Provisioning

Table of Contents

1. Scope 2
2. Terminology and Conventions 3
   2.1. Conventions 3
   2.2. Definitions 3
   2.3. Conventions 4
   2.4. References 5
3. Use cases 6
   3.1. UC.01 - Cold Boot Charge Point 6
   3.2. UC.01a - First Boot Charge Point (pending) 9
   3.3. UC.01b - First Boot Charge Point (rejected) 11
4. Messages 14
   4.1. BootNotification.req 14
   4.2. BootNotification.conf 15
   4.3. Heartbeat.req 15
   4.4. Heartbeat.conf 15
   4.5. StatusNotification.req 16
   4.6. StatusNotification.conf 16
5. DataTypes 17
   5.1. ChargePointStatus 17
   5.2. ChargePointErrorCode 18
   5.3. MeterValue 19
   5.4. RegistrationStatus 19
6. Configuration Keys 20
   6.1. `HeartbeatInterval` 20

Use case based functional description between Charge Point and Central System based on OCPP 1.6.

Document Version	0.1
Document Status	DRAFT
Document Release Date	2016-01-27

Copyright © 2010 – 2016 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
0.1	2016-01-27	Jonel Timbergen *ElaadNL* Robert de Leeuw *IHomer*	First 3 test use cases for Cold Boot Charge Point

1. Scope

This document defines the protocol used between a **Charge Point** and **Central System** in use cases. If the protocol requires a certain action or response from one side or the other, then this will be stated in this document.

The specification does not define the communication technology. Any technology will do, as long as it supports TCP/IP connectivity.

For complex systems, the use case methodology supports a common understanding of functionalities, actors and processes across different technical committees or even different organizations. Developed as software engineering tool, the methodology can be used to support the development of standards as it the analysis of requirements in relation to new or existing standards.

2. Terminology and Conventions

2.1. Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119]], subject to the following additional clarification clause:

The phrase “valid reasons in particular circumstances” relating to the usage of the terms “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, and “NOT RECOMMENDED” is to be taken to mean technically valid reasons, such as the absence of necessary hardware to support a function from a charge point design: for the purposes of this specification it specifically excludes decisions made on commercial, or other non-technical grounds, such as cost of implementation, or likelihood of use.

All sections and appendixes, except “Scope” and “Terminology and Conventions”, are normative, unless they are explicitly indicated to be informative.

2.2. Definitions

This section contains the terminology that is used throughout this document.

Central System	Charge Point Management System: the central system that manages Charge Points and has the information for authorizing users for using its Charge Points.
CiString	Case Insensitive String. Only printable ASCII allowed.
Charge Point	The Charge Point is the physical system where an electric vehicle can be charged. A Charge Point has one or more connectors.
Charging Profile	Generic Charging Profile, used for different types of Profiles. Contains information about the Profile and holds the Charging Schedule. In future versions of OCPP it might hold more than 1 Charging Schedule.
Charging Schedule	Part of a Charging Profile. Defines a block of charging Power or Current limits. Can contain a start time and length.
Charging Session	Part of a transaction during which the EV is allowed to request energy
Composite Charging Schedule	The charging schedule as calculated by the Charge Point. It is the result of the calculation of all active schedules and possible local limits present in the Charge Point. Also IEC 15118 limits might be taken into account.
Connector	The term “Connector”, as used in this specification, refers to an independently operated and managed electrical outlet on a Charge Point. This usually corresponds to a single physical connector, but in some cases a single outlet may have multiple physical socket types and/or tethered cable/connector arrangements to facilitate different vehicle types (e.g. four-wheeled EVs and electric scooters).

Control Pilot signal	signal used by a Charge Point to inform EV of maximum Charging power or current limit, as defined by [IEC61851-1].
Energy Transfer Period	Time during which an EV chooses to take offered energy, or return it. Multiple Energy Transfer Periods are possible during a Transaction.
Local Controller	Optional device in a smart charging infrastructure. Located on the premises with a number of Charge Points connected to it. Sits between the Charge Points and Central System. Understands and speaks OCPP messages. Controls the Power or Current in other Charge Point by using OCPP smart charging messages. Can be a Charge Point itself.
OCPP-J	OCPP via JSON over WebSocket
OCPP-S	OCPP via SOAP
Phase Rotation	Defines the wiring order of the phases between the energy meter (or if absent, the grid connection), and the Charge Point connector.
Transaction	The part of the charging process that starts when all relevant preconditions (e.g. authorization, plug inserted) are met, and ends at the moment when the Charge Point irrevocably leaves this state.
String	Case Sensitive String. Only printable ASCII allowed. All strings in messages and enumerations are case sensitive, unless explicitly stated otherwise.

2.3. Conventions

Column heading	Description
CH	Section of OCPP 1.6 specification document
FR/NF/C	Identifies whether it is a functional requirement, non-functional requirement or a constraint.
Functional requirement	Functional requirements are requirements that specify a function that a system or system component must be able to perform.
Non-functional requirement	Non-functional requirements are requirements which are not specifically concerned with the functionality of a system but place restrictions on the product being developed
Constraint	Constraints are a type of non-functional requirement that is imposed by the client that restricts the implementation of the system or the development process.
ID	Use-case identification number.
Use-case	A use case is a specification of sequences of actions, including variant sequences and error sequences, that a system, subsystem, or class can perform by interacting with outside actors
Actor	The actor(s) involved in the use-cases and associated requirements.
Pre-condition	Lists the conditions that must be true before the Use Case starts

ID	Requirement identification number.
Requirement definition	The condition or capability needed by a user, Charge Point and Central System to satisfy the contract, standard, specification, or other formally imposed document.
M/O/C	This column defines whether requirements are mandatory (M) or optional (O) or conditional © for the instantiation of a specific logical node.
Messages	This colomn defines the messages which are used in the use-cases.
Rationale	The logical basis for the requirement.
Note	Extra annotations required for understaning of the requirement.
Requirement OCPP 1.6 specification	Former requirement specification.
Specification part	Section of OCPP 1.6 specification document.
pp	Page number of OCPP 1.6 specification document.

2.4. References

[IEC61851-1]	“IEC 61851-1 2010: Electric vehicle conductive charging system - Part 1: General requirements” https://webstore.iec.ch/publication/6029
[OCPP1.5]	“OCPP 1.5: Open Charge Proint Protocol 1.5” http://www.openchargealliance.org/downloads/
[OCPP_1.6CT]	“OCPP 1.6 Compliance testing” http://www.openchargealliance.org/downloads/
[OCPP_IMP_J]	“OCPP JSON Specification” http://www.openchargealliance.org/downloads/
[OCPP_IMP_S]	“OCPP SOAP Specification” http://www.openchargealliance.org/downloads/
[RFC2119]	“Key words for use in RFCs to Indicate Requirement Levels”. S. Bradner. March 1997. http://www.ietf.org/rfc/rfc2119.txt

3. Use cases

<insert Provisioning description>

3.1. UC.01 - Cold Boot Charge Point

No.	Type	Description
1	**Use case element name**	Cold Boot Charge Point
2	**ID**	UC.01
3	**Objectives**	Give the Central System a way to control which Charge Points are allowed to connect to the system.
4	**Description**	This use case describes how the Central System can control Charge Point accessing its system. To be able to control Charge Points connecting to a Central System, Charge Points are required to send a BootNotification.req, this request contains some information about the Charge Point.
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System receives the status Rejected from the Charge Point. **2.** The Charge Point will resend the BootNotification.req every XX seconds (Interval from the BootNotification.conf) **3.** When the Central System returns with BootNotification.conf with the status accepted. **4.** UC.01 - Psychical Cold Boot Charge Point is valid.
	Alternative scenario's	UC.01a - First Boot Charge Point (pending) UC.01b - First Boot Charge Point (rejected)
5	**Prerequisites**	The Charge Point is powered down.
6	**Post conditions**	The Charge Point is in *idle* state, and *Accepted*.
7	**Sequence diagram**	

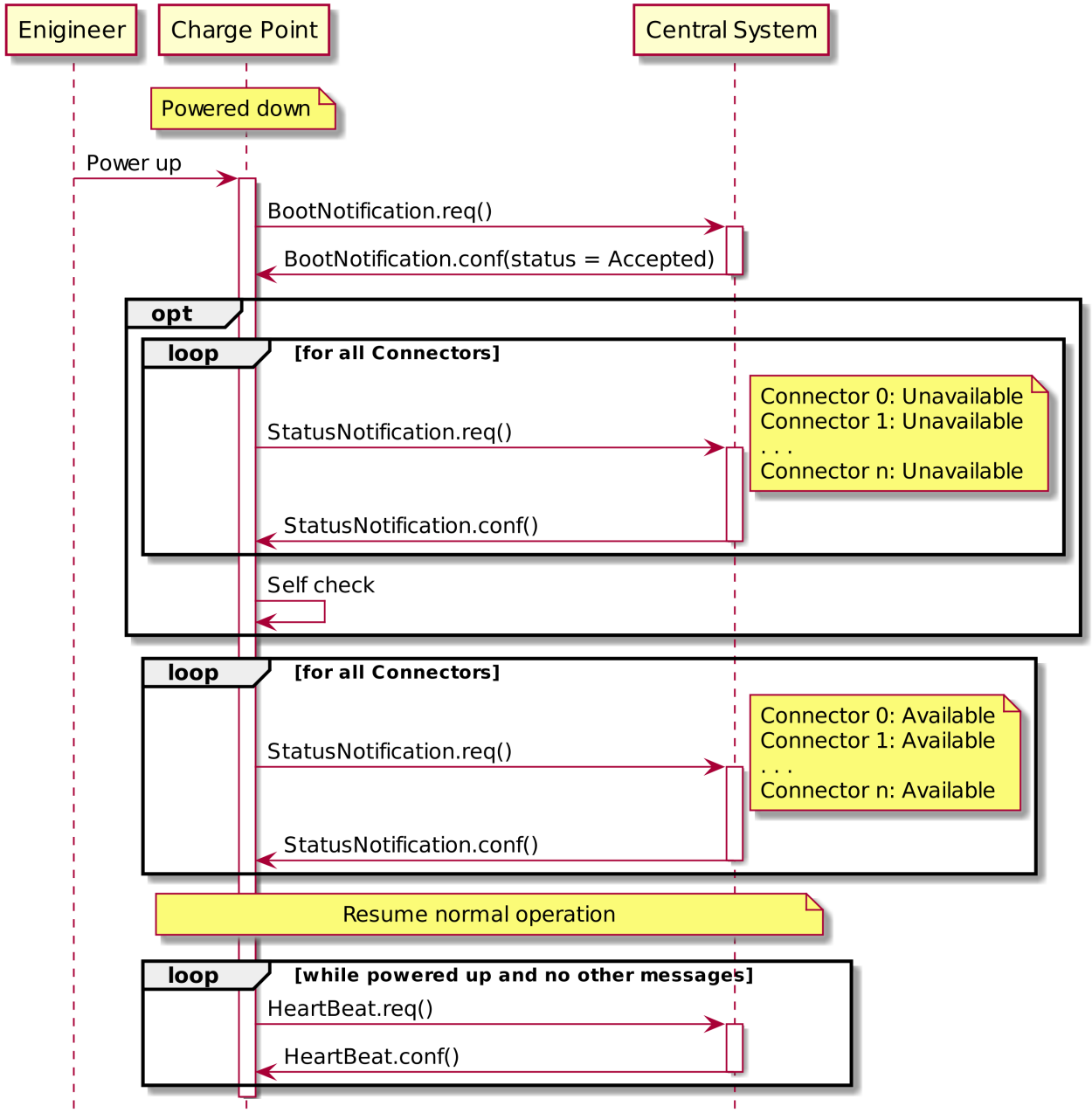

Figure 1. Sequence Diagram: ColdBoot

8	**Error handling**	No initial establishment of connection of high level communication between the Central System and Charge Point.
9	**Remarks**	Most Charge Points boot up and send StatusNotifications with “Unavailable” then do a check of all the hardware and send new StatusNotifications with status “Available” when the Charge Point is up and running. When something is wrong with the Charge Point or connector, the status should be set to “Faulted”. Persistent states: for example: Connector set to Unavailable shall persist a reboot.

3.1.1. UC.01 - Requirements

ID.	Precondition	Requirement	M/O/C
FR.01.001	After start-up.	The Charge Point shall send a request to the Central System with information about its configuration	**M**
FR.01.002	The Central System has received a bootnotification.req from the Charge Point.	The Central System shall respond to indicate whether it will accept the Charge Point.	**M**
FR.01.003	Each time the Charge Point boots or reboots.	The Charge Point shall send a BootNotification.req PDU each time it boots or reboots.	**M**
FR.01.004	Between the physical power-on/reboot and the successful completion of a BootNotification, where the Central System returns Accepted or Pending.	The Charge Point shall NOT send any other request to the Central System. This includes cached messages that are still present in the Charge Point from before.	**C**
FR.01.005	When the Central System responds with a BootNotification.conf with a status Accepted.	The Charge Point shall adjust the heartbeat interval in accordance with the interval from the response PDU.	**M**
FR.01.006	The Charge Point has received the BootNotification.conf.	It is RECOMMENDED to synchronize the Charge Point' its internal clock with the supplied Central System’s current time.	**O**
FR.01.007	When a Charge Point or a Connector iis set to status Unavailable by a Change Availability command.	The 'Unavailable' status MUST be persistent across reboots.	**M**

3.2. UC.01a - First Boot Charge Point (pending)

No.	Type	Description
1	**Use case element name**	First Boot Charge Point (pending)
2	**ID**	UC.01a
3	**Objectives**	**1.** To retrieve or set certain configuration information. **2.** The Central System might be not ready not the accept the Charge Point.
	Actors	Charge Point, Central System
	Scenario description	**1.** The Central System receives the status pending from the Charge Point. **2.** The Central System then, is able to send messages to the Charge Point in order to change the configuration of the Charge Point. **3.** The Charge Point will resend the BootNotification.req every XX seconds (Interval from the BootNotification.conf) **4.** When the Central System returns with BootNotification.conf with the status accepted, UC.01 - Psychical Cold Boot Charge Point is valid.
	Alternative scenario's	UC.01a - First Boot Charge Point (pending) UC.01b - First Boot Charge Point (rejected)
5	**Prerequisites**	The Charge Point received the status Pending.
6	**Post conditions**	The Charge Point is in *Idle* state, and *Accepted*.
7	**Sequence diagram**	

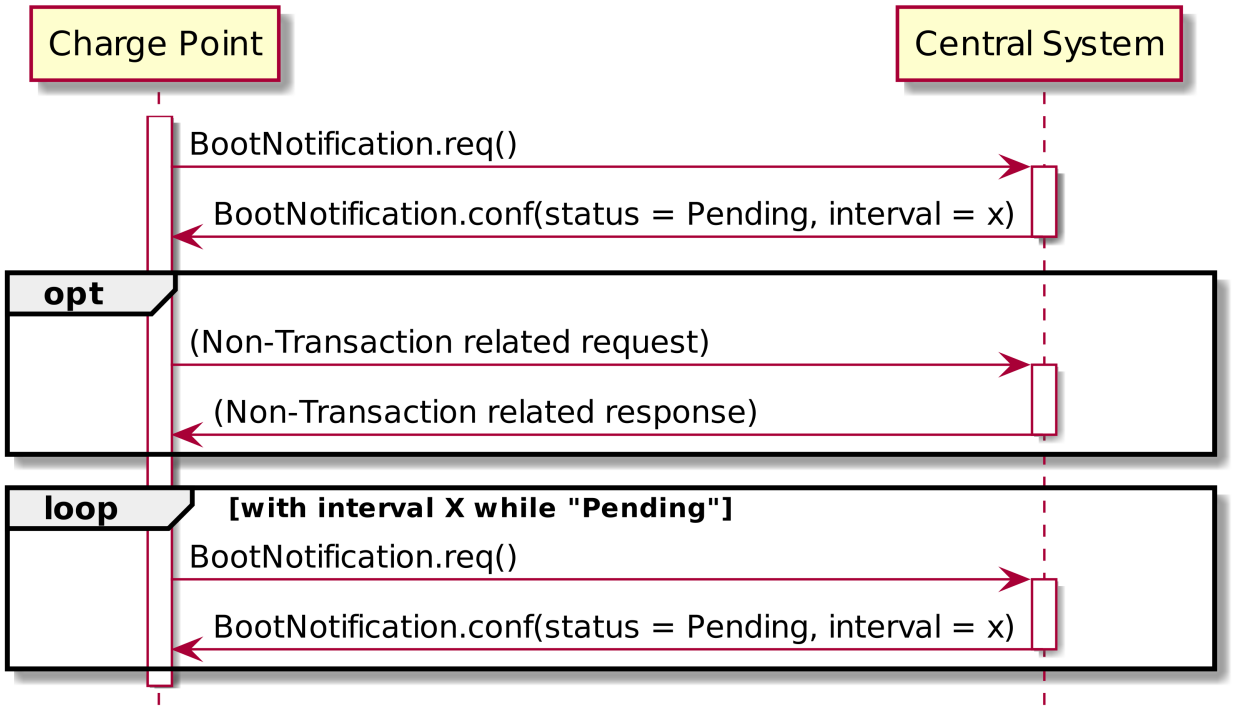

Figure 2. Sequence Diagram: ColdBootA

8	**Error handling**	What to do when a non allowed command is received by the Charge Point? CHECK
9	**Remarks**	CHECK

3.2.1. UC.01a - Requirements

ID.	Precondition	Requirement	M/O/C
FR.01a.001		The Charge Point SHALL NOT interrupt communication and respond normally to all messages from Central System. However, the ChargePoint SHALL NOT initiate any messages.	**C**
FR.01a.002	While in pending state.	The following Central System initiated messages are not allowed: RemoteStartTransaction.req and RemoteStopTransaction.req	**c**
FR.01a.003	If the Central System returns the Pending status.	The communication channel SHOULD NOT be closed by either the Charge Point or the Central System.	**C**
FR.01a.004		The Central System MAY send request messages to retrieve information from the Charge Point or change its configuration. The Charge Point SHOULD respond to these messages.	**O**
FR.01a.005		The Charge Point SHALL NOT send request messages to the Central System unless it has been instructed by the Central System to do so with a TriggerMessage.req request.	**C**
FR.01a.006		A Charge Point SHOULD NOT send a BootNotification.req earlier than the value of the Interval field in the BootNotification.conf. unless requested to do so with a TriggerMessage.req.	**C**
FR.01a.007	If the Charge Point is configured to do.	While not yet accepted by the Central System, the Charge Point may allow locally- authorized transactions.	**O**

3.3. UC.01b - First Boot Charge Point (rejected)

No.	Type	Description
1	**Use case element name**	First Boot Charge Point (rejected)
2	**ID**	UC.01b
3	**Objectives**	To inform the Charge Point that the Central System does not accept the Charge Point (yet).+ The Charge Point should retry, as the Central System MAY accept the Charge Point at a later time.
	Actors	Charge Point, Central System

No.	Type	Description
	Scenario description	**1.** The Central System receives the status Rejected from the Charge Point. **2.** The Charge Point will resend the BootNotification.req every XX seconds (Interval from the BootNotification.conf) **3.** When the Central System returns with BootNotification.conf with the status accepted, UC.01 - Psychical Cold Boot Charge Point is valid.
	Alternative scenario's	n/a
4	**Prerequisites**	The Charge Point received the status *Rejected*.
5	**Post conditions**	Charge Point is in *Idle* state.
6	**Sequence diagram**	

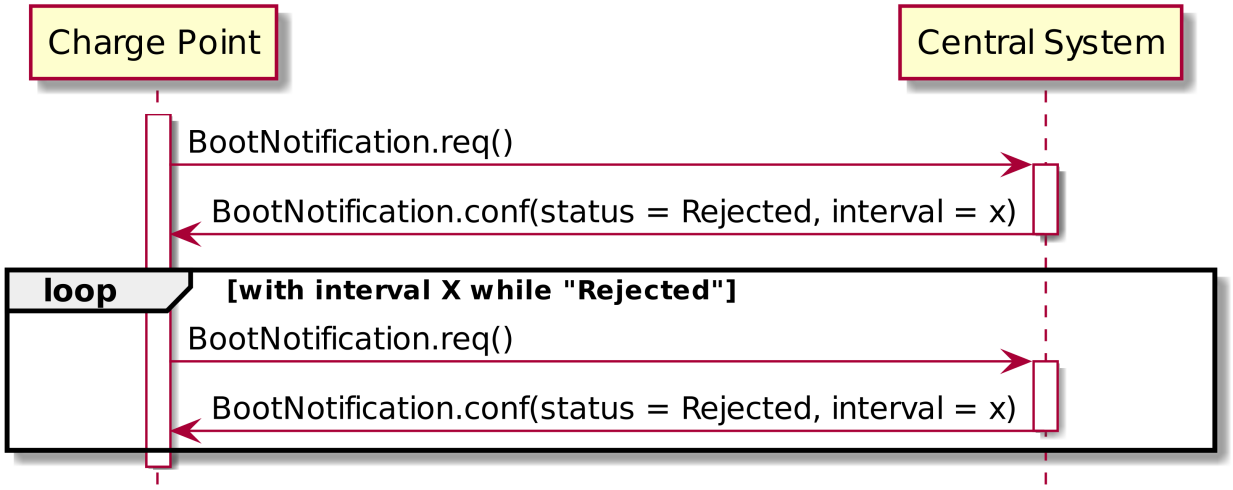

Figure 3. Sequence Diagram: ColdBootB

7	**Error handling**	CHECK
8	**Remarks**	During the state Rejected, the Charge Point may no longer be reachable from the Central System. It MAY for instance close its communication channel or shut down its communication hardware. Also the Central System MAY close the communication channel, for instance to free up system resources. CHECK

3.3.1. UC.01a - Requirements

ID.	Precondition	Requirement	M/O/C
FR.01b.001	If the Central System returns the status Rejected.	The Charge Point shall NOT send any OCPP message to the Central System until the aforementioned retry interval has expired.	**C**
FR.01b.002	While rejected	The Charge Point SHALL NOT respond to any Central System initiated message. the Central System SHOULD NOT initiate any.	**C**

ID.	Precondition	Requirement	M/O/C
FR.01b.003	Between the physical power-on/reboot and the successful completion of a BootNotification, where Central System returns Accepted or Pending.	The Charge Point shall NOT send any other request to the Central System.	**C**
FR.01b.004	If the Charge Point is configured to do. CHECK	While not yet accepted by the Central System, the Charge Point MAY allow locally- authorized transactions.	**O**

4. Messages

4.1. BootNotification.req

This contains the field definition of the BootNotification.req PDU sent by the Charge Point to the Central System. See also [boot-notification]

Field Name	Field Type	Card.	Description
chargeBoxSerialNumber	CiString25Type	0..1	Optional. This contains a value that identifies the serial number of the Charge Box inside the Charge Point. Deprecated, will be removed in future version
chargePointModel	CiString20Type	1..1	Required. This contains a value that identifies the model of the ChargePoint.
chargePointSerialNumber	CiString25Type	0..1	Optional. This contains a value that identifies the serial number of the Charge Point.
chargePointVendor	CiString20Type	1..1	Required. This contains a value that identifies the vendor of the ChargePoint.
firmwareVersion	CiString50Type	0..1	Optional. This contains the firmware version of the Charge Point.
iccid	CiString20Type	0..1	Optional. This contains the ICCID of the modem's SIM card.
imsi	CiString20Type	0..1	Optional. This contains the IMSI of the modem's SIM card.
meterSerialNumber	CiString25Type	0..1	Optional. This contains the serial number of the main power meter of the Charge Point.
meterType	CiString25Type	0..1	Optional. This contains the type of the main power meter of the Charge Point.

4.2. BootNotification.conf

This contains the field definition of the BootNotification.conf PDU sent by the Central System to the Charge Point in response to a BootNotification.req PDU. See also [boot-notification]

Field Name	Field Type	Card.	Description
currentTime	dateTime	1..1	Required. This contains the Central System's current time.
interval	integer	1..1	Required. When RegistrationStatus is *Accepted*, this contains the heartbeat interval in seconds. If the Central System returns something other than Accepted, the value of the interval field indicates the minimum wait time before sending a next BootNotification request.
status	RegistrationStatus	1..1	Required. This contains whether the Charge Point has been registered within the System Central.

4.3. Heartbeat.req

This contains the field definition of the Heartbeat.req PDU sent by the Charge Point to the Central System. See also [heartbeat]

No fields are defined.

4.4. Heartbeat.conf

This contains the field definition of the Heartbeat.conf PDU sent by the Central System to the Charge Point in response to a Heartbeat.req PDU. See also [heartbeat]

Field Name	Field Type	Card.	Description
currentTime	dateTime	1..1	Required. This contains the current time of the Central System.

4.5. StatusNotification.req

This contains the field definition of the StatusNotification.req PDU sent by the Charge Point to the Central System. See also [status-notification]

Field Name	Field Type	Card.	Description
connectorId	integer connectorId >= 0	1..1	Required. The id of the connector for which the status is reported. Id '0' (zero) is used if the status is for the Charge Point main controller.
errorCode	ChargePointErrorCode	1..1	Required. This contains the error code reported by the Charge Point.
info	CiString50Type	0..1	Optional. Additional free format information related to the error.
status	ChargePointStatus	1..1	Required. This contains the current status of the Charge Point.
timestamp	dateTime	0..1	Optional. The time for which the status is reported. If absent time of receipt of the message will be assumed.
vendorId	CiString255Type	0..1	Optional. This identifies the vendor-specific implementation.
vendorErrorCode	CiString50Type	0..1	Optional. This contains the vendor-specific error code.

4.6. StatusNotification.conf

This contains the field definition of the StatusNotification.conf PDU sent by the Central System to the Charge Point in response to an StatusNotification.req PDU. See also [status-notification]

No fields are defined.

5. DataTypes

5.1. ChargePointStatus

Enumeration

Status reported in StatusNotification.req. A status can be reported for the Charge Point main controller (connectorId = 0) or for a specific connector. Status for the Charge Point main controller is a subset of the enumeration: *Available*, *Unavailable* or *Faulted*.

States considered Operative are: *Available*, *Preparing*, *Charging*, *SuspendedEVSE*, *SuspendedEV*, *Finishing*, *Reserved*. States considered Inoperative are: *Unavailable*, *Faulted*.

Status	Condition
Available	When a Connector becomes available for a new user (Operative)
Preparing	When a Connector becomes no longer available for a new user but no charging session is active. Typically a Connector is occupied when a user presents a tag, inserts a cable or a vehicle occupies the parking bay (Operative)
Charging	When the contactor of a Connector closes, allowing the vehicle to charge (Operative)
SuspendedEVSE	When the contactor of a Connector opens upon request of the EVSE, e.g. due to a smart charging restriction or as the result of StartTransaction.conf indicating that charging is not allowed (Operative)
SuspendedEV	When the EVSE is ready to deliver energy but contactor is open, e.g. the EV is not ready.
Finishing	When a charging session has stopped at a Connector, but the Connector is not yet available for a new user, e.g. the cable has not been removed or the vehicle has not left the parking bay (Operative)
Reserved	When a Connector becomes reserved as a result of a Reserve Now command (Operative)

Status	Condition
Unavailable	When a Connector becomes unavailable as the result of a Change Availability command or an event upon which the Charge Point transitions to unavailable at its discretion. Upon receipt of a Change Availability command, the status MAY change immediately or the change MAY be scheduled. When scheduled, the Status Notification shall be send when the availability change becomes effective (Inoperative)
Faulted	When a Charge Point or connector has reported an error and is not available for energy delivery . (Inoperative).

5.2. ChargePointErrorCode

Enumeration

Charge Point status reported in StatusNotification.req.

Value	Description
ConnectorLockFailure	Failure to lock or unlock connector.
EVCommunicationError	Communication failure with the vehicle, might be Mode 3 or other communication protocol problem. This is not a real error in the sense that the Charge Point doesn't need to go to the faulted state. Instead, it should go to the SuspendedEVSE state.
GroundFailure	Ground fault circuit interrupter has been activated.
HighTemperature	Temperature inside Charge Point is too high.
InternalError	Error in internal hard- or software component.
LocalListConflict	The authorization information received from the Central System is in conflict with the LocalAuthorizationList.
NoError	No error to report.
OtherError	Other type of error. More information in vendorErrorCode.
OverCurrentFailure	Over current protection device has tripped.
OverVoltage	Voltage has risen above an acceptable level.
PowerMeterFailure	Failure to read power meter.

Value	Description
PowerSwitchFailure	Failure to control power switch.
ReaderFailure	Failure with idTag reader.
ResetFailure	Unable to perform a reset.
UnderVoltage	Voltage has dropped below an acceptable level.
WeakSignal	Wireless communication device reports a weak signal.

5.3. MeterValue

Class

Collection of one or more sampled values in MeterValues.req. All sampled values in a MeterValue are sampled at the same point in time.

Field Name	Field Type	Card.	Description
timestamp	dateTime	1..1	Required. Timestamp for measured value(s).
sampledValue	SampledValue	1..*	Required. One or more measured values

5.4. RegistrationStatus

Enumeration

Result of registration in response to BootNotification.req.

Value	Description
Accepted	Charge point is accepted by Central System.
Pending	Central System is not yet ready to accept the Charge Point. Central System may send messages to retrieve information or prepare the Charge Point.
Rejected	Charge point is not accepted by Central System. This may happen when the Charge Point id is not known by Central System.

6. Configuration Keys

6.1. `HeartbeatInterval`

Required/optional	required
Accessibility	RW
Type	int
Unit	seconds
Description	Interval of inactivity (no OCPP exchanges) with central system after which the Charge Point should send a Heartbeat.req PDU