configuration Keys	Configuration key 'AdditionalRootCertificateCheck' added.
8. Security Events	'FailedToAuthenticateAtCentral System' changed to: 'FailedToAuthenticateAtCentralSystem' removed incorrect whitespace.
8. Security Events	'Central SystemFailedToAuthenticate' changed to: 'CentralSystemFailedToAuthenticate' removed incorrect whitespace.

10. Changelog Edition 3

SECTION / USE CASE	CHANGE
6.1. CertificateHashDataType	Updated descriptions of the fields of this type, clarifying that the contents of these fields must follow the OCSP specification. Corrected the type of the fields to CiStrings.