DEGREE PROJECT, IN ELECTRIC POWER ENGINEERING , SECOND LEVEL
STOCKHOLM, SWEDEN 2015

OCPP Compatibility between a Central System and Electric Vehicle Charging Stations

ALEXANDRE COURT

XR-EE-ETK 2015:005

KTH ROYAL INSTITUTE OF TECHNOLOGY

ELECTRICAL ENGINEERING

Abstract

Nowadays, the growing CO2 emissions is one of the main international issues. The world is becoming aware that the current climate issues start being critic and that something has to be done. In parallel, Earth starts running out of fossil fuels so alternative energies and alternative ways of producing energy have to be found. Driving electric vehicles would reduce the CO2 emissions and the use of fossil fuels. Of course, it would not make it possible to solve all the current issues but it could be part of a global solution.

Over the past few years, the production of electric vehicle has grown faster and faster and consequently so did the production of electric vehicle charging stations. International and European standards have been set for electric vehicles and electrical vehicle charging stations. Besides, the growing number of charging stations entails a need of supervision. Supervision makes it possible for instance to control the charging stations remotely or to manage the transaction and the energy transmissions.

Given the large number of charging station constructors and supervision system suppliers, the need of a common communication protocol was imperative. The Open Charge Alliance (OCA) has developed a standard communication protocol named Open Charge Point Protocol (OCPP). This protocol is still in development but it enables the different actors to have a common communication protocol and so to possibly interconnect their systems.

Given OCPP is still under construction, the communication between a charging station and a supervision system is not trivial and some adjustments usually have to be made. The aim of this Thesis is to work on the compatibility between a supervision system and several charging stations from different constructors.

Key words: Charge Point, OCPP, EV, Central System, ConnectorId, ChargeBoxIdentity

Sammanfattning

Idag är växande CO2-utsläpp en av de viktigaste internationella frågorna. Världen är medveten om att de nuvarande klimatfrågorna börjar bli kritiskar och att något måste göras. Parallellt börjar jorden rinner ut av fossila bränslen så alternativa energikällor och alternativa sätt att producera energi måste hittas. Användningen av elfordon skulle minska CO2-utsläppen och användningen av fossila bränslen. Naturligtvis skulle det inte göra det möjligt att lösa alla aktuella frågor, men det kan vara en del av en global lösning.

Under de senaste åren har produktionen av elektriska fordon vuxit snabbare och snabbare och därmed också produktionen av elfordonens laddstationer. Internationella och europeiska standarder har fastställts för elfordon och elfordons laddstationer. Ett växande antal laddstationer medför ett behov av tillsyn. Tillsyn gör det möjligt att till exempel fjärrstyra de laddstationer eller att hantera transaktionen och energiöverföringar.

Med tanke på det stora antalet laddningsstation konstruktörer och övervakning systemleverantörer, var behovet av en gemensam kommunikationsprotokoll absolut nödvändigt. Open Charge Alliance (OCA) har utvecklat ett standard kommunikationsprotokoll som heter Open Charge Point Protocol (OCPP). Detta protokoll är fortfarande under utveckling, men det gör det möjligt för olika aktörer att ha ett gemensamt kommunikationsprotokoll och att eventuellt sammankoppla sina system.

Med tanke på att OCPP fortfarande är under uppbyggnad, är kommunikationen mellan en laddningsstation och ett övervakningssystem inte trivialt och vissa justeringar måste oftast göras. Syftet med dettaa examensarbete är att titta på förenlighet mellan ett övervakningssystem och flera laddningsstationer från olika leverantörer.

Nyckelord: Laddningspunkt, OCPP, Elfordon, Centralt System, AnslutningsId, LaddningsBoxId

Content

Abstract 2
Sammanfattning 4
Content 6
List of Acronyms 8
1 Introduction 10
1.1 Electric Vehicle Charging Mode and Plug Type 10
1.2 OCPP History [7] 15
1.3 OCPP Communication Protocol 15
1.4 Terminology and Conventions 15
1.5 Aim of the Master Thesis 16
1.6 Outline of the thesis 16
2 OCPP Specifications 18
2.1 General operations initiated by Charge Point 18
2.2 General operations initiated by Central System 26
3 Writing of the test protocol 34
3.1 Basic actions 34
3.2 Actions from the web-based management platform 34
3.3 Functions not available on the web platform 37
3.4 Additional tests 37
4 Configuration of the Communication Charge Point – Central System 40
4.1 Communication Charge Point – Central System 40
4.2 Configuration of the Communication 40
5 Compatibility Tests between the Central System and a Charge Point 42
5.1 Carrying out of the tests 42
5.2 Company A 42
5.3 Company B 48
5.4 Company C 56
6 Conclusion 58
Bibliography 60
Appendix A – Tests with Company A 62
Appendix B – Tests with Company B 80
Appendix C – Tests with Company C 92

List of Acronyms

AC	Alternative Current
DC	Direct Current
EV	Electrical Vehicle
EVSE	Electrical Vehicle Supply Equipment
HTTP	HyperText Transport Protocol
IEC	International Electrotechnical Commission
M2M	Machine-to-Machine
OCA	Open Charge Alliance
OCPP	Open Charge Point Protocol
PLC	Power-Line Communication
SOAP	Simple Object Access Protocol
URL	Uniform Resource Locator

Chapter 1

1 Introduction

1.1 Electric Vehicle Charging Mode and Plug Type

IEC 62196 [1] is the international standard for set of electrical connectors and charging modes for EV and is maintained by the International Electrotechnical Commission (IEC).

1.1.1 Charging modes

IEC 62196 defines four different charging modes.

1.1.1.1 Mode 1: Slow charging from a household-type socket-outlet

The EV is connected to the power grid through standard socket-outlets present in residences, which depending on the country varies from 8 to 16 A. The main risk with this charging mode is the heating of the socket and cables following intensive use for several hours at or near the maximum power. It is even more risky if the electrical installation is obsolete and if certain protective devices are absent. Consequently, mode 1 is not recommended because of these safety issues.

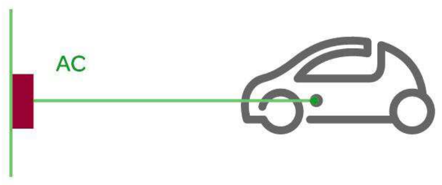


Figure 1.1: Mode 1: Fixed, non-dedicated socket

1.1.1.2 Mode 2: Slow charging from a household-type socket-outlet with control and protection device

The EV is connected to the main power grid via household socket-outlets. In this case, a protection device is built into the cable in order to avoid the problems mentioned above.

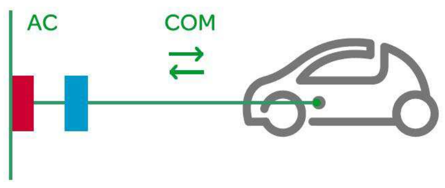


Figure 1.2: Mode 2 : Non-dedicated socket with cable-incorporated protection device

1.1.1.3 Mode 3: Slow or fast charging using a specific EV socket-outlet with control and protection function installed [2] [5]

The EV is connected directly to the electrical network via specific socket and plug and a dedicated circuit. A control and protection function is also installed permanently in the installation. This is the only charging mode that meets the applicable standards regulating electrical installations.

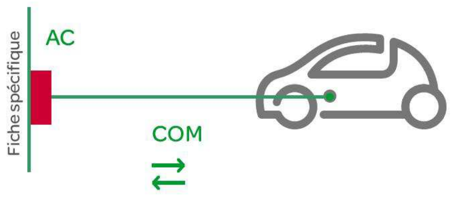


Figure 1.3 Mode 3 : Fixed, dedicated circuit-socket

1.1.1.4 Mode 4 : Fast charging using an external charger [3] [6]

The EV is connected to the main power grid throught an external charger. Control and protection functions and the vehicle charging cable are installed permanently in the installation.

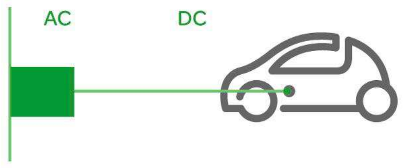


Figure 1.4: Mode 4: DC Connection

1.1.2 Plug Types

1.1.2.1 Plug Types for charging mode 1 and 2

Charging mode 1 and charging mode 2 entail the use of the domestic plugs and socket-outlets standardized in member countries of IEC.

1.1.2.2 Plug Types for charging mode 3

IEC 62196-2 defines three plug types for the charging mode 3.

<u>Type 1: Single phase vehicle coupler</u>

The SAE J1772-2009 (Type 1) connector, the North-American standard, is not used in Europe on the charging station side. However, this plug is the most common one on the EV side.

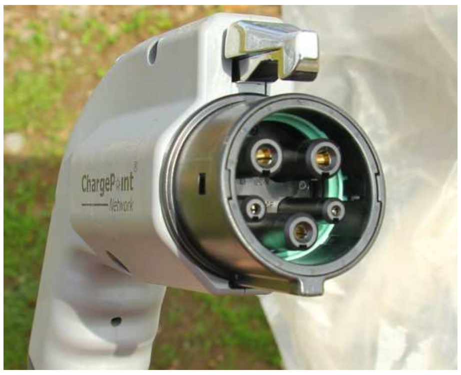


Figure 1.5: SAE J1772-2009 Plug (Type 1)

Type 2: Single and three phase vehicle coupler

The European Automobile Manufacturers (ACEA) has decided to use the Type 2 connector for deployment in European Union and is now the European standard. This plug has a single size and layout for currents from 16A single-phase up to 63A three-phase (3.7 kVA to 43.5 kVA). As it can be seen in the Figure 1.6, the plug Type 2 has 7 pins:

- "L1", "L2" and "L3" correspond to Phase 1, 2 and 3. With a three-phase charger, the three pins will be used. In case of a single-phase charger, only one pin will be used.
- "Neutral" corresponds to the neutral conductor.
- "Earth" corresponds to the link with the ground.
- "Proximity Pilot" and "Control Pilot" correspond to the control pins. "Proximity" checks and signals the maximum capacity of the cable. "Control Pilot" enables the communication with the EV and makes it possible to limit and control the charging current.

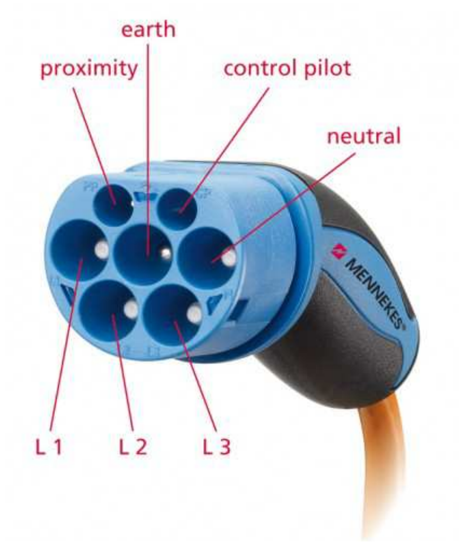


Figure 1.6: Plug Type 2

Type 3: Single and three phase vehicle coupler with shutters

The plug Type 3 was proposed by the EV Plug Alliance. The EV Plug Alliance was formed by the electrical French companies Schneider Electric and Legrand. Plug Type 3 was the standard in France for a long time but in January 2015, the French government decided to stop the use of this plug type and aligned with the European standard of plug Type 2.

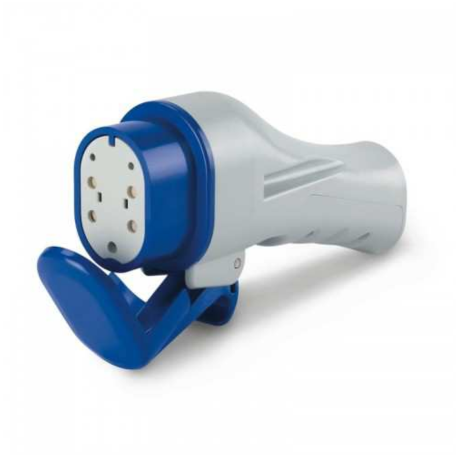


Figure 1.7: Plug Type 3

1.1.2.3 Plug Types for charging mode 4

IEC 62196-3 defines two plug types for the DC charging mode.

CHAdeMO

CHAdeMO is the name of a quick charging method for electric vehicle and became the name of the plug that can be seen in the Figure 1.8. The CHAdeMO can deliver up to 62.5 kW of high-voltage direct current via a special electrical connector.

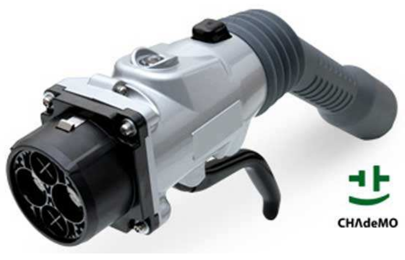


Figure 1.8: CHAdeMO Plug

Combined Charging System

The Combined Charging System usually abbreviated by COMBO is another DC charging plug. COMBO and CHAdeMO have nearly the same properties and are nearly equally used and currently none of them seems to overcome the other.

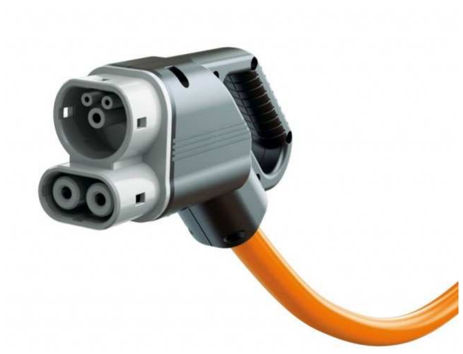


Figure 1.9: COMBO Plug

1.1.3 Electric Vehicles Charging Summary

One could think that in order to charge an EV, one has only to plug the EV to a random charger but in the reality, it is far more complicated. There are different charging modes and different plugs. Depending on the plug, the power available on the charger and the power accepted by the EV, the charging time will be different. The table 1.1 below sums up most of these parameters.

Socket on the charger side	**Charging Mode**	**Power and Currents**	**Approximate Charging time**	**Socket on the EV side**
Household socket	Mode 1	2.2 kVA 10 A single-phase	9 to 12 hours	Electric cable attached to the EV
	Mode 2	2.2 kVA 10 A single-phase	9 to 12 hours	Socket Type 2
Socket Type 2	Mode 3	3.7 kVA 16 A single-phase	6 to 8 hours	
		7 kVA 32 A single-phase	3 to 4 hours	
		22 kVA 32 A three-phase	1 hour	
No socket (Electric cable attached to the charger)	Mode 4	43 kVA 63 A three-phase	30 minutes	
		50 kVA 125 A DC	30 minutes	COMBO or CHAdeMO

Table 1.1: EV charging summary

1.2 OCPP History [7]

The Dutch foundation E-laad was founded by the Dutch grid owners in 2009. With the help of its initial partners Logica and Alfen, E-laad started the development of OCPP, an application protocol for the communication between EV charging stations and a central management system. The aim was to make EV charging stations and central management systems from different vendors able to communicate thanks to the standard protocol OCPP.

When E-laad released the first version of OCPP (OCPP 1.2), the EV market was not very developed. E-laad still provided a free version and simulation software to enable charge point manufacturers and central management system to implement OCPP. The interest in OCPP grew the next years. Both charge point manufacturers and central management system quickly realized the benefit in standardizing the protocol of communication.

In 2013, E-laad foundation (Netherlands), Greenlots (North America) and ESB (Ireland) founded the OCA. OCA is a global consortium of public and private EV infrastructure leaders that have come together to promote the adoption of a uniform communication method, OCPP.

OCA released a second version, OCPP 1.5 [8]. Important improvements have been done between OCCP 1.2 and OCPP 1.5 but there were still improvements needed to get a more mature protocol. They have been implemented in the newly released version, OCPP 2.0 [9]. Most of the current charging stations use OCPP 1.5 and are trying to move to OCPP 2.0.

1.3 OCPP Communication Protocol

OCPP 1.5 is a SOAP (Simple Object Access Protocol) based protocol communication. SOAP is an XML-based protocol. This protocol makes it possible to perform RPC (Remote Procedure Call) request-response dialogues. SOAP can operate over any transport protocol but it is mainly used with HTTP. The protocol, as of OCPP version 1.5, consists in 25 operations. 10 are initiated by the charging station and 15 are initiated by the central system

1.4 Terminology and Conventions

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.
Charge Point	The Charge Point is the physical system where an electric vehicle can be charged. A Charge Point will have one or more connectors.
Connector	The term “Connector”, as used in OCPP 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.

The above definitions are the only three used in OCPP 1.5 to define a charging station infrastructure. The vagueness of these terms leads to different interpretations as regard of OCPP users and consequently compatibility issues between a Central System and a Charge Point. OCPP 2.0 is much clearer but most of the charging stations constructors have not yet implemented it.

1.5 Aim of the Master Thesis

The Master Thesis consisted in a research work in the French company 'Bouygues Energies & Services'. This company has a partnership with a French start-up who is a member of OCA. This start-up is a charge point [4] constructor but develops also a web-based management platform. This platform makes it possible to manage charge points. Among the diverse functionalities are the possibility to control a charge point remotely and know in real-time its state ('Available', 'Occupied', 'Unavailable'…) and the energy transferred if a charging session is in progress. The start-up and consequently 'Bouygues Energies & Services' try to have the widest possible offer in term of compatibility between their platform and the charging stations from other constructors.

As said in the previous sub-part, OCPP 1.5 is rather unclear which leads to different implementations of the communication protocol. Consequently, even if OCPP has been designed to be vendor independent, making a Central System and a Charge Point compatible is complicated depending on the constructor interpretation.

The aim of the thesis is to make the start-up's Central System compatible with several charge point's constructors.

1.6 Outline of the thesis

Chapter 1 presents the background and the aim of this thesis.

Chapter 2 is a review of the OCPP protocol and presents the main OCPP functions.

Chapter 3 details the test protocol written for the compatibility tests between a charge point and Bouygues Energies & Services Central System.

Chapter 4 describes how to make a charge point and a Central System able to communicate.

Chapter 5 describes the results of the compatibility tests and the corrective actions implemented in order to make the charge points and the Central System compatible.

Chapter 6 presents the overall conclusion of the thesis.

Chapter 2

2 OCPP Specifications

This part is based on the fully detailed documentation of OCPP 1.5 [8] [10] [11] [12] and explains the 25 operations of OCPP 1.5 in order to understand the rest of the thesis. The functions are more or less detailed depending on their importance and their usefulness for the testing part. Some functions are not yet supported by the tested charge points so they are briefly explained. On the contrary, the main functions and their main parameters are more detailed. The first and the second parts describe the general operations initiated by the Charge Point and the Central System respectively.

2.1 General operations initiated by Charge Point

2.1.1 Authorize

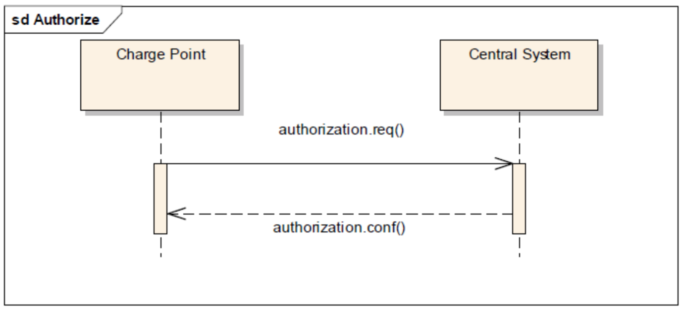


Figure 2.1: Authorize message

Before the owner of an electric vehicle can start or stop charging, the transaction has to be accepted. The Charge Point sends an Authorization.req to the Central System. The Central System shall respond with an Authorization.conf which indicates, whether or not the id-tag is accepted. The response must include an authorization status value indicating acceptance or a reason for rejection. The Charge Point will unlock the connector only after the authorization.

Message	Field Name	Description
Authorization.req	idTag	Contains the identifier that needs to be authorized.
Authorization.conf	idTagInfo	Contains information about authorization status, expiry and group id.

Table 2.1: Authorize message main parameters

Note:
If the id-tag for stopping is the same as the one for starting, the Charge Point shall stop the transaction without sending another Authorization.req.
A Charge Point may cache previously authorized id-tag in the local authorization list and may use these to authorize a user. When the owner of an electric vehicle tries to start a transaction, the local authorization list shall be checked first. If the id-tag is not present, then the Charge Point shall send an Authorization.req for requesting authorization.

2.1.2 Boot Notification

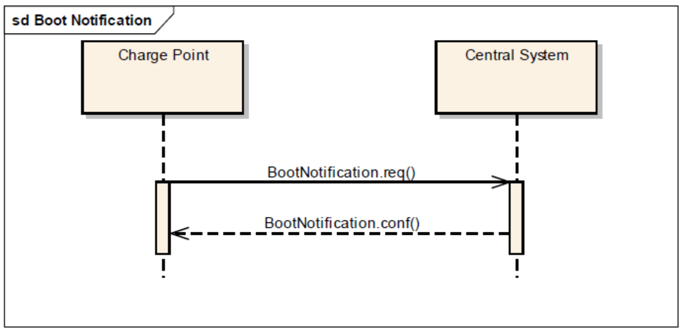


Figure 2.2: BootNotification message

After start-up a Charge Point sends a notification to the Central System with information about its configuration. The Central System will only accept Charge Points that are registered in the Central System. The Charge Point shall send a BootNotification.req each time it (re-) boots. The Central System shall respond with a BootNotification.conf which indicates whether or not the Central System accepts the Charge Point. If the Charge Point has been accepted, the response shall contain the Central System's current time and heartbeat interval.

Message	Field Name	Description
BootNotification.req	chargePointModel	Contains a value that identifies the model of the Charge Point
	chargePointVendor	Contains a value that identifies the vendor of the Charge Point
	firmwareVersion	Contains the firmware version of the Charge Point
BootNotification.conf	currentTime	Contains the Central System's current time
	heartbeatInterval	Contains the interval in seconds of the heartbeats
	status	Contains whether the Charge Point has been registered within the Central System

Table 2.2: BootNotification message main parameters

Part of BootNotification.req message:

```
<cs:bootNotificationRequest>
    <cs:chargePointModel>ModelX</cs:chargePointModel>
    <cs:chargePointVendor>VendorX</cs:chargePointVendor>
    <cs:firmwareVersion>3.4.2</cs:firmwareVersion>
</cs:bootNotificationRequest>
```

Part of BootNotification.conf message:

```
<cs:bootNotificationResponse>
    <cs:currentTime>2014-12-03T15:17:18.945Z</cs:currentTime>
    <cs:heartbeatInterval>300</cs:heartbeatInterval>
    <cs:status>Accepted</cs:status>
</cs:bootNotificationResponse>
```

2.1.3 Data Transfer

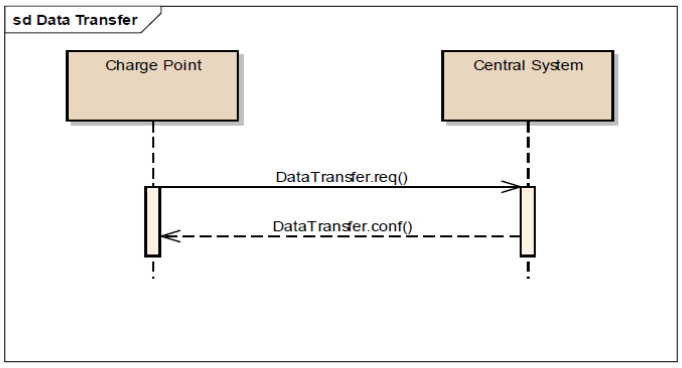
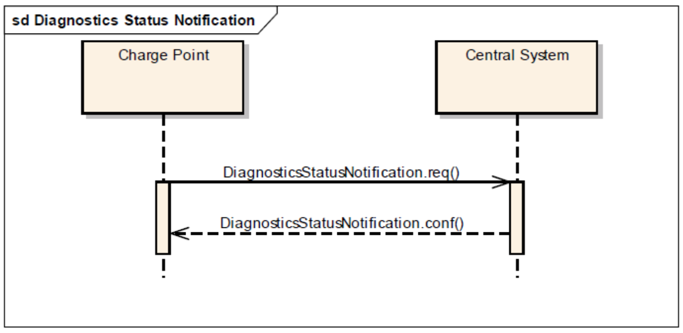


Figure 2.3: DataTransfer message

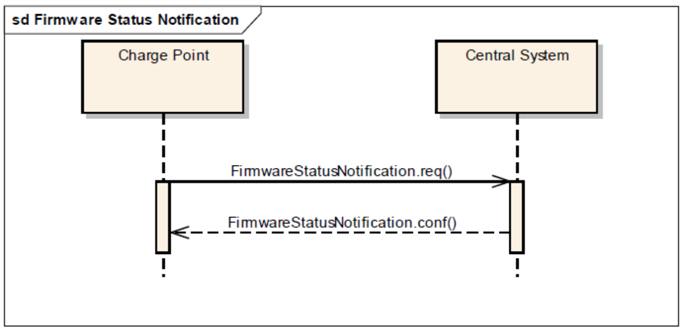
The function DataTransfer shall be used if the Charge Point needs to send information to the Central System for a function not supported by OCPP.

2.1.4 Diagnostics Status Notification


sd Diagnostics Status Notification
Charge Point
Central System
DiagnosticsStatusNotification.req()
DiagnosticsStatusNotification.conf()
Figure 2.4: DiagnosticsStatusNotification message

With the function GetDiagnostics, the Central System can request a Charge Point for diagnostic information. The Charge Point shall send a DiagnosticsStatusNotification.req to inform the Central System that the upload of diagnostics has finished successfully or failed. The Central System shall respond with a DiagnosticsStatusNotification.conf to inform the Charge Point that the message has been well received.

2.1.5 Firmware Status Notification


sd Firmware Status Notification
Charge Point
Central System
FirmwareStatusNotification.req()
FirmwareStatusNotification.conf()
Figure 2.5: FirmwareStatusNotification message

With the function UpdateFirmware, the Central System can notify a Charge Point that it needs to update its firmware. The Charge Point shall send a FirmwareStatusNotification.req to inform the Central System about the progress of the installation of the firmware update. The Central System shall respond with a FirmwareStatusNotification.conf to inform the Charge Point that the message has been well received.

2.1.6 Heartbeat

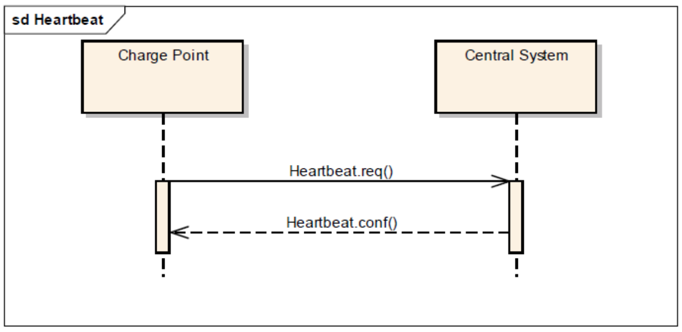


Figure 2.6: Heartbeat message

The function Heartbeat is used to let the Central System know that a Charge Point is still connected. A Charge Point sends a heartbeat after a configurable time interval. The default heartbeat interval is the one defined by the Central System in its response to the BootNotification. The Central System shall respond to the Heartbeat.req with a Heartbeat.conf which shall contain the current time of the Central System. Hence the Charge Point can synchronize its internal clock.

Message	Field Name	Description
Heartbeat.req	No fields are defined	
Heartbeat.conf	currentTime	Contains the current time of the Central System

Table 2.3: Heartbeat message main parameters

2.1.7 Meter Values

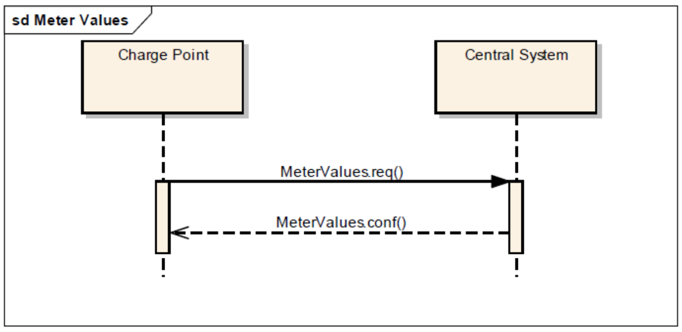


Figure 2.7: MeterValues message

If the Charge Point is able to sample the electricity meter, it shall use a MeterValues.req to upload meter values. The Central System shall respond with a MeterValues.conf to inform the Charge Point that the message has been well received.

If meter values are related to a transaction then the **transactionId** shall be included in the MeterValues.req. A transaction is related to a specific **connectorId** of the Charge Point so the **connectorId** shall be included in the MeterValues.req.

Message	Field Name	Description
MeterValues.req	connectorId	Contains a number designating a connector of the Charge Point
	transactionId	The transaction to which these meter samples are related
	values	The sampled meter values with timestamps
MeterValues.conf	No fields are defined	

Table 2.4: MeterValues message main parameters

A MeterValues.req contains one or more **values** elements. Each **values** element contains a timestamp and a set of one or more individual **value** elements, all captured at the same point in time. The nature of each **value** is determined by the optional attributes. **Measurand** and **unit** are the two main optional attributes.

The optional **unit** attribute defines the unit of the measurand value. The default unit is “Wh” if the default measurand is an “Energy” type.

The optional **measurand** attribute specifies the type of value being measured. The default measurand value is “Energy.Active.Import.Register”. The main **measurand** values with their description can be seen on the table below.

Measurand values	Description
Energy.Active.Import.Register	Energy imported by EV (Wh of kWh)
Power.Active.Import	Instantaneous active power imported by EV (W or kW)
Current.Import	Instantaneous current flow to EV (A)
Voltage	AC RMS supply voltage (V)
Temperature	Temperature reading inside the charge point

Table 2.5: Values of the measurand attribute

Part of a MeterValues.req message:

```
<cs:meterValuesRequest>
  <cs:connectorId>0</cs:connectorId>
  <cs:transactionId>170</cs:transactionId>
  <cs:values>
    <cs:timestamp>2014-12-03T10:52:59.410Z</cs:timestamp>
    <cs:value cs:measurand="Current.Import" cs:unit="Amp">41.384</cs:value>
    <cs:value cs:measurand="Voltage" cs:unit="Volt">226.0</cs:value>
    <cs:value cs:measurand="Power.Active.Import" cs:unit="W">7018</cs:value>
    <cs:value cs:measurand="Energy.Active.Import.Register"
cs:unit="Wh">2662</cs:value>
    <cs:value cs:measurand="Temperature" cs:unit="Celsius">24</cs:value>
  </cs:values>
</cs:meterValuesRequest>
```

Note:

The MeterValues.req is a transaction-related message. In an offline situation the charge Point shall queue messages that need to be sent to the Central System and shall transmit transaction-related messages in chronological order as soon as the connection to the Central System is restored.

2.1.8 Start Transaction

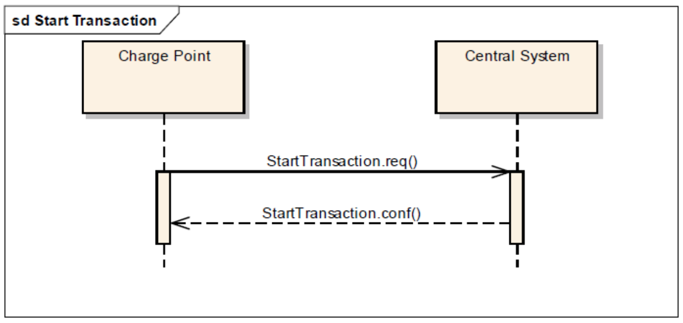


Figure 2.8: StartTransaction message

When an electric vehicle is allowed to start charging, the Charge Point needs to inform the Central System about this. The Charge Point shall send a StartTransaction.req to the Central System to inform it about the start of a charging transaction. The Central System must verify validity of the transaction. Indeed, the identifier might have been authorized locally by the Charge Point using an out-of-date white list while he may have been blocked since he was added to the local white list.

The Central System shall respond to the StartTransaction.req with a StartTransaction.conf. The response must include a transaction id and an authorization status value. With an authorization status value other than 'Accepted', the Charge Point must stop the transaction.

Message	Field Name	Description
StartTransaction.req	connectorId	Identifies which connector of the Charge Point is used
	idTag	Contains the identifier for which a transaction has to be started
	meterStart	Contains the meter value in Wh for the connector at start of the transaction
	timestamp	Contains the date and time on which the transaction is started
StartTransaction.conf .	idTagInfo	Contains information about authorization status, expiry and group id
	transactionId	Contains the transaction id supplied by the Central System

Table 2.6: StartTransaction message main parameters

Part of StartTransaction.req message:

```
<cs:startTransactionRequest>
    <cs:connectorId>255</cs:connectorId>
    <cs:idTag>A6DA6EF0</cs:idTag>
    <cs:meterStart>16365</cs:meterStart>
    <cs:timestamp>2014-11-27T13:32:12.000Z</cs:timestamp>
</cs:startTransactionRequest>
```

Part of StartTransaction.conf message:

```
<cs:startTransactionResponse>
    <cs:idTagInfo>
       <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>171</cs:transactionId>
</cs:startTransactionResponse>
```

<u>Note:</u>
In an offline situation the Charge Point shall queue messages that need to be sent to the Central System and shall transmit transaction-related messages in chronological order as soon as the connection to the Central System is restored.

2.1.9 Status Notification

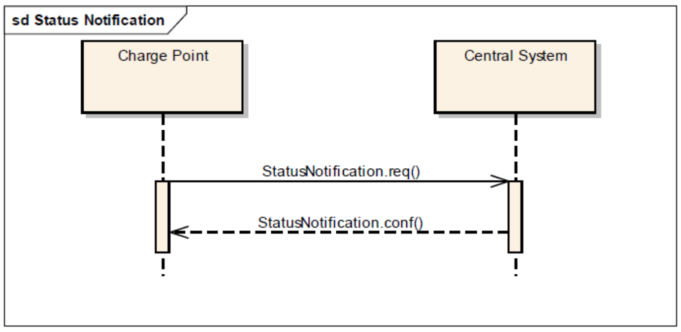


Figure 2.9: StatusNotification message

The Charge Point shall send a StatusNotification.req to the Central System each time it changes its status. The Central System shall respond with a StatusNotification.conf to inform the Charge Point that the message has been well received.

Message	Field Name	Description
StatusNotification.req	connectorId	The id of the connector for which the status is reported. Id ‘0’ is used if the status is for the Charge Point as a whole
	errorCode	Contains the error code reported by the Charge Point
	Info	Additional free format information
	Status	Contains the current status of the Charge Point
	Timestamp	Contains the time for which the status is reported
StatusNotification.conf	No fields are defined	

Table 2.8: StatusNotification message main parameters

Status	Description
Available	At least one connector is available for charging (Operative)
Occupied	All connectors of the Charge Point are occupied (Operative)
Reserved	All connectors of the Charge Point are reserved (Operative)
Unavailable	Charge Point or connector is not available for charging or it has been explicitly set to unavailable (Inoperative)
Faulted	The Charge Point or connector has reported an error and is no longer available (Inoperative)

Table 2.7: Charge point possible statuses

Part of StatusNotification.req message:

```
<cs:statusNotificationRequest>
    <cs:connectorId>0</cs:connectorId>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:info>Charging</cs:info>
    <cs:status>Occupied</cs:status>
    <cs:timestamp>2014-12-03T15:23:14.662Z</cs:timestamp>
</cs:statusNotificationRequest>
```

2.1.10 Stop Transaction

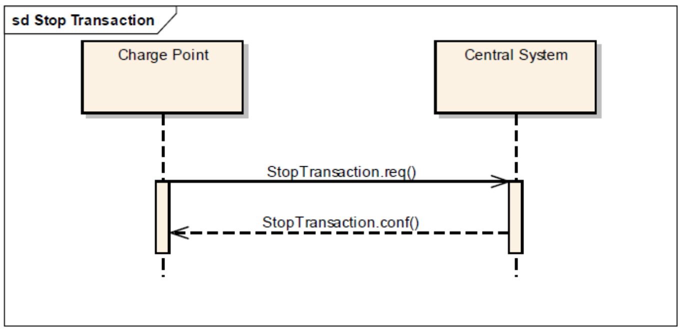


Figure 2.10: StopTransaction message

The Charge Point shall send a StopTransaction.req when a transaction is stopped. A transaction must be stopped explicitly by the Charge Point. The Central System shall respond with a StopTransaction.conf and always stop the transaction.

Message	Field Name	Description
StopTransaction.req	meterStop	Contains the meter value in Wh for the connector at end of the transaction
	Timestamp	Contains the date and time on which the transaction is stopped
	transactionId	Contains the transaction id as received by the StartTransaction.conf
StopTransaction.conf	No mandatory fields	

Table 2.9: StopTransaction message main parameters

Part of StopTransaction.req message:

```
<cs:stopTransactionRequest>
    <cs:meterStop>16365</cs:meterStop>
    <cs:timestamp>2014-11-27T13:32:14.000Z</cs:timestamp>
    <cs:transactionId>157</cs:transactionId>
</cs:stopTransactionRequest>
```

Note:

In an offline situation the Charge Point shall queue messages that need to be sent to the Central System and shall transmit transaction-related messages in chronological order as soon as the connection to the Central System is restored.

2.2 General operations initiated by Central System

2.2.1 Cancel Reservation

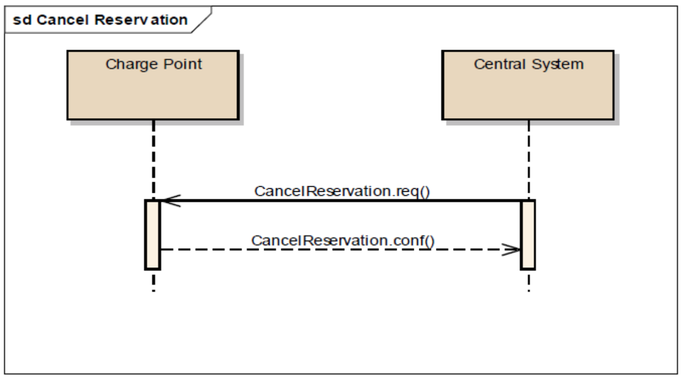


Figure 2.11: CancelReservation message

The Central System shall send a CancelReservation.req to the Charge Point to cancel a reservation. If the Charge Point has a reservation matching the reservationId in the request, it shall return status ‘Accepted’. Otherwise it shall return ‘Rejected’.

2.2.2 Change Availability

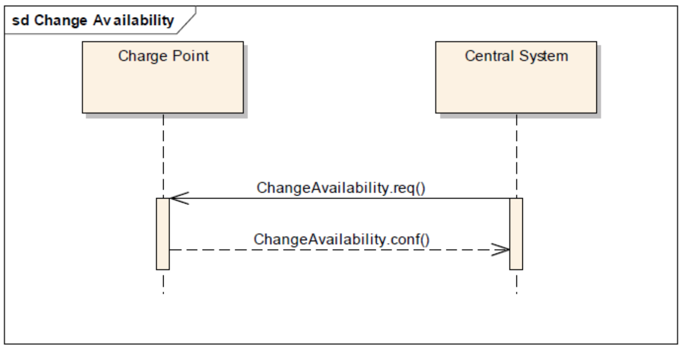


Figure 2.12: ChangeAvailability message

The Central System shall send a ChangeAvailability.req for requesting a Charge Point to change its availability. The Central System can change the availability to available or unavailable. Available (“Operative”) means that the Charge Point is charging or ready for charging. Unavailable (“Inoperative”) means that the Charge Point does not allow any charging. The Charge Point shall respond with a ChangeAvailabitily.conf to indicate whether or not it is able to change the availability.

<u>Note:</u>
If a transaction is in progress when the Central System send a ChangeAvailability.req, the Charge Point shall respond with the availability status ‘Scheduled’ to indicate that it is scheduled after the transaction has finished.

If the Central System requests the Charge Point to a status it is already in, the Charge Point shall respond with availability status ‘Accepted’.

Message	Field Name	Description
ChangeAvailability.req	connectorId	The id of the connector for which availability needs to change. Id '0' is used if the availability of the charge point as a whole needs to change
	availabilityType	Contains the type of availability change that the Charge Point should perform ('Operative' or 'Inoperative')
ChangeAvailabitily.conf	status	Indicates whether the Charge Point can perform the availability change

Table 2.10: ChangeAvailability message main parameters

2.2.3 Change Configuration

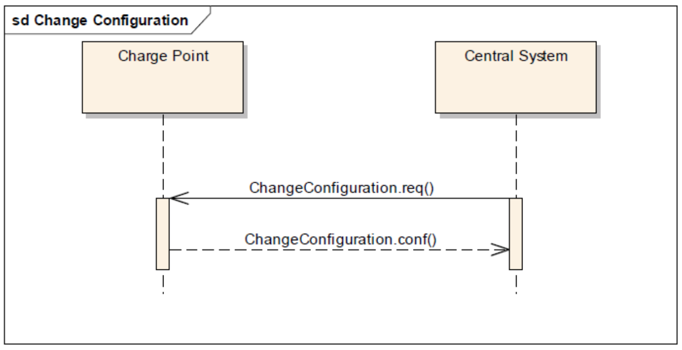


Figure 2.13: ChangeConfiguration message

The Central System shall send a ChangeConfiguration.req for requesting a Charge Point to change configuration parameters. This request contains a "key" and a "value".

The Charge Point shall respond with a ChangeConfiguration.conf and indicate either success ('Accepted') or failure ('Rejected') or that the "key" does not correspond to a configuration setting supported by the Charge Point ('NotSupported').

Message	Field Name	Description
ChangeConfiguration.req	key	The name of the configuration setting to change
	value	The new value as string for the setting
ChangeConfiguration.conf	status	Returns whether the configuration change has been accepted

Table 2.11: ChangeConfiguration message main parameters

2.2.4 Clear Cache

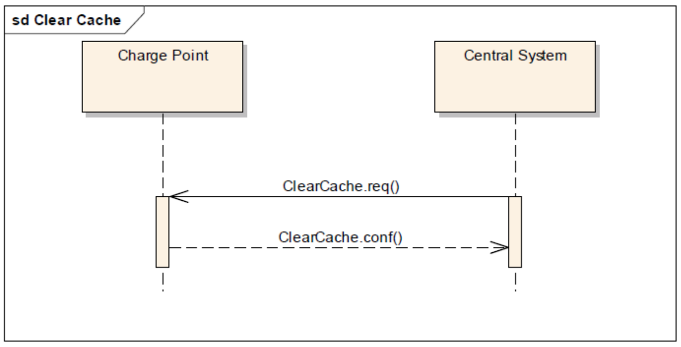


Figure 2.14: ClearCache message

The Charge Point saves the id tag used for the previous transactions in its cache. The Central System shall send a ClearCache.req for requesting the Charge Point to clear its cache. The Charge Point shall respond with a ClearCache.conf to indicate whether the Charge Point was able to clear its cache.

2.2.5 Data Transfer

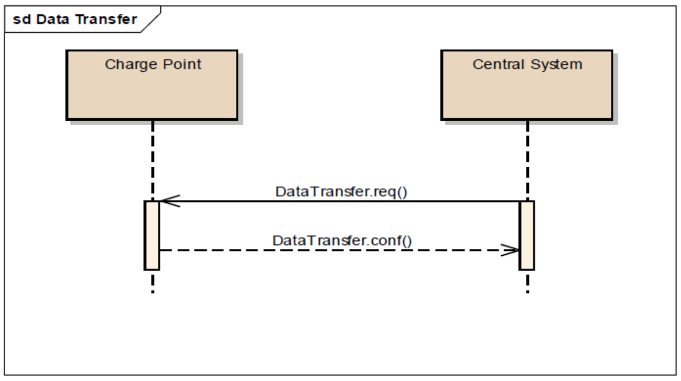


Figure 2.15: DataTransfer message

The function DataTransfer shall be used if the Central System needs to send information to the Charge Point for a function not supported by OCPP.

2.2.6 Get Configuration

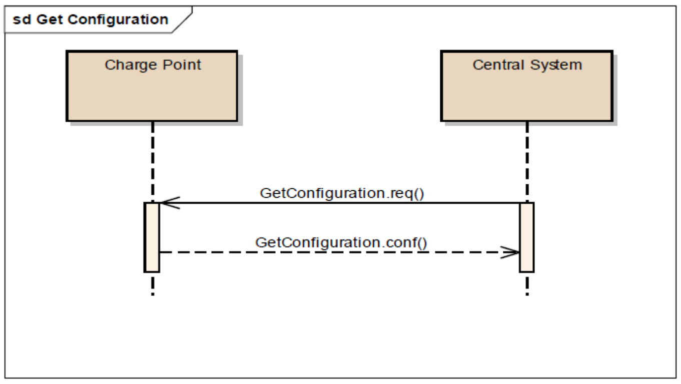


Figure 2.16: GetConfiguration message

The Central System shall send a GetConfiguration.req to the Charge Point to retrieve the value of configuration settings. The Charge Point shall respond with a GetConfiguration.conf and indicate the recognized keys with their corresponding values and the unrecognized keys with the value 'Unknown'.

2.2.7 Get Diagnostics

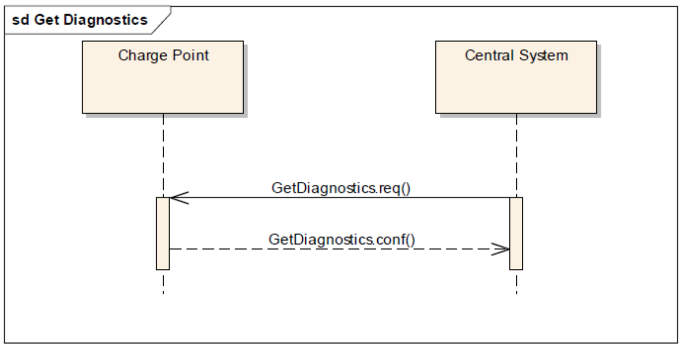


Figure 2.17: GetDiagnostics message

With the function GetDiagnostics, the Central System can request a Charge Point for diagnostic information. The Central System shall send a GetDiagnostics.req for getting diagnostic information specifying the location where the Charge Point must upload its diagnostic data. The Charge Point shall respond with a GetDiagnostic.conf specifying the name of the file that will be uploaded.

2.2.8 Get Local List Version

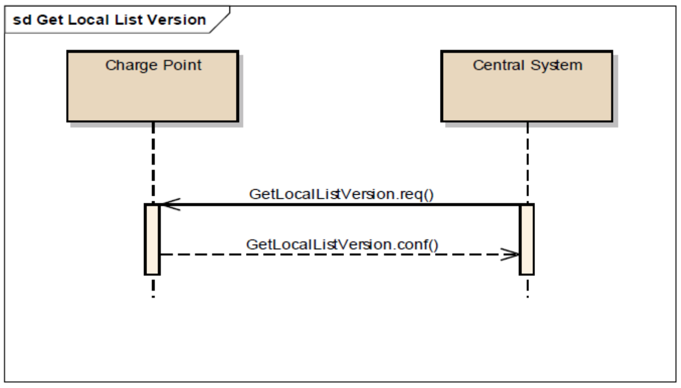

sd Get Local List Version
Charge Point
Central System
GetLocalListVersion.req()
GetLocalListVersion.conf()

Figure 2.18: GetLocalListVersion message

The Central System shall send a GetLocalListVersion.req to request the version number of the local authorization list. The Charge Point shall respond with a GetLocalListVersion.conf and indicate the version number of its local authorization list.

2.2.9 Remote Start Transaction

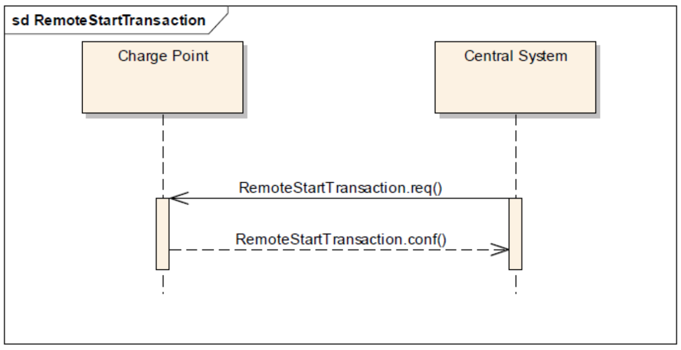

sd RemoteStartTransaction
Charge Point
Central System
RemoteStartTransaction.req()
RemoteStartTransaction.conf()

Figure 2.19: RemoteStartTransaction message

The Central System can request a Charge Point to start a transaction by sending a RemoteStartTransaction.req. The Charge Point shall respond with a RemoteStartTransaction.conf to indicate whether it is able to start the transaction or not.

Message	Field Name	Description
RemoteStartTransaction.req	connectorId	Number of the connector on which to start the transaction
	idTag	The identifier that the Charge Point must use to start the transaction
RemoteStartTransaction.conf	Status	Indicates whether the Charge Point accepts the request to start a transaction

Table 2.12: RemoteStartTransaction message main parameters

2.2.10 Remote Stop Transaction

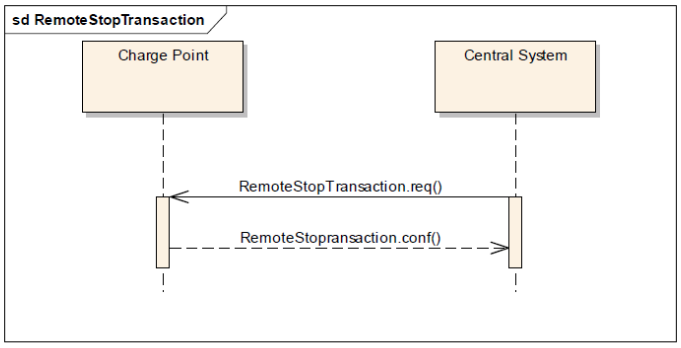


Figure 2.20: RemoteStopTransaction message

The Central System can request a Charge Point to stop a transaction by sending a RemoteStopTransaction.req to the Charge Point with the identifier of the transaction. The Charge Point shall respond with a RemoteStopTransaction.conf to indicate whether it is able to stop the transaction or not.

Message	Field Name	Description
RemoteStopTransaction.req	transactionId	The identifier of the transaction which the Charge Point is requested to stop
RemoteStopTransaction.conf	Status	Status indicating whether the Charge Point accepts the request to stop a transaction

Table 2.13: RemoteStopTransaction message main parameters

2.2.11 Reserve Now

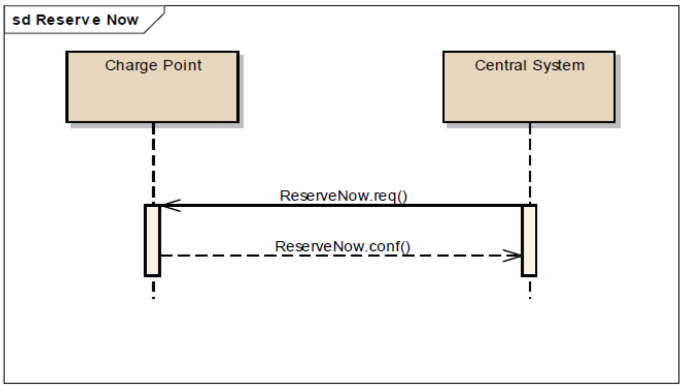


Figure 2.21: ReserveNow message

To request a reservation the Central System shall send a ReserveNow.req to a Charge Point. The Charge Point shall respond with a ReserveNow.conf.

Messages	Field Name	Description
ReserveNow.req	connectorId	Contains the id of the connector to be reserved. A value of 0 means that the reservation is not for a specific connector
	expiryDate	Contains the date and time when the reservation ends
	idTag	The identifier for which the Charge Point has to reserve a connector
	reservationId	Unique id for this reservation
ReserveNow.conf	Status	Indicates the success or failure of the reservation

Table 2.12: ReserveNow message main parameters

2.2.12 Reset

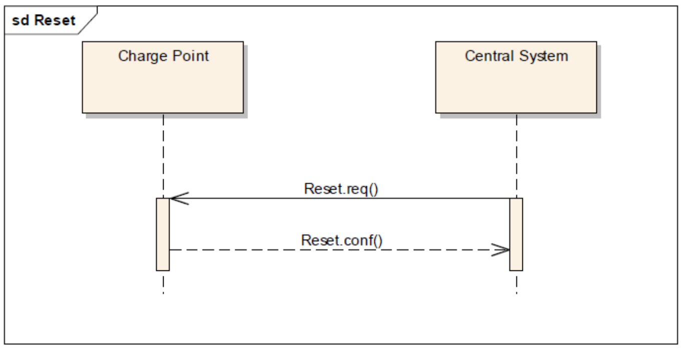


Figure 2.22: Reset message

The Central System shall send a Reset.req for requesting a Charge Point to reset itself. The Central System can request a hard or a soft reset. The Charge Point shall respond with a Reset.conf and indicate whether it is willing to reset itself. At receipt of a soft request, the Charge Point shall return to its basic idle state. At receipt of a hard request, the Charge Point shall reboot. For both soft and hard requests, any transaction in progress shall be terminated immediately.

2.2.13 Send Local List

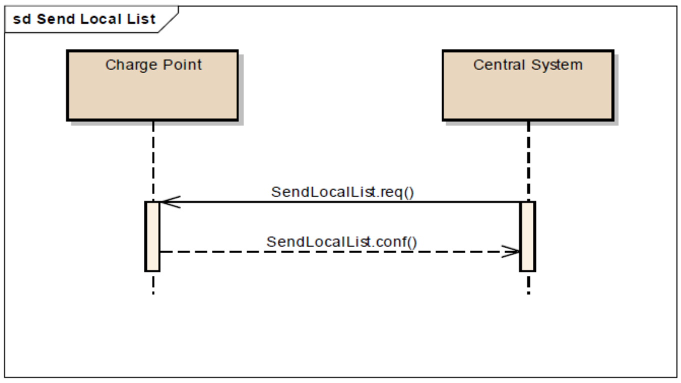


Figure 2.23: SendLocalList message

The Central System shall send a SendLocalList.req to send a local authorization list to the Charge Point. The Charge Point shall respond with a SendLocalList.conf and indicate whether it has accepted the update of the local authorization list.

2.2.14 Unlock Connector

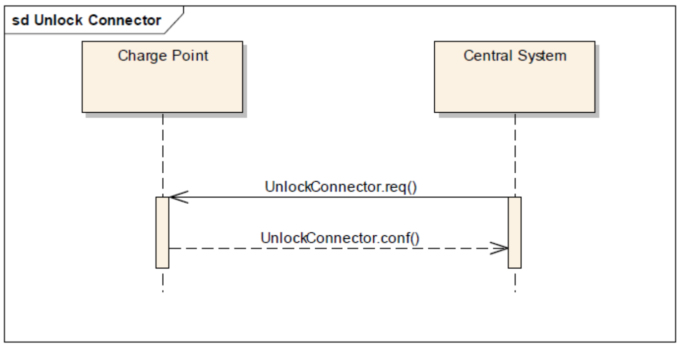


Figure 2.24: UnlockConnector message

The Central System can unlock a connector of a Charge Point by sending an UnlockConnector.req to the Charge Point. The Charge Point shall respond with an UnlockConnector.conf and indicate whether or not it was able to unlock its connector.

2.2.15 Update Firmware

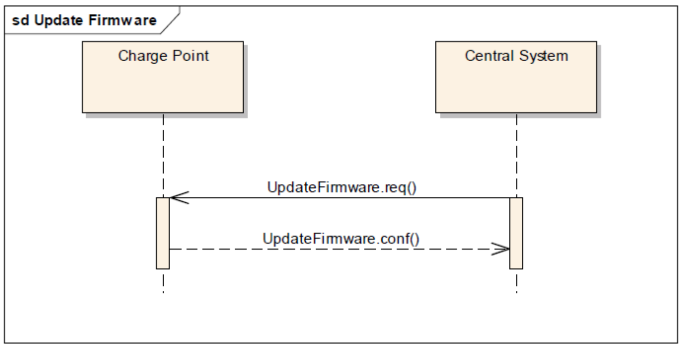


Figure 2.25: UpdateFirmware message

The Central System shall send an UpdateFirmware.req to make the Charge Point update its firmware. The Charge Point shall respond with an UpdateFirmware.conf to inform the Central System that the message has been well received.

3 Writing of the test protocol

Before starting the testing phase, a test protocol was written in order to define which actions will be tested and which OCPP messages will be expected after each of these actions. This document summed up below was then sent to the different constructors with whom the tests were done.

3.1 Basic actions

3.1.1 Start-up of the charge point and the communication

- Turn on the charge point and start the communication between the charge point and the server.

  Expected OCPP messages:
  - *BootNotification*
  - *StatusNotification*
  - *HeartBeat*

3.1.2 Regular charging session

- Present identification with a swipe card saved in the data base and start a charging session.

  Expected OCPP messages:
  - *Authorize →Accepted*
  - *StatusNotification →Occupied*
  - *StartTransaction*
  - *MeterValues*

- Present identification with a swipe card saved in the data base but different from the first one.

  Expected OCPP messages:
  - *Authorize →Invalid*

- Present identification with the swipe card used for starting the charging session.

  Expected OCPP messages:
  - *StopTransaction*
  - *StatusNotification →Available*

3.2 Actions from the web-based management platform

3.2.1 Start a charging session

- Start a charging session from the web-based management platform for a swipe card saved in the data base.

<u>*Expected OCPP messages*</u>*:*
- *RemoteStartTransaction*
- *StatusNotification → Occupied*
- *StartTransaction*
- *MeterValues*

3.2.2 Stop a charging session

- Stop a charging session from the web platform in the two following cases:
    - The session has been started with a RemoteStartTransaction from the management platform
    - The session has been started with a regular identification with a swipe card

<u>*Expected OCPP messages*</u>*:*
- *RemoteStopTransaction*
- *StopTransaction*
- *StatusNotification → Available*

3.2.3 Change the availability of the Charge Point

Change the availability of the charge point to 'Inoperative' in the two following cases:

3.2.3.1 The Charge Point is initially 'Available'

- The Charge Point is initially 'Available' and a ChangeAvailability is sent to move the Charge Point to the state 'Unavailable'. The charge point must directly change its availability and become 'Unavailable'.

<u>*Expected OCPP messages*</u>*:*
- *StatusNotification → Unavailable*

3.2.3.2 The Charge Point is initially 'Occupied'

- The charge point is initially 'Occupied' because a charging session is in progress. A ChangeAvailability is sent. The two following cases are possible:
    - The charging session in progress keeps going. The charge point becomes 'Unavailable' after the end of the current session.
    - The charging session in progress stops and the charge point becomes 'Unavailable'.

<u>*Expected OCPP messages*</u>*:*
- *StopTransaction*
- *StatusNotification → Unavailable*

3.2.4 Reservation

3.2.4.1 Reservation from the web platform

- Perform a reservation from the web platform for the swipe card number X.
- Present identification with the swipe card number Y → Swipe card refused

- Present identification with the swipe card number X → Swipe card accepted

  Expected OCPP messages:
  - *ReserveNow*
  - *StatusNotification → Reserved*
  - *Authorize → Invalid (badge n° Y)*
  - *Authorize → Accepted (badge n° X)*
  - *StatusNotification → Occupied*
  - *StartTransaction*

3.2.4.2 Cancellation from the web platform

- Perform a reservation from the web platform and then cancel this reservation from the web platform as well.

  Expected OCPP messages:
  - *ReserveNow*
  - *StatusNotification → Reserved*
  - *CancelReservation*
  - *StatusNotification → Available*

3.2.5 UnlockConnector

3.2.5.1 Charge Point 'Available'

Perform an UnlockConnector while the charge point is 'Available'.

3.2.5.2 Charge Point 'Occupied'

- Perform a UnlockConnector while there is a transaction in progress
  Does the Charge Point stop the transaction in progress immediately?

  Expected OCPP messages:
  - *UnlockConnector*
  - *StopTransaction*
  - *StatusNotification → Available*

3.2.6 ClearCache

- Perform a ClearCache

  Expected OCPP messages:
  - *ClearCache*

3.2.7 Reset

- Perform a Reset of the charge point.

  Expected OCPP messages:
  - *Reset*
  - *BootNotification*
  - *StatusNotification → Available*

3.3 Functions not available on the web platform

For the functions which cannot be performed from the web-based management platform, HTTP requests can be directly sent.

3.3.1 ChangeConfiguration

3.3.1.1 HeartbeatInterval - MeterValueSampleInterval

- Change the HeartbeatInterval and the MeterValueSampleInterval

Expected OCPP messages:
- *ChangeConfiguration*

3.3.1.2 MeterValuesSampledData

- The default value the charge point must send in the MeterValues is the « Energy.Active.Import.Register ».
- If the charge point has the required meter, a MeterValuesSampledData can be sent to get other values such as « Current.Import », « Voltage », « Power.Active.Import », « Temperature »…

Expected OCPP messages:
- *ChangeConfiguration*

3.3.2 GetConfiguration

3.3.2.1 HeartbeatInterval - MeterValueSampleInterval

- Perform a GetConfiguration in order to receive the HeartbeatInterval and the MeterValueSampleInterval

Expected OCPP messages:
- *ChangeConfiguration*

3.3.2.2 GetConfiguration empty

- If one sends a GetConfiguration empty, does the charge point respond with the list of all its settings configuration?

3.4 Additional tests

3.4.1 Simulate a communication failure

- Start a charging session
- Turn off the communication between the charge point and the web platform
- Stop the charging session in progress
- Perform several charging sessions
- Turn on the communication

Have the sessions that occurred during the communication failure and the MeterValues been well sent after the communication recovery?

3.4.2 Swipe card with different id-tag

- Tests can be performed to check that there is not any problem whatever the number of digit the swipe card has.

4 Configuration of the Communication Charge Point – Central System

4.1 Communication Charge Point – Central System

The communication between the charge point and the server of the Central System is most of the time a 3G connection. A router equipped with a M2M SIM card can be directly installed inside the charge point. Sometimes when several charge points are closed, they are all connected to a single router. The connection between the charge points and the router can be done using different technologies. Ethernet, PLC or ZigBee (radio communication) are three different options. When several charge points are closed, it is less expensive to use only one router. Indeed, first, only one router will have to be installed and paid. Besides, even if several charge points means more data which will transit, the fixed cost of the subscription package for the SIM card will only have to be paid once.

4.2 Configuration of the Communication

In order to make the web-based management platform and a charge point able to communicate, some settings configuration have to be done on both side.

4.2.1 Charge Point Configuration

From the charge point side, the constructor only has to configure the URL to which the charge point will send its messages. The URL has to be the URL of the Central System server.

4.2.2 Central System Configuration

From de Central System side, the following information is needed to configure a new charge point on the server:

- URL of the charge point
- ChargeBoxIdentity
- ConnectorId

URL of the charge point

The URL of the charge point is the address to which the Central System will send its messages to the charge point. If it is a 3G connection, it usually consists in an IP address and a port number. The charge point is the first to send a message and its reply address is written in the header of this message so the URL is not necessary all the time.

When there are several charge points and only one router, the URL is still needed. Indeed, the Central System replies to the router which will rout the messages to the right charge point. In this case, the charge point address is a local one and the Central System cannot directly send a message to it.

ChargeBoxIdentity

The ChargeBoxIdentity is the OCPP identifier of the charge point. This parameter is decided by the constructor when he configures the charge point.

OCPP 1.5 specification was not very clear but there should be one ChargeBoxIdentity for each EVSE. EVSE is a term introduced in OCPP 2.0 in order to clarify it. EVSE is the logical unit in a charge point that supplies electric energy via a Connector for recharging. An EVSE can have one or multiple Connector(s).

Example:

- A charge point with two outlets that cannot be used at the same time corresponds to one EVSE with two Connectors.
- A charge point that can recharge two electric vehicles at the same time has two EVSEs. The two EVSEs should have a different ChargeBoxIdentity. Otherwise, it will imply communication issues between the charge point and the Central System.

ConnectorId

An EVSE can have one or multiple Connector(s). If the EVSE has for instance three Connectors, they will be identified with the connectorId ‘1’, ‘2’ and ‘3’. In addition, the connectorId ‘0’ refers to the EVSE as a whole. For instance, it is used to get the status of the EVSE, or to send a RemoteStartTransaction to the EVSE and not to a particular Connector.

With all these parameters set, the communication between the Central System and a Charge Point can be established.

5 Compatibility Tests between the Central System and a Charge Point

5.1 Carrying out of the tests

Given the constructors did not send a charge point for testing it, the tests were conducted by telephone. Someone of the company was told which action to do on the charge point. Meanwhile, the changes on the web platform were checked. As for the remote actions, they were sent from the Central System and the interlocutor checked what happened on the charge point and did the actions needed.

5.2 Company A

5.2.1 Configuration settings

Company A gave the following information for their charge point:

- URL and Port of the Charger: sent in the header of the OCPP message
- ChargeBoxIdentity: CB100
- ConnectorId: 1, 2
- Swipe cards: 07000014B9F38E and 07000014B9F38F

These swipe cards were added to the data base so they will be accepted during the test.

With this information it was possible to create a new charge point on the web-based management platform. Company A configured its charge point with the right URL and it was possible to start the test.

5.2.2 Tests Phase 1

5.2.2.1 Carrying out of the test

- A phone call has been set up with Company A to do the test. For this first phase, all the tests were done first and the logs were analyzed later.

- First Company A turned on the charge point. After a while, one could see on the web platform that the connection had been made between the charge point and the Central System.

- Company A used the swipe card 07000014B9F38F to start a regular charging session. The electric vehicle started charging which could be seen on the web platform as well.

- It could be seen on the web platform that the energy data from the charge point were well received.

- After a while, Company A tried to stop the charging session with the same swipe card. The charging session stopped but nothing changed on the web platform. The charge point was still in the ‘connected’ state.

- Company A tried to start another transaction but the charge point moved to an unstable state and they had to reboot it.
- After the reboot, the charge point came back to its idle state. A RemoteStartTransaction was sent from the web platform but it did not work. A ChangeAvailability was sent as well but it did not work either.
- Company A tried to start a regular charging session again but the charge point went back to its unstable state so the tests were stopped.

5.2.2.2 Preliminary Analysis

At first sight, there seemed to have some important compatibility issues between the Central System and Company A's charge point. Some statuses on the web platform were wrong, the charge point moved strangely to an unstable state and a RemoteStartTransaction did not work.

After a talk with Company A, one figured out why a RemoteStartTransaction could not be sent from the Central System. When Company A gave the settings configuration of its charge point, they said there were two outlets (ConnectorId 1 and ConnectorId 2). However, they did not specify that the two outlets could work at the same time. On the Central System, only one EVSE, configured with two ConnectorId, had been added but two EVSEs should have been added (one for each ConnectorId because they could work at the same time). Company A did not respect the OCPP specification because there is only one ChargeBoxIdentity while there are two EVSEs.

As said before, ConnectorId '0' refers to the EVSE as a whole. According to the start-up, when someone wants to start a transaction, the transaction is started on an EVSE and not on a specific ConnectorId of this EVSE. The user can choose the outlet he wants. Consequently, the start-up decided to send their RemoteStartTransaction on the ConnectorId '0' and this is the reason why it did not worked with Company A

5.2.2.3 Analysis

The logs have been analyzed after the test.
Given the high length of these logs, only those relevant and useful for the analysis can be seen in Appendix A.

<u>BootNotification</u> (Appendix A.1.1)

After Company A turned on its charge point a BootNotification has been sent to the Central System with all the mandatory parameters (**chargePointVendor**, **chargePointModel**, **firmwareVersion**). In its response, the Central System specified the **heartbeatInterval**.

<u>StatusNotification</u> (Appendix A.1.2)

Then the charge point sent a StatusNotification for **connectorId** 1 and **connectorId** 2 with the **status** 'Available'.

<u>HeartBeat</u> (Appendix A.1.3)

A heartbeat is sent every 300 seconds as asked in the bootNotificationResponse (8:50:17 then 8:55:18…)

Regular charging session (Appendix A.1.4)

When Company A started a charging session, first the charge point asked the Central System the authorization. Given the swipe card was in the data base, the Central System sent ‘Accepted’ in the authorizeResponse.

The charge point sent a StatusNotificationRequest ‘Occupied’ for **connectorId** ‘1’.

The charge point sent a startTransactionRequest containing all the required parameters: the **connectorId**, the **idTag**, the **timestamp** and the **meterStart**. The Central System accepted and gave the charge point the **transactionId**.

MeterValues (Appendix A.1.5)

The charge point sent a meterValuesRequest with the **connectorId**, the **transactionId**, the **timestamp** and the **value**. The charge point only sent the **measurand** **‘**Energy.Active.Import.Register’ in ‘Wh’.

StopTransaction (Appendix A.1.6)

When Company A tried to stop the transaction, the charge point sent a stopTransactionRequest with correct values for the **transactionId**, the **idTag**, the **timestamp** and the **meterStop** but it did not send a StatusNotification then so the status did not change on the web platform.

RemoteStartTransaction (Appendix A.1.7)

The Central System sent a remoteStartTransactionRequest on **connectorId** ‘0’. As said before, it did not work because the charge point expected something on connectorId ‘1’ or connectorId ‘2’ given that they defined two distinct EVSEs.

5.2.2.4 *Corrective actions*

The analysis was shared with Company A. Company A agreed it was their fault if the charge point did not send a StatusNotification after the StopTransaction and that the charge point moved to an unstable state. They worked on it to find out the reason of these problems.

On the other hand, the configuration of Company A’s charge point on the server has to be modified in order to take into account that there were two EVSEs. Two charge points have to appear on the web platform:

- The first one corresponds to the connectorId ‘1’ of ChargeBoxIdentity CB100 (named charge point CB100-1).
- The second one corresponds to the connectorId ‘2’ of ChargeBoxIdentity CB100 (named charge point CB100-2).

Usually when the server receives a message, the ChargeBoxIdentity is extracted from the header of the message. Then the server checks in the data base if a charge point corresponds to this ChargeBoxIdentity. If one corresponds, the server processes the message for this charge point. Otherwise, the message is not processed.

For Company A case, after the modification, two charge points have the same ChargeBoxIdentity. When the data base is checked, two charge points match. In this case, the connectorId is checked as well. Depending if the message is related to connectorId 1 or 2, it will be affected to CB100-1 or CB100-2.

With this modification, the problem should be solved.

5.2.3 Tests Phase 2

5.2.3.1 *Tests and Analysis*

Contrary to the first phase, the logs were checked and analyzed while the tests were progressing.

- Regular charging session

First, one tested a regular charging session. Company A started and then stopped a charging session. The statuses changed as expected on the web platform. All the required OCPP messages were sent in the right order: Authorize – StatusNotification – StartTransaction – MeterValues – StopTransaction – StatusNotification. Company A's modification solved one of the problems because a StatusNotification was well sent after the StopTransaction and the charge point came back to its idle state as it can be seen in Appendix A.2.1.

- RemoteStartTransaction - RemoteStopTransaction

Previously, it was not possible to send a RemoteStartTransaction or a RemoteStopTransaction because the requests were sent to connectorId '0'. With the modification, the requests are now sent specifically to connectorId '1' or '2' depending which outlet you want to use. One tried to send a RemoteStartTransaction to connectorId '1' for a specific swipe card. Company A was able to start a transaction and the status on the web platform moved to 'Connected' and then 'Charging'. This function seemed to work well. The three following tests were tried:

  - A charging session was started with a RemoteStartTransaction and then stopped with a RemoteStopTransaction (Appendix A.2.2)
  - A charging session was started with the swipe card and then stopped with a RemoteStopTransaction.
  - A charging session was started with a RemoteStartTransaction and then stopped with the swipe card.

The first two tests worked well. It can be seen in Appendix A.2.2 that all the required messages and parameters were sent: RemoteStartTransaction – StatusNotification – StartTransaction – MeterValues – RemoteStopTransaction – StopTransaction.

For the third one, it was not possible to stop the charging session with the swipe card. The swipe card was rejected whereas no problem could be seen in the logs. The id-tag specified in the StartTransaction was correct and corresponded to the swipe card used. Consequently, the problem did not come from the Central System but from Company A's charge point.

The explanation may be the following one. For a regular charging session, the id-tag of the swipe card is cached by the charge point. When the user stops the transaction with the same swipe card, the charge point will not send another Authorize to the Central System but will stop the session directly. In this case, if the swipe card is rejected, it may be because the id-tag is not cached by the charge point.

- <u>ChangeAvailability</u>

A ChangeAvailability 'Inoperative' was sent from the Central System. The charge point moved to the state 'Unavailable' which could be seen on the web platform as well so this function worked well. Then a ChangeAvailability 'Inoperative' was sent during a charging session (Appendix A.2.3). The charge point replied with the status 'Scheduled'. The charging session in progress kept going. After a while, Company A stopped the session with the swipe card. After that, the charge point status was 'Unavailable'. The charge point waited until the end of the charging session before changing its status to 'Unavailable'

- <u>Reset</u>

The function Reset was tested. A Reset Hard was sent from the web platform and worked well. The charge point rebooted and then sent the BootNotification and the StatusNotification for its 2 connectors.

- <u>ClearCache</u>

The function ClearCache was tested as follows. A charging session was started with the swipe card 07000014B9F38F so the charge point sent an Authorize. Another session was started with the same swipe card after the end of first one and the charge point did not send an Authorize because the id-tag was in its cache. A ClearCache was then sent from the Central System. The next time Company A started a charging session, the charge point sent an Authorize which means that the ClearCache worked well.

- <u>UnlockConnector</u>

The function UnlockConnector is rather complicated to interpret. For the charge points with a lockable outlet, the function UnlockConnector could unlock the outlet. However if a charging session is in progress, unlocking the outlet implies the stop of the session. Consequently for the start-up, the functions UnlockConnector and RemoteStopTransaction are equivalent. An UnlockConnector while no charging session is in progress has no effect. If a charging session is in progress, the UnlockConnector will stop the session and unlock the outlet exactly like a RemoteStopTransaction.

Company A's interpretation is not exactly the same. A RemoteStopTransaction stops the charging session in progress but it does not unlock the outlet. The outlet can be unlocked only by sending an UnlockConnector. Consequently, sending an UnlockConnector while a session is in progress will not have any effect. The UnlockConnector will be taken into account only if the transaction is finished and the outlet still locked.

This function was tested and worked well. After a RemoteStopTransaction, an UnlockConnector has to be sent for enable the user to unplug the electrical outlet.

- <u>Communication failure</u>

A communication failure was tested. Company A started a charging session and then stopped the communication by switching off the 3G. After a while, they ended the session in process, then started and stopped another session and finally switched on the 3G again. After the communication was restored, all the OCPP messages were well sent to the Central System in chronological order (Appendix A.2.4). A particularity is that only one message is sent with all the MeterValues data instead of sending all the MeterValues separately but it is not a problem for the Central System to process the message. Consequently, the charge point worked perfectly well in an offline situation.

- GetConfiguration - ChangeConfiguration

Some OCPP functions such as GetConfiguration or ChangeConfiguration cannot be tested from the web platform because the Central System does not withstand these functions yet. These two functions were tested with Postman. Postman is an application that can be downloaded from “Chrome Web Store”. This application makes it possible to send XML request. Consequently, with the right configuration, it is possible to send a SOAP message directly to the charge point (or the router). Then, the charge point replies to Postman. It was therefore possible to test these two functions

With GetConfiguration, one tried to get the MeterValueSampleInterval which corresponds to the interval between two MeterValues (Appendix A.2.5). By sending a GetConfiguration empty, the charge point replies with all its configuration keys. These two tests worked well. The function ChangeConfiguration was tested and worked as well. It was possible to change the heartbeatInterval and the MeterValueSampleInterval.

5.2.3.2 Conclusion

The following functions were not tested because Company A charge point cannot yet process them: DataTransfer, DiagnosticsStatusNotification, FirmwareStatusNotification, ReserveNow, CancelReservation, GetDiagnostics, GetLocalListVersion, SendLocalList, UpdateFirmware.

This second phase of tests was far more positive. Except that a charging session could not be stopped with a swipe card after a RemoteStartTransaction, all the other tests worked perfectly well. The messages from the charge point contain all the required information and can be processed by the Central System. Charging session can be started with a swipe card or remotely and the energy data are well sent. The charge point sends a StatusNotification each time its status changes so the availability that can be seen on the web platform reflects reality. In an offline situation, the messages are queued and sent later so no information is lost.

The main problem was Company A’s definition of an EVSE as regards of the ChargeBoxIdentity and the connectorId. This problem was overcome and the Central System can now adapted to Company A’s charge points. Company A will try to solve the remaining problem but it does not prevent the Central System from supervising Company A’s charge points.

5.3 Company B

5.3.1 Configuration settings

Company B gave the following information for their charge point:

- URL and Port of the Charger: http://81.56.166.41:8080/ocpp/
- ChargeBoxIdentity: QC-02350F
- ConnectorId:
  - 0 for the charge point as a whole
  - 1 for the DC connector CHAdeMO
  - 2 for the AC connector 43kW
  - 3 for the DC connector COMBO
- Swipe cards: A60AD65D

With this information it was possible to create a new charge point on the web-based management platform. Company B configured its charge point with the correct URL and it was possible to start the tests.

5.3.2 Tests Phase 1

5.3.2.1 Tests and Analysis

The tests were conducted by telephone again with an interlocutor from Company B. The interlocutor was asked to do some actions on the charge point and the logs were analyzed at the same time.

- Startup of the charge point (Appendix B.1.1)

After Company B turned on their charge point a BootNotification was sent to the Central System with all the mandatory parameters (**chargePointVendor**, **chargePointModel**, **firmwareVersion**). In its response, the Central System specified the **heartbeatInterval**.

After the BootNotification, the charge point sent several StatusNotification. It can be seen in Appendix B.1.1 that the StatusNotification sent are a bit strange. The succession of the messages sent is the following one:

- *StatusNotification connectorId 2 → Available*
- *StatusNotification connectorId 3 → Available*
- *StatusNotification connectorId 0 → Available*
- *StatusNotification connectorId 1 → Faulted*
- *StatusNotification connectorId 3 → Unavailable*
- *StatusNotification connectorId 1 → Available*
- *StatusNotification connectorId 3 → Available*

At first sight, this succession of messages is unexpected but Company B explained the reason. A StatusNotification is first sent for connectorId 2 because the AC connector 43 kW is the one which starts the quickest. The StatusNotification are then sent for connectorId 3, 0 and 1 almost at the same time. Contrary to the DC connector CHAdeMO (connectorId 1), the AC connector 43 kW (connectorId 2) and the DC connector COMBO (connectorId 3) do not need the startup of the automaton to be in their idle state. For the DC connector CHAdeMO, the outlet starts before the automaton so its first status is ‘Faulted’. It induces the status ‘Unavailable’ on connectorId 3 because the two DC outlets are linked and so do their statuses. After a short time, the automaton starts, the status on connectorId 1 moves to ‘Available’ and consequently so does the status on connectorId 3. At the end, all the connectorId are ‘Available’.

The heartbeat messages are then sent at a regular time interval as defined in the bootNotificationResponse (300 seconds).

- Regular charging session (Appendix B.1.2)

One tested then a regular charging session. First, Company B tried to start a charging session with the swipe card 5EADFA47, not registered in the data base. The charge point requested the authorization to the Central System which denied it because the swipe card number was ‘Invalid’**.** Then Company B used the swipe card A60AD65D, registered in the data base and the Central System replied with an authorizeResponse ‘Accepted’.

The charge point sent a startTransactionRequest containing all the required parameters: the **connectorId**, the **idTag**, the **timestamp** and the **meterStart**. The Central System answered ‘Accepted’, specifying the **transactionId**. Like for Company A’s charge point, the **meterStart** is set to ‘0’ for each new charging session.

The charge point sent the following StatusNotification for the different connectorId:

- *StatusNotification connectorId 1: Occupied*
- *StatusNotification connectorId 2: Unavailable*
- *StatusNotification connectorId 3: Unavailable*

Instead of sending only one StatusNotification for the whole charge point on connectorId 0, Company B sends one StatusNotification for each connectorId which causes a misinterpretation by the Central System. Indeed, Company B charge point is defined by the ChargeBoxIdentity QC-02350F with 3 connectorId. When the status ‘Occupied’ is sent on connectorId 1, the charge point moves briefly to the status ‘Occupied’ on the web platform. However, directly after, the Central System receives a status ‘Unavailable’ for connectorId 2 and then for connectorId 3 so the charge point moves to the status ‘Unavailable’. The StatusNotification are sent in the same order all the time and it is the last status received which is taken into account. Consequently, if the charging session occurs on connectorId 1 or 2, at the end, the status of the charge point on the web platform will be ‘Unavailable’ while the real one for the whole charge point is ‘Occupied’. However, if the charging session occurs on connectorId 3, the status is first ‘Unavailable’ but after a few seconds, it moves to its real one, ‘Occupied’.

Consequently, there is a problem of coherence between the statuses on the web platform and the real ones.

During the charging session, the charge point sent meterValuesRequest with the required parameters (the **connectorId**, the **transactionId**, the **timestamp** and the **value**). Like for Company A, the charge point only sent the **measurand ‘**Energy.Active.Import.Register’. The **unit** is not specified in the **value**. However, it corresponds to ‘Wh’ which is the default one on the Central System so there is not any misinterpretation of the energy data.

Company B used the same swipe card and stopped the charging session. The charge point sent a stopTransactionRequest with the expected **transactionId**, **idTag**, **timestamp** and **meterStop**.

The charge point also sent a StatusNotification 'Available' for each connectorId. ConnectorId 1, moved from 'Occupied' to 'Available' which induced a change from 'Unavailable' to 'Available' for connectorId 2 and connectorId 3. At the end, the status on the web platform is 'Available' like the real one.

- Remote Actions

One tried to send remote actions from the web platform but none of them worked. No action could be sent remotely. The reason of this failure is that the Central System sends remote actions on an EVSE which means on the whole charge point (connectorId '0'). Company B's charge point does not take into account the requests on connectorId 0 but expects requests on connectorId 1, 2 or 3.

- Communication failure (Appendix B.1.3)

One simulated a communication failure. Company B stopped the communication during a charging session by unplugging the Ethernet cable. During the offline time, Company B stopped and started several charging sessions. After the communication was restored, all the OCPP messages were well sent to the Central System in chronological order. Contrary to Company A, the MeterValues messages are sent one by one and all the information is not store in only one message.

Consequently, the charge point works perfectly well in an offline situation. All the OCPP messages are queued and sent when the communication is restored.

5.3.2.2 Corrective actions

The connection was made between Company B's charge point and the Central System. And charging sessions could be done. There are still two important issues: the misinterpretation of the charge point status and the fact that no action can be sent from the Central System.

These two problems are linked and due to the fact that Company B's charge point does not use the connectorId '0' but processes each outlet independently. It would be far easier if Company B could modify and adapt its software. The problem is that Company B has a lot of charge points installed all around France and Europe. Making a modification would mean updating all these charge points to make the Central System able to supervise Company B's charge points. This solution is not conceivable right now. Maybe for its next versions, Company B will modify its software but at the present time, the Central System has to adapt to Company B's charge points.

The first problem is the status of the charge point. As said before, the charge point sends successively a StatusNotification for all its connectorId so only the last one is taken into account by the Central System. The solution chosen is to store the statuses of the three connectorId on a table. Depending on the combination of these three statuses and on the former statuses, the Central System deduces a global status for the charge point.

However, the statuses must not be stored for all the charge points managed by the Central System. If it is done for the start-up's charge points as well, it will be a loss of time and a loss of space. It has to be done only for Company B's charge points. The solution chosen to do that is the following: when a charge point boots and sends a BootNotification, the Central System checks the **chargePointVendor**. If the **chargePointVendor** is 'Company B', the Central System will apply the modification and cache the statuses. Otherwise, the charge point will be processed in a common way.

As regards of the remote action problem, it was decided to allow the sending of remote actions on a specific connectorId. On the web-platform, a pop-up window opens and the user is asked on which connectorId he wants to send the remote action. This solution should work but it means that the user knows in advance which outlet he wants to use.

5.3.3 Tests Phase 2

5.3.3.1 Tests and Analysis

Thanks to the storage of the statuses, the global status of the charge point deduced corresponds now to the real one. This first problem is solved.

- <u>Remote actions</u>

Concerning the remote actions, it is now possible to choose the connectorId on which you want to send the action. Several remote actions were tried but none of them worked. In the logs, it could be seen that the messages were well sent from the Central System but no response was received. Company B sent the logs they got on their charge point and one tried to find out the reason of this failure. The charge point well received the messages and replied. However, the answers were not in the expected format. The answers were in the Gzip compressed format which could not be processed by the Central System.

When a message is sent from the charge point, the default format is SOAP because the charge point does not know which format the Central System will accept. In this case there is not any problem. The Central System receives a SOAP message and answers with a SOAP message as well.

A message sent from the Central System is in the SOAP format. However, in the request, it is not specified that the answer is expected in a specific format. Consequently, the charge point answers with its default format which is a compressed one, Gzip. The answer is thus filtered and is not processed by the Central System.
Given it was a bit complicated to change the parameters on the Central System side, Company B changed a parameter on its charge point. The charge point is now forced to answer in SOAP format. With this new configuration, remote actions could be tested again.

- <u>RemoteStartTransaction – RemoteStopTransaction</u>

Like how we did for Company A, the three following scenarios were tested in order to check the well-functioning of the RemoteStartTransaction and the RemoteStopTransaction.

- A charging session was started with a RemoteStartTransaction and then stopped with a RemoteStopTransaction (Appendix B.2.1)
- A charging session was started with the swipe card and then stopped with a RemoteStopTransaction.

- A charging session was started with a RemoteStartTransaction and then stopped with the swipe card.

These three tests worked perfectly well. As it can be seen in Appendix B.2.1 for the first test, all the required messages were sent and contained the required parameters. Charging sessions could be started and stopped remotely without any problem.

- ChangeAvailability (Appendix B.2.2)

The function ChangeAvailability was tested and worked as well. It was possible to put the charge point in an 'Unavailable' state and then put it back in its idle state. Like for Company A's charge point, if a ChangeAvailability 'Inoperative' is sent while a charging session is in progress, the action is 'Scheduled' and the charge point will move to the 'Unavailable' state after the end of the transaction.

- Reset Hard (Appendix B.2.3)

The function Reset was tested. A Reset Hard was sent from the web platform and worked well. The charge point rebooted and then sent the BootNotification and the StatusNotification as expected.

- UnlockConnector (Appendix B.2.4)

An UnlockConnector was sent during a charging session and it was immediately followed by a StopTransaction. Consequently, Company B decided to make the function UnlockConnector work like the function RemoteStopTransaction.

5.3.3.2 Conclusion

Thanks to the modifications, the problem of the StatusNotification is solved. The Central System stores the statuses of the different connectors in order to deduce the global status of the charge point.

It is now also possible to send remote actions from the Central System. By default, the charge point replies to a message from the Central System with a compressed message but with a modification on the charge point, the problem is solved and the Central System can send remote actions to the charge point.

At the end, this charge point is totally manageable. However, contrary to the Company A's ones, every Company B's charge points do not work exactly in the same way. The Central System has to adapt to the different type of charge points.

5.3.4 Adjustments for other charge points of Company B

5.3.4.1 Compatibility issues with the different models of charge point

As said previously, every Company B's charge points do not work exactly in the same way. Company B has two main types of charge point:

- The Quick Chargers

- The Normal Chargers

Depending on their version, the Quick Chargers have one or more of these three connectors: CHAdeMO-44kW, AC-43kW, and COMBO 44kW. The Quick Chargers have necessarily only one ChargeBoxIdentity but depending on how there are configured, several connectors can work at the same time. The Quick Charger tested had the three connectors but they could not work simultaneously.

The Normal Chargers have household plug or plug type 2. The plugs type 2 can operate between 3 and 22kW depending on how the charge point is configured. Most of the time, the Normal Chargers have either one household plug and one plug type 2 or two plugs type 2. In the default configuration, the charge point has only one ChargeBoxIdentity and the two connectors can work at the same time which is the problem that occurred with Company A's charge points.

Consequently, the modification done will not suit for all Company B's charge points. Depending on the type of the charge point, the Central System has to process the information in a different way. The idea that arouse from that is that the Central System must know the configuration of the charge point in order to process the messages in a correct way.

5.3.4.2 Possible solutions

The first option could be to manually specify the configuration of the charge point when it is created in the Central System. This solution could work but if the number of charge points increases it would not be very efficient. Besides, if the information given for the charge point is not correct or is modified, it would lead to dysfunction.

Company B's BootNotification contains some optional parameters such as the 'chargePointModel' which may be used. More details were asked to Company B about it in order to find out if it could be usable. For instance, if the 'chargePointModel' is not the same for two identical charge points or if there is a high number of 'chargePointModel', it could not be used to distinguish the different cases.

Company B gave the following precisions as regard of the 'chargePointModel':

The 'chargePointModel' is the same for identically configured charge points.

- *For the Quick Chargers, the following values are used:*
  - *"NQC": 1 connector CHAdeMO-44kW*
  - *"NQC-ACDC": 2 connectors: CHAdeMO-44kW and AC-43kW*
  - *"NQC-COMBO": 2 connectors: CHAdeMO-44kW and COMBO-44kW*
  - *"NQC-ACDC-COMBO": 3 connectors: CHAdeMO-44kW, COMBO-44kW and AC-43kW*

By default, if the charge point has one connector AC and one connector DC, they cannot work at the same time. The simultaneity constraint can be configured with the OCPP message ChangeConfiguration.

- *For the Normal Chargers, the following values are used:*
  - *"GNS100": charge points with a screen and with a mother broad version 1*
  - *"GNS102": charge points without screen and with a mother broad version 1*
  - *"GNS200": charge points with a screen and with a mother broad version 2*
  - *"GNS201": charge points without screen and with a mother broad version 2*

By default, if the charge point has two connectors or more, they can work simultaneously. A ChangeConfiguration can change it as well.

5.3.4.3 *Protocol chosen to process the different cases*

When the Central System receives a BootNotification, the following parameters are extracted: chargeBoxIdentity, chargePointVendor and chargePointModel.

<u>Start-up's charge point</u>

If the chargePointVendor is the one of the start-up, the messages will be process in the normal way.

<u>Company A's charge point</u>

If the chargePointVendor is 'Company A', the message will be process as follows. First the Central System search for a charge point whose chargeBoxIdentity is the same as the one extracted from the BootNotification. If there is only one match, the messages will be process in the normal way (it means that the two connectors cannot work at the same time). If there are two matches, the messages will be then routed depending on the connectorId. If the message does not contain any connectorId, it will be sent to the two charge points (named chargeBoxIdentity-1 and chargeBoxIdentity-2). Otherwise, the messages will be routed either to chargeBoxIdentity-1 or chargeBoxIdentity-2.

<u>Company B's charge point</u>

If the chargePointVendor is 'Company B', the process is a bit more complex because there are more cases.

<u>Normal Chargers process</u>

Processing the Normal Chargers is not complicated because it is the same as the process of Company A's charge points.

<u>Quick Chargers process</u>

- 'chargePointModel = NQC': the messages are processed is the usual way because it corresponds to a basic case (one chargeBoxIdentity and one connectorId)
- 'chargePointModel = NQC-ACDC': in this case there are one AC connector and one DC connector and these two connectors can sometimes work at the same time.
  - If the two connectors cannot work at the same time, only one charge point will match with the chargeBoxIdentity extracted from the BootNotification message. In this case, the messages will be process as explained in the previous part. The statuses of the two connectors will be cached to deduce the global status of the charge point
  - If the two connectors can work at the same time, two charge points will match with the chargeBoxIdentity extracted from the BootNotification message. The messages will be routed either to chargeBoxIdentity-1 or to chargeBoxIdentity-2 like how it is done for Company A so there is no need to cache the statuses.
- 'chargePointModel = NQC-COMBO': in this case there are two DC connectors so they can never work simultaneously. Consequently, the messages will be processed as

explained in the previous part: the global status of the charge point will be deduced from the statuses cached.

- 'chargePointModel = NQC-ACDC-COMBO': it is the more complex case. First the Central System will check how many charge points match with the chargeBoxIdentity extracted from the message.
  - If only one charge point matches, it means that the connectors cannot work at the same time. Consequently, it corresponds to the case seen during the tests and the messages will be process as explained, with the statuses cached.
  - If three charge point matches, the messages will be processed in the same way as Company A. There will be three charge points which correspond to chargeBoxIdentity-1, chargeBoxIdentity-2 and chargeBoxIdentity-3.

With this protocol, the different cases can be processed and Company B and Company A' charge points can be supervise.

5.4 Company C

5.4.1 Configuration settings

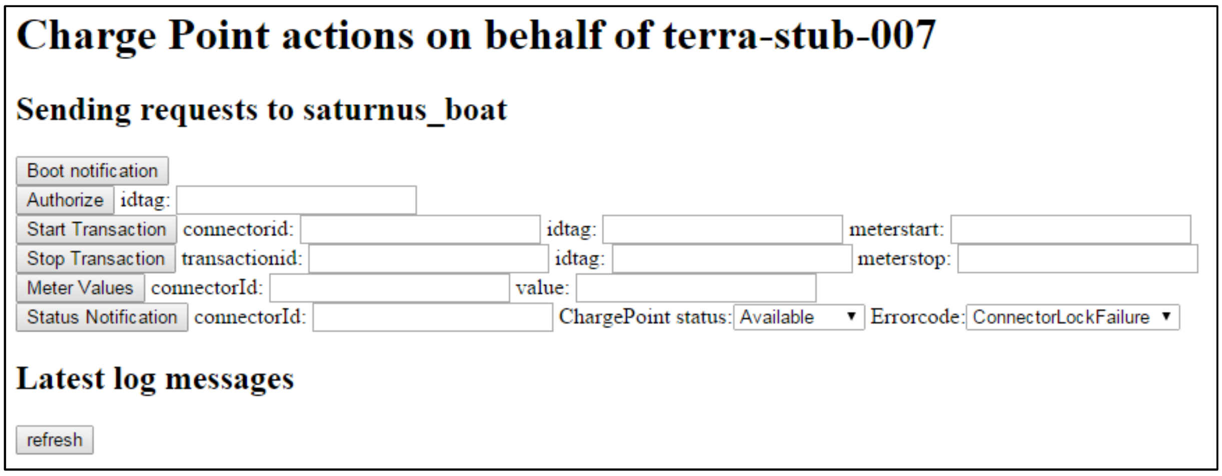
In this case, the configuration is a bit different. The communication is not done directly between Company C's charge point and the Central System. Company C has its own platform which gathers the information from all its charge points. Consequently, the communication has to be made between Bouygues's server and Company C's server.

Contrary to the other constructors, Company C has a standardized procedure with several stages to conduct tests with a supervision supplier.

5.4.2 Tests stage 1

Before testing the compatibility with a real charge point, Company C wanted to do some preliminaries tests in order to ensure that the Central System will be able process OCPP messages and reply to it.

Company C developed a web-browser from which it is possible to send the basic OCPP messages. A screenshot of the web page can be seen in the Figure 5.1 below.


Charge Point actions on behalf of terra-stub-007

Sending requests to saturnus_boat

Boot notification
Authorize idtag:
Start Transaction connectorid: idtag: meterstart:
Stop Transaction transactionid: idtag: meterstop:
Meter Values connectorId: value:
Status Notification connectorId: ChargePoint status: Available Errorcode: ConnectorLockFailure

Latest log messages

refresh

Figure 5.1: Screenshot of Company C browser

Hence this browser makes it possible to send the basic OCPP messages by specifying the different parameters.

The information entered in the Central System was the following:

- URL: http://www401.stage.com:8080/saturnus/chargePointService15/chargePointService15
- ChargeBoxIdentity: terra_stub_007
- ConnectorId: 1

The URL corresponds to Company C server.

It was test first with this configuration. It did not work because of the ChargeBoxIdentity. The Central System cannot process messages when the name of the charge point has an underscore. Company C was asked to change it and the new ChargeBoxIdentity was then 'terra-stub-007'.

With this new ChargeBoxIdentity, the communication could be made. The different functions were therefore tested to check how the Central System reacts to the messages from the web-browser. Everything worked well so it was possible to move to the second stage and start the tests on a real charge point.

5.4.3 Tests stage 2

For this second stage, the configuration on the Central System is the following:

- URL: http://www401.stage.com:8080/saturnus/chargePointService15/chargePointService15
- ChargeBoxIdentity: t53-ocpp-007
- ConnectorId:
    - 1 for the AC connector 43kW
    - 2 for the DC connector COMBO

The tests were almost the same as those done for Company A and Company B. Company C still has some tiny difference with Company A and Company B. For instance, Company C's charge point does not withstand a Reset but do withstand the reservation. However, the overall functioning is the same.

Only one problem was noticed during the tests. Company C had a problem with the StopTransaction. They sent all the time the StopTransaction message twice (Appendix C.1). After the charge point sent a stopTransactionRequest, the Central System replied and accepted the request. A StatusNotification was then sent and the status moved to 'Available'. However, directly after, the charge point sent another stopTransactionRequest, exactly the same as the one sent before. Given there was not anymore a transaction in progress, the Central System replied with an error message. The consequence was that the 'idTag' used for the transaction was removed from the local white list of the charge point.

Hence, after each transaction, the 'idTag' is removed from the local white list and no 'idTag' is saved which will be a problem in case of a communication failure. In case of a communication failure, the charge point uses the 'idTag' cached previously to authorize or a transaction. Because of this problem, no 'idTag' will be cached and no user will be able to start a transaction in an offline situation.

One figured out where the problem came from. In the StopTransaction message Company C sends, there is the optional parameter 'idTag'. Given this parameter is optional, the Central System ignores it. The stopTransactionResponse sent by the Central System does not contain any parameter. However, Company C expects an 'idTagInfo' in the stopTransactionResponse given an 'idTag' is sent. The Central System does not specify the 'idTagInfo' so Company C's server considers that the reply is erroneous. The server tries to send another StopTransaction and the 'idTag' is considered as 'Invalid' given nothing is answered.

One agreed to make a change in our Central System in order to send back the parameter 'idTagInfo' in the stopTransactionResponse if the parameter 'idTag' was present in the stopTransactionRequest. After this modification, a charging session was tried and it worked well (Appendix C.2). Company C's server did not send any error message because the Central System replied with the parameters it expected.

After this modification, Company C's could be supervised by the Central System without any problem.

6 Conclusion

The aim of the thesis was to make charge points from different constructors compatible with a single Central System. Tests have been done with three charge point manufacturers. Nowadays, the OCA protocol OCPP is very widely used in order to make the communication easier but the specification of OCPP 1.5 is not crystal clear for some particular points. Besides, some constructors cannot implement perfectly what the specification tells because of technological issues. Even if a single protocol was used, it did not imply compatibility for sure.

Some problems arose during the tests with every constructor. For Company A, the main problem was the use of the couple ChargeBoxIdentity/connectorId to define an EVSE. For Company B the main problem was that they send a StatusNotification 'Unavailable' for the connectorId which are not used instead of a global status of the charge point. Some adjustments had to be made on the Central System in order to adapt to these different configurations. At the end, the charge points tested were manageable by the Central System so the overall result of the thesis is very positive.

On the other hand, the compatibility is not certain for all the charge points of a company. Only one charge point from each company was tested and even if the company tells that the software is the same for all their products, extensive tests have to be done. Besides, the compatibility is not persistent either because the tests were done for a specific software version. If the constructor releases a new version, new problems could appear.

Consequently, the overall compatibility with a constructor has been checked thanks to the tests done. However, if Bouygues wins a bid for the supervision of charge points, extensive tests will have to be done especially for the charge points which will be installed.

Bibliography

[1] IEC. *International Standard 62196-1:2014*, Plugs, socket-outlets, vehicle connectors and vehicle inlets – Conductive charging of electric vehicles – Part 1: General requirements

[2] IEC. *International Standard 62196-2:2011*, Plugs, socket-outlets, vehicle connectors and vehicle inlets – Conductive charging of electric vehicles – Part 2: Dimensional compatibility and interchangeability requirements for a.c. pin and contact-tube accessories

[3] IEC. *International Standard 62196-3:2014*, Plugs, socket-outlets, vehicle connectors and vehicle inlets – Conductive charging of electric vehicles – Part 3: Dimensional compatibility and interchangeability requirements for d.c. and a.c./d.c. pin and contact-tube vehicle couplers

[4] IEC. *International Standard 61851-1:2010*, Electric vehicle conductive charging system – Part 1: General requirements

[5] IEC. *International Standard 61851-22:2001*, Electric vehicle conductive charging system – Part 22: AC electric vehicle charging station

[6] IEC. *International Standard 61851-23:2014*, Electric vehicle conductive charging system – Part 21: DC electric vehicle charging station

[7] OCA website, [Online]. Available: http://www.openchargealliance.org/ [Accessed December 2014]

[8] OCA. *Open Charge Point Protocol 1.5*, 010.030.007, 2012-06-08

[9] OCA. *Open Charge Point Protocol 2.0*, Release Candidate 2, 2014-11-03

[10] OCA. *OCPP v1.5 - A functional description*

[11] OCA. *OCPP 1.5 Central System Service WSDL*

[12] OCA. *OCPP 1.5 Charge Point Service WSDL*

Appendix A – Tests with Company A

A.1 Tests with Company A - Phase 1

A.1.1 BootNotification

```xml
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:00000061-1787-49f8-ab8b-4567327b23c6</wsa5:MessageID>
  <wsa5:From>
     <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
     <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/BootNotification</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:bootNotificationRequest>
     <cs:chargePointVendor>Company A </cs:chargePointVendor>
     <cs:chargePointModel>MONOBLOCK</cs:chargePointModel>
     <cs:chargePointSerialNumber>3N143750423H2S1B7551700009</cs:chargePointSerialNumber>
     <cs:chargeBoxSerialNumber>EV.1S22P22R3N1442208001001501322B</cs:chargeBoxSerialNumber>
     <cs:firmwareVersion>2.7.1-11</cs:firmwareVersion>
     <cs:iccid>89330153264443853850</cs:iccid>
     <cs:imsi>208018300258427</cs:imsi>
  </cs:bootNotificationRequest>
</SOAP-ENV:Body>
```

```xml
<v:Header>
  <a:Action>/BootNotificationResponse</a:Action>
  <a:MessageID>urn:uuid:ab904f7b-1193-4b0c-a95f-c733f4c209bf</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:00000061-1787-49f8-ab8b-4567327b23c6</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:bootNotificationResponse>
     <cs:currentTime>2014-12-11T08:45:15.764Z</cs:currentTime>
     <cs:heartbeatInterval>300</cs:heartbeatInterval>
     <cs:status>Accepted</cs:status>
  </cs:bootNotificationResponse>
  </v:Body>
```

A.1.2 StatusNotification

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:5489599b-58f9-4555-a43c-986966334873</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/StatusNotification</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Available</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-11T08:45:15Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Header>
  <a:Action>/StatusNotificationResponse</a:Action>
  <a:MessageID>urn:uuid:5c08f24f-c453-487f-9e94-22edb2faebf7</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:5489599b-58f9-4555-a43c-986966334873</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:5489599b-1787-49f8-ab8b-4567327b23c6</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/StatusNotification</wsa5:Action>
  </SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>2</cs:connectorId>
    <cs:status>Available</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-11T08:45:15Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Header>
  <a:Action>/StatusNotificationResponse</a:Action>
  <a:MessageID>urn:uuid:9b2126de-fffe-48e8-baea-a34bc495f49d</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:5489599b-1787-49f8-ab8b-4567327b23c6</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

A.1.3 HeartBeat

```xml
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:54895ac7-2950-43d7-b4b0-dc5119495cff</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/Heartbeat</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:heartbeatRequest />
</SOAP-ENV:Body>
```

```xml
<v:Header>
  <a:Action>/HeartbeatResponse</a:Action>
  <a:MessageID>urn:uuid:aefcdb01-1e58-46e7-b7f4-2e652f35a409</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:54895ac7-2950-43d7-b4b0-dc5119495cff</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:heartbeatResponse>
    <cs:currentTime>2014-12-11T08:50:18.081Z</cs:currentTime>
  </cs:heartbeatResponse>
</v:Body>
```

```xml
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:54895bf4-72f1-4ca7-aae8-944a625558ec</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/Heartbeat</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:heartbeatRequest />
</SOAP-ENV:Body>
```

```xml
<v:Header>
  <a:Action>/HeartbeatResponse</a:Action>
  <a:MessageID>urn:uuid:5bf41f07-7290-474e-a230-5cf64c328e76</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:54895bf4-72f1-4ca7-aae8-944a625558ec</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:heartbeatResponse>
    <cs:currentTime>2014-12-11T08:55:17.990Z</cs:currentTime>
  </cs:heartbeatResponse>
</v:Body>
```

A.1.4 Regular Charging Session

Authorize

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:548963e2-6881-4cf7-8216-231b1f16e9e8</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/Authorize</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:authorizeRequest>
    <cs:idTag>07000014B9F38F</cs:idTag>
  </cs:authorizeRequest>
</SOAP-ENV:Body>
```

```
<v:Header>
  <a:Action>/AuthorizeResponse</a:Action>
  <a:MessageID>urn:uuid:34768779-0c8b-4b51-9cf7-d8323051858a</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:548963e2-6881-4cf7-8216-231b1f16e9e8</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:authorizeResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
  </cs:authorizeResponse>
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:548963ec-025d-45d8-9190-cde766ef438d</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/StatusNotification</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Occupied</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-11T09:29:16Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Header>
  <a:Action>/StatusNotificationResponse</a:Action>
  <a:MessageID>urn:uuid:cb2402d4-7125-43ae-947f-51d06a8753a5</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:548963ec-025d-45d8-9190-cde766ef438d</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StartTransaction

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
```

```xml
  <wsa5:MessageID>urn:uuid:548963f0-8ffd-4acc-940e-0f763352255a</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/StartTransaction</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:startTransactionRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:idTag>07000014B9F38F</cs:idTag>
    <cs:timestamp>2014-12-11T09:29:18Z</cs:timestamp>
    <cs:meterStart>0</cs:meterStart>
  </cs:startTransactionRequest>
</SOAP-ENV:Body>
```

```xml
<v:Header>
  <a:Action>/StartTransactionResponse</a:Action>
  <a:MessageID>urn:uuid:d9b0e1a3-9f99-4b95-ab15-e510cb5cab04</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:548963f0-8ffd-4acc-940e-0f763352255a</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24324</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

A.1.5 MeterValues

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:548963fd-8745-4562-909c-f92e0ded7263</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true"> http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/MeterValues</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:meterValuesRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:transactionId>24324</cs:transactionId>
    <cs:values>
      <cs:timestamp>2014-12-11T09:29:33Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">1</cs:value>
    </cs:values>
  </cs:meterValuesRequest>
</SOAP-ENV:Body>
```

```
<v:Header>
  <a:Action>/MeterValuesResponse</a:Action>
  <a:MessageID>urn:uuid:b483b076-58a6-44ec-ab96-767deead494a</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:548963fd-8745-4562-909c-f92e0ded7263</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

A.1.6 StopTransaction

```
<SOAP-ENV:Header>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
  <wsa5:MessageID>urn:uuid:54896460-ceba-4bd0-836c-6125628c895d</wsa5:MessageID>
  <wsa5:From>
    <wsa5:Address>http://90.121.54.198:8082</wsa5:Address>
  </wsa5:From>
  <wsa5:ReplyTo>
    <wsa5:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa5:Address>
  </wsa5:ReplyTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/StopTransaction</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cs:stopTransactionRequest>
    <cs:transactionId>24324</cs:transactionId>
    <cs:idTag>07000014B9F38F</cs:idTag>
    <cs:timestamp>2014-12-11T09:31:12Z</cs:timestamp>
    <cs:meterStop>8</cs:meterStop>
  </cs:stopTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Header>
  <a:Action>/StopTransactionResponse</a:Action>
   <a:MessageID>urn:uuid:fddb5309-3d1d-433a-a8a3-10561a0f66ac</a:MessageID>
  <a:To>http://www.w3.org/2005/08/addressing/anonymous</a:To>
  <a:RelatesTo>urn:uuid:54896460-ceba-4bd0-836c-6125628c895d</a:RelatesTo>
  <cs:chargeBoxIdentity>CB100</cs:chargeBoxIdentity>
</v:Header>
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
</v:Envelope>
```

A.1.7 RemoteStartTransaction

```xml
<v:Header>
  <cp:chargeBoxIdentity>CB100</cp:chargeBoxIdentity>
  <a:Action v:mustUnderstand="true">/RemoteStartTransaction</a:Action>
  <a:MessageID>urn:uuid:bce85b8d-0caa-4f52-b74b-c1ed4065102d</a:MessageID>
  <a:To v:mustUnderstand="true">http://90.117.162.192:8082/</a:To>
  <a:ReplyTo v:mustUnderstand="true">
    <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
  </a:ReplyTo>
</v:Header>
<v:Body>
  <cp:remoteStartTransactionRequest>
    <cp:connectorId>0</cp:connectorId>
    <cp:idTag>07000014B9F38F</cp:idTag>
  </cp:remoteStartTransactionRequest>
</v:Body>
```

```xml
<SOAP-ENV:Header>
  <cp:chargeBoxIdentity>CB100</cp:chargeBoxIdentity>
  <wsa5:RelatesTo>urn:uuid:bce85b8d-0caa-4f52-b74b-c1ed4065102d</wsa5:RelatesTo>
  <wsa5:To SOAP-ENV:mustUnderstand="true">http://www.w3.org/2005/08/addressing/anonymous</wsa5:To>
  <wsa5:Action SOAP-ENV:mustUnderstand="true">/RemoteStartTransactionResponse</wsa5:Action>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cp:remoteStartTransactionResponse>
    <cp:status>Rejected</cp:status>
  </cp:remoteStartTransactionResponse>
</SOAP-ENV:Body>
```

A.2 Tests with Company A - Phase 2

From now on, there will be only the “Body” part of the OCPP messages. The “Header” part is not very relevant for the analysis and takes up a lot of space in the Appendix.

A.2.1 Regular charging session

StopTransaction

```
<SOAPENV:Body>
  <cs:stopTransactionRequest>
    <cs:transactionId>24375</cs:transactionId>
    <cs:idTag>07000014B9F38E</cs:idTag>
    <cs:timestamp>2014-12-16T09:16:06Z</cs:timestamp>
    <cs:meterStop>4</cs:meterStop>
  </cs:stopTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:stopTransactionResponse/>
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Available</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-16T09:16:10Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:statusNotificationResponse/>
</v:Body>
```

A.2.2 RemoteStartTransaction - RemoteStopTransaction

RemoteStartTransaction

```
<v:Body>
  <cp:remoteStartTransactionRequest>
    <cp:connectorId>1</cp:connectorId>
    <cp:idTag>07000014B9F38E</cp:idTag>
  </cp:remoteStartTransactionRequest>
</v:Body>

<SOAP-ENV:Body>
  <cp:remoteStartTransactionResponse>
    <cp:status>Accepted</cp:status>
  </cp:remoteStartTransactionResponse>
</SOAP-ENV:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Occupied</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-17T13:26:37Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StartTransaction

```
<SOAP-ENV:Body>
  <cs:startTransactionRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:idTag>07000014B9F38F</cs:idTag>
    <cs:timestamp>2014-12-17T13:26:43Z</cs:timestamp>
    <cs:meterStart>0</cs:meterStart>
  </cs:startTransactionRequest>
</SOAP-ENV:Body>


<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24389</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

MeterValues

```
<SOAP-ENV:Body>
  <cs:meterValuesRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:transactionId>24389</cs:transactionId>
    <cs:values>
      <cs:timestamp>2014-12-17T13:26:55Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">106</cs:value>
    </cs:values>
  </cs:meterValuesRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

RemoteStopTransaction

```
<v:Body>
  <cp:remoteStopTransactionRequest>
    <cp:transactionId>24389</cp:transactionId>
  </cp:remoteStopTransactionRequest>
```

```
</v:Body>

<SOAP-ENV:Body>
  <cp:remoteStopTransactionResponse>
    <cp:status>Accepted</cp:status>
  </cp:remoteStopTransactionResponse>
</SOAP-ENV:Body>
```

StopTransaction

```
<SOAP-ENV:Body>
  <cs:stopTransactionRequest>
    <cs:transactionId>24389</cs:transactionId>
    <cs:timestamp>2014-12-17T13:27:40Z</cs:timestamp>
    <cs:meterStop>109</cs:meterStop>
  </cs:stopTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

A.2.3 ChangeAvailability

ChangeAvailability

```
<v:Body>
  <cp:changeAvailabilityRequest>
    <cp:connectorId>2</cp:connectorId>
    <cp:type>Inoperative</cp:type>
  </cp:changeAvailabilityRequest>
</v:Body>

<SOAP-ENV:Body>
  <cp:changeAvailabilityResponse>
    <cp:status>Scheduled</cp:status>
  </cp:changeAvailabilityResponse>
</SOAP-ENV:Body>
```

StopTransaction

```
<SOAP-ENV:Body>
  <cs:stopTransactionRequest>
    <cs:transactionId>24394</cs:transactionId>
    <cs:timestamp>2014-12-17T13:53:37Z</cs:timestamp>
    <cs:meterStop>9</cs:meterStop>
  </cs:stopTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>2</cs:connectorId>
    <cs:status>Unavailable</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-17T13:53:46Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

A.2.4 Communication failure

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Occupied</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-16T10:22:28Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StartTransaction

```
<SOAP-ENV:Body>
  <cs:startTransactionRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:idTag>07000014B9F38F</cs:idTag>
    <cs:timestamp>2014-12-16T10:22:30Z</cs:timestamp>
    <cs:meterStart>89</cs:meterStart>
  </cs:startTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24385</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

MeterValues

```
<SOAP-ENV:Body>
  <cs:meterValuesRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:transactionId>24385</cs:transactionId>
    <cs:values>
      <cs:timestamp>2014-12-16T10:22:46Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">90</cs:value>
    </cs:values>
  </cs:meterValuesRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

STOP OF THE COMMUNICATION

MeterValues

```
<SOAP-ENV:Body>
  <cs:meterValuesRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:transactionId>24385</cs:transactionId>
    <cs:values>
      <cs:timestamp>2014-12-16T10:23:01Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">91</cs:value>
    </cs:values>
    <cs:values>
      <cs:timestamp>2014-12-16T10:23:16Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">92</cs:value>
    </cs:values>
    <cs:values>
      <cs:timestamp>2014-12-16T10:23:31Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
```

```
                 context="Sample.Periodic">93</cs:value>
    </cs:values>
    <cs:values>
      <cs:timestamp>2014-12-16T10:23:46Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                 context="Sample.Periodic">94</cs:value>
   </cs:values>
    <cs:values>
      <cs:timestamp>2014-12-16T10:24:01Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                 context="Sample.Periodic">95</cs:value>
    </cs:values>
    <cs:values>
      <cs:timestamp>2014-12-16T10:24:16Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                 context="Sample.Periodic">96</cs:value>
    </cs:values>
   <cs:values>
      <cs:timestamp>2014-12-16T10:24:31Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                 context="Sample.Periodic">97</cs:value>
    </cs:values>
  </cs:meterValuesRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

StopTransaction

```
<SOAP-ENV:Body>
  <cs:stopTransactionRequest>
    <cs:transactionId>24385</cs:transactionId>
    <cs:idTag>07000014B9F38F</cs:idTag>
    <cs:timestamp>2014-12-16T10:24:41Z</cs:timestamp>
    <cs:meterStop>98</cs:meterStop>
  </cs:stopTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Available</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-16T10:24:48Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Occupied</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-16T10:25:05Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StartTransaction

```
<SOAP-ENV:Body>
```

```
  <cs:startTransactionRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:idTag>07000014B9F38E</cs:idTag>
    <cs:timestamp>2014-12-16T10:25:07Z</cs:timestamp>
    <cs:meterStart>98</cs:meterStart>
  </cs:startTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24386</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

MeterValues

```
<SOAP-ENV:Body>
  <cs:meterValuesRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:transactionId>24386</cs:transactionId>
    <cs:values>
      <cs:timestamp>2014-12-16T10:25:22Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">99</cs:value>
    </cs:values>
    <cs:values>
      <cs:timestamp>2014-12-16T10:25:37Z</cs:timestamp>
      <cs:value unit="Wh" location="Outlet" measurand="Energy.Active.Import.Register" format="Raw"
                context="Sample.Periodic">100</cs:value>
    </cs:values>
  </cs:meterValuesRequest>
</SOAP-ENV:Body>

  <v:Body>
    <cs:meterValuesResponse />
  </v:Body>
```

StopTransaction

```
<SOAP-ENV:Body>
  <cs:stopTransactionRequest>
    <cs:transactionId>24386</cs:transactionId>
    <cs:idTag>07000014B9F38E</cs:idTag>
    <cs:timestamp>2014-12-16T10:26:04Z</cs:timestamp>
    <cs:meterStop>102</cs:meterStop>
  </cs:stopTransactionRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <cs:statusNotificationRequest>
    <cs:connectorId>1</cs:connectorId>
    <cs:status>Available</cs:status>
    <cs:errorCode>NoError</cs:errorCode>
    <cs:timestamp>2014-12-16T10:26:07Z</cs:timestamp>
  </cs:statusNotificationRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

HeartBeat

```
<SOAP-ENV:Body>
  <cs:heartbeatRequest />
</SOAP-ENV:Body>

<v:Body>
```

```
  <cs:heartbeatResponse>
    <cs:currentTime>2014-12-16T10:31:16.216Z</cs:currentTime>
  </cs:heartbeatResponse>
</v:Body>
```

A.2.5 Get configuration

GetConfiguration ‘MeterValueSampleInterval’

```
<s:Body>
  <getConfigurationRequest>
    <key>MeterValueSampleInterval</key>
  </getConfigurationRequest>
</s:Body>

<SOAP-ENV:Body>
  <cp:getConfigurationResponse>
    <cp:configurationKey>
      <cp:key>MeterValueSampleInterval</cp:key>
      <cp:readonly>false</cp:readonly>
      <cp:value>15</cp:value>
    </cp:configurationKey>
  </cp:getConfigurationResponse>
</SOAP-ENV:Body>
```

GetConfiguration empty

```
<s:Body>
  <getConfigurationRequest>
</s:Body>

<SOAP-ENV:Body>
  <cp:getConfigurationResponse>
    <cp:configurationKey>
      <cp:key>ocppcentraladdress</cp:key>
      <cp:readonly>false</cp:readonly>
      <cp:value>http://alize.bouygues-es.com:9998/ocpp_soap/ocpp1_5</cp:value>
    </cp:configurationKey>
    <cp:configurationKey>
      <cp:key>ocppboxlocalport</cp:key>
      <cp:readonly>false</cp:readonly>
      <cp:value>8080</cp:value>
    </cp:configurationKey>
      <cp:key>operatingmode</cp:key>
      <cp:readonly>false</cp:readonly>
      <cp:value>2</cp:value>
    </cp:configurationKey>
    <cp:configurationKey>
      <cp:configurationKey>
      <cp:configurationKey>
      <cp:key>voltagererefence</cp:key>
      <cp:readonly>false</cp:readonly>
      <cp:value>400</cp:value>
      (…)
  </cp:getConfigurationResponse>
</SOAP-ENV:Body>
```

Appendix B – Tests with Company B

B.1 Tests with Company B - Phase 1

B.1.1 Start-up of the charge point

BootNotification

```
   <SOAP-ENV:Body>
     <ocppCs15:bootNotificationRequest>
       <ocppCs15:chargePointVendor>Company B</ocppCs15:chargePointVendor>
       <ocppCs15:chargePointModel>NQC-ACDC-COMBO</ocppCs15:chargePointModel>
       <ocppCs15:chargePointSerialNumber>gir.vat.mx.01acd4</ocppCs15:chargePointSerialNumber>
       <ocppCs15:chargeBoxSerialNumber>gir.vat.mx.01acd4</ocppCs15:chargeBoxSerialNumber>
       <ocppCs15:firmwareVersion>1.2.8</ocppCs15:firmwareVersion>
       <ocppCs15:iccid />
       <ocppCs15:imsi />
       <ocppCs15:meterType>NQC-ACDC-COMBO-AO-UR</ocppCs15:meterType>
       <ocppCs15:meterSerialNumber>gir.vat.mx.01acd4</ocppCs15:meterSerialNumber>
     </ocppCs15:bootNotificationRequest>
   </SOAP-ENV:Body>
```

```
<v:Body>
  <cs:bootNotificationResponse>
    <cs:currentTime>2014-12-09T13:35:16.533Z</cs:currentTime>
    <cs:heartbeatInterval>300</cs:heartbeatInterval>
    <cs:status>Accepted</cs:status>
  </cs:bootNotificationResponse>
</v:Body>
```

StatusNotification connectorId 2

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
     <ocppCs15:timestamp>2014-12-09T13:35:16.533Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification connectorId 3

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:36:18Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification connectorId 0

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>0</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:36:24Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification connectorId 1

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Faulted</ocppCs15:status>
    <ocppCs15:errorCode>OtherError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:36:25Z</ocppCs15:timestamp>
    <ocppCs15:vendorId>com.klervi.kvcbx.nqc</ocppCs15:vendorId>
    <ocppCs15:vendorErrorCode>0x050d</ocppCs15:vendorErrorCode>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification connectorId 3

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:36:25Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification connectorId 1

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:36:30Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StatusNotification connectorId 3

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:36:30Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

HeartBeat

```
<SOAP-ENV:Body>
  <ocppCs15:heartbeatRequest />
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:heartbeatResponse>
```

```
    <cs:currentTime>2014-12-09T13:46:45.166Z</cs:currentTime>
  </cs:heartbeatResponse>
</v:Body>
```

B.1.2 Regular Charging session

Authorize → Invalid

```
<SOAP-ENV:Body>
  <ocppCs15:authorizeRequest>
   <ocppCs15:idTag>5EADFA47</ocppCs15:idTag>
  </ocppCs15:authorizeRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:authorizeResponse>
    <cs:idTagInfo>
      <cs:status>Invalid</cs:status>
    </cs:idTagInfo>
  </cs:authorizeResponse>
</v:Body>
```

Authorize → Accepted

```
<SOAP-ENV:Body>
  <ocppCs15:authorizeRequest>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
  </ocppCs15:authorizeRequest>
</SOAP-ENV:Body>

<v:Body>
  <cs:authorizeResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
  </cs:authorizeResponse>
</v:Body>
```

StartTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:startTransactionRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2014-12-09T13:16:47Z</ocppCs15:timestamp>
    <ocppCs15:meterStart>0</ocppCs15:meterStart>
  </ocppCs15:startTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24299</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Occupied</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:17:11Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:17:11Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:17:11Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

MeterValues

```
<SOAP-ENV:Body>
  <ocppCs15:meterValuesRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:transactionId>24299</ocppCs15:transactionId>
    <ocppCs15:values>
      <ocppCs15:timestamp>2014-12-09T13:17:47Z</ocppCs15:timestamp>
      <ocppCs15:value>400</ocppCs15:value>
    </ocppCs15:values>
  </ocppCs15:meterValuesRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCs15:meterValuesRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:transactionId>24299</ocppCs15:transactionId>
    <ocppCs15:values>
      <ocppCs15:timestamp>2014-12-09T13:19:47Z</ocppCs15:timestamp>
      <ocppCs15:value>1400</ocppCs15:value>
    </ocppCs15:values>
  </ocppCs15:meterValuesRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

Authorize → Invalid

```
<SOAP-ENV:Body>
  <ocppCs15:authorizeRequest>
    <ocppCs15:idTag>5EADFA47</ocppCs15:idTag>
  </ocppCs15:authorizeRequest>
</SOAP-ENV:Body>
<v:Body>
```

```
 <cs:authorizeResponse>
    <cs:idTagInfo>
      <cs:status>Invalid</cs:status>
    </cs:idTagInfo>
  </cs:authorizeResponse>
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
```

```
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:20:19Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StopTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:stopTransactionRequest>
    <ocppCs15:transactionId>24299</ocppCs15:transactionId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2014-12-09T13:20:26Z</ocppCs15:timestamp>
    <ocppCs15:meterStop>1700</ocppCs15:meterStop>
  </ocppCs15:stopTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:20:29Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCs15:statusNotificationRequest>
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
    <ocppCs15:errorCode>NoError</ocppCs15:errorCode>
    <ocppCs15:timestamp>2014-12-09T13:20:29Z</ocppCs15:timestamp>
  </ocppCs15:statusNotificationRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

B.1.3 Simulation perte de communication

StartTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:startTransactionRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2014-12-09T13:48:24Z</ocppCs15:timestamp>
    <ocppCs15:meterStart>0</ocppCs15:meterStart>
  </ocppCs15:startTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24303</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

StatusNotification connectorId 1,2 and 3

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Occupied</ocppCs15:status>
```

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

STOP OF THE COMMUNICATION (between 13:48 and 13:58)

Authorize → Invalid

```
<SOAP-ENV:Body>
  <ocppCs15:authorizeRequest>
    <ocppCs15:idTag>5EADFA47</ocppCs15:idTag>
  </ocppCs15:authorizeRequest>
</SOAP-ENV:Body>
<v:Body>
```

```
  <cs:authorizeResponse>
    <cs:idTagInfo>
      <cs:status>Invalid</cs:status>
    </cs:idTagInfo>
  </cs:authorizeResponse>
</v:Body>
```

StatusNotification connectorId 1

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

StopTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:stopTransactionRequest>
    <ocppCs15:transactionId>24303</ocppCs15:transactionId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2014-12-09T13:49:29Z</ocppCs15:timestamp>
    <ocppCs15:meterStop>300</ocppCs15:meterStop>
  </ocppCs15:stopTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification connectorId 2 and 3

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

StartTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:startTransactionRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2014-12-09T13:50:00Z</ocppCs15:timestamp>
    <ocppCs15:meterStart>0</ocppCs15:meterStart>
  </ocppCs15:startTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24304</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

StatusNotification connectorId 1,2 and 3

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Occupied</ocppCs15:status>
```

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

MeterValues

```
<SOAP-ENV:Body>
  <ocppCs15:meterValuesRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:transactionId>24304</ocppCs15:transactionId>
    <ocppCs15:values>
      <ocppCs15:timestamp>2014-12-09T13:51:00Z</ocppCs15:timestamp>
      <ocppCs15:value>300</ocppCs15:value>
    </ocppCs15:values>
  </ocppCs15:meterValuesRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:meterValuesResponse />
</v:Body>
```

StatusNotification connectorId 1

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

StopTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:stopTransactionRequest>
    <ocppCs15:transactionId>24304</ocppCs15:transactionId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2014-12-09T13:51:22Z</ocppCs15:timestamp>
    <ocppCs15:meterStop>400</ocppCs15:meterStop>
  </ocppCs15:stopTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification connectorId 2 and 3

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

B.2 Tests with Company B - Phase 2

B.2.1 RemoteStartTransaction – RemoteStopTransaction

RemoteStartTransaction

```
<v:Body>
  <cp:remoteStartTransactionRequest>
    <cp:connectorId>1</cp:connectorId>
    <cp:idTag>A60AD65D</cp:idTag>
  </cp:remoteStartTransactionRequest>
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCp15:remoteStartTransactionResponse>
    <ocppCp15:status>Accepted</ocppCp15:status>
  </ocppCp15:remoteStartTransactionResponse>
</SOAP-ENV:Body>
```

StartTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:startTransactionRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2015-01-08T15:51:31Z</ocppCs15:timestamp>
    <ocppCs15:meterStart>0</ocppCs15:meterStart>
  </ocppCs15:startTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24445</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

StatusNotification connectorId 1

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Occupied</ocppCs15:status>
```

StatusNotification connectorId 2

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

StatusNotification connectorId 3

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

RemoteStopTransaction

```
<v:Body>
  <cp:remoteStopTransactionRequest>
    <cp:transactionId>24445</cp:transactionId>
  </cp:remoteStopTransactionRequest>
</v:Body>
```

```
   <SOAP-ENV:Body>
     <ocppCp15:remoteStopTransactionResponse>
       <ocppCp15:status>Accepted</ocppCp15:status>
     </ocppCp15:remoteStopTransactionResponse>
   </SOAP-ENV:Body>
```

StatusNotification connectorId 1

```
       <ocppCs15:connectorId>1</ocppCs15:connectorId>
       <ocppCs15:status>Available</ocppCs15:status>
```

StopTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:stopTransactionRequest>
    <ocppCs15:transactionId>24445</ocppCs15:transactionId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2015-01-08T15:52:18Z</ocppCs15:timestamp>
    <ocppCs15:meterStop>200</ocppCs15:meterStop>
  </ocppCs15:stopTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification connectorId 2

```
 <ocppCs15:connectorId>2</ocppCs15:connectorId>
 <ocppCs15:status>Available</ocppCs15:status>
```

StatusNotification connectorId 3

```
 <ocppCs15:connectorId>3</ocppCs15:connectorId>
 <ocppCs15:status>Available</ocppCs15:status>
```

B.2.2 ChangeAvailability

```
<v:Body>
  <cp:changeAvailabilityRequest>
    <cp:connectorId>1</cp:connectorId>
    <cp:type>Inoperative</cp:type>
  </cp:changeAvailabilityRequest>
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCp15:changeAvailabilityResponse>
    <ocppCp15:status>Accepted</ocppCp15:status>
  </ocppCp15:changeAvailabilityResponse>
</SOAP-ENV:Body>
```

B.2.3 Reset Hard

```
<v:Body>
  <cp:resetRequest>
    <cp:type>Hard</cp:type>
  </cp:resetRequest>
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCp15:resetResponse>
    <ocppCp15:status>Accepted</ocppCp15:status>
  </ocppCp15:resetResponse>
</SOAP-ENV:Body>
```

B.2.4 UnlockConnector

StartTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:startTransactionRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2015-01-08T16:03:19Z</ocppCs15:timestamp>
    <ocppCs15:meterStart>1</ocppCs15:meterStart>
  </ocppCs15:startTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>24448</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

StatusNotification connectorId 1

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Occupied</ocppCs15:status>
```

StatusNotification connectorId 2

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

StatusNotification connectorId 3

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Unavailable</ocppCs15:status>
```

UnlockConnector

```
<v:Body>
  <cp:unlockConnectorRequest>
    <cp:connectorId>1</cp:connectorId>
  </cp:unlockConnectorRequest>
</v:Body>
```

```
<SOAP-ENV:Body>
  <ocppCp15:unlockConnectorResponse>
    <ocppCp15:status>Accepted</ocppCp15:status>
  </ocppCp15:unlockConnectorResponse>
</SOAP-ENV:Body>
```

StatusNotification

```
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

StopTransaction

```
<SOAP-ENV:Body>
  <ocppCs15:stopTransactionRequest>
    <ocppCs15:transactionId>24448</ocppCs15:transactionId>
    <ocppCs15:idTag>A60AD65D</ocppCs15:idTag>
    <ocppCs15:timestamp>2015-01-08T16:04:17Z</ocppCs15:timestamp>
    <ocppCs15:meterStop>300</ocppCs15:meterStop>
  </ocppCs15:stopTransactionRequest>
</SOAP-ENV:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification connectorId 2

```
    <ocppCs15:connectorId>2</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

StatusNotification connectorId 3

```
    <ocppCs15:connectorId>3</ocppCs15:connectorId>
    <ocppCs15:status>Available</ocppCs15:status>
```

Appendix C – Tests with Company C

C.1 StopTransaction Problem

Authorize

```
     <soap:Body>
       <authorizeRequest xmlns="urn://Ocpp/Cs/2012/06/">
         <idTag>06208FC0</idTag>
       </authorizeRequest>
     </soap:Body>
```

```
  <v:Body>
    <cs:authorizeResponse>
      <cs:idTagInfo>
        <cs:status>Accepted</cs:status>
      </cs:idTagInfo>
    </cs:authorizeResponse>
  </v:Body>
```

StartTransaction

```
<soap:Body>
  <ocppCs15:startTransactionRequest>
    <ocppCs15:connectorId>1</ocppCs15:connectorId>
    <ocppCs15:idTag>06208FC0</ocppCs15:idTag>
    <ocppCs15:timestamp>2015-02-18T10:17:32.000Z</ocppCs15:timestamp>
    <ocppCs15:meterStart>0</ocppCs15:meterStart>
  </ocppCs15:startTransactionRequest>
</soap:Body>

<v:Body>
  <cs:startTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
    <cs:transactionId>4</cs:transactionId>
  </cs:startTransactionResponse>
</v:Body>
```

StatusNotification

```
<soap:Body>
  <statusNotificationRequest xmlns="urn://Ocpp/Cs/2012/06/">
    <connectorId>1</connectorId>
    <status>Occupied</status>
    <errorCode>NoError</errorCode>
  </statusNotificationRequest>
</soap:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StopTransaction

```
<soap:Body>
  <stopTransactionRequest xmlns="urn://Ocpp/Cs/2012/06/">
    <transactionId>4</transactionId>
    <idTag>06208FC0</idTag>
    <timestamp>2015-02-18T10:18:46.000Z</timestamp>
    <meterStop>907559</meterStop>
  </stopTransactionRequest>
</soap:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse />
</v:Body>
```

StatusNotification

```
<soap:Body>
  <statusNotificationRequest xmlns="urn://Ocpp/Cs/2012/06/">
    <connectorId>1</connectorId>
    <status>Available</status>
    <errorCode>NoError</errorCode>
  </statusNotificationRequest>
</soap:Body>
```

```
<v:Body>
  <cs:statusNotificationResponse />
</v:Body>
```

StopTransaction

```
<soap:Body>
  <stopTransactionRequest xmlns="urn://Ocpp/Cs/2012/06/">
    <transactionId>4</transactionId>
    <idTag>06208FC0</idTag>
    <timestamp>2015-02-18T10:18:46.000Z</timestamp>
    <meterStop>907559</meterStop>
  </stopTransactionRequest>
</soap:Body>
```

```
<v:Body>
  <v:Fault>
    <v:Code>
      <v:Value>v:Receiver</v:Value>
      <v:Subcode>
        <v:Value>cs:InternalError</v:Value>
      </v:Subcode>
    </v:Code>
    <v:Reason>
      <v:Text xml:lang="en-GB">An internal error occurred and the receiver was not able to process the
                             requested Action successfully.</v:Text>
    </v:Reason>
  </v:Fault>
</v:Body>
```

C.2 StopTransaction Problem Solved

StopTransaction

```
<soap:Body>
  <stopTransactionRequest xmlns="urn://Ocpp/Cs/2012/06/">
    <transactionId>8</transactionId>
    <idTag>06208FC0</idTag>
    <timestamp>2015-02-23T10:12:37.000Z</timestamp>
    <meterStop>908324</meterStop>
  </stopTransactionRequest>
</soap:Body>
```

```
<v:Body>
  <cs:stopTransactionResponse>
    <cs:idTagInfo>
      <cs:status>Accepted</cs:status>
    </cs:idTagInfo>
  </cs:stopTransactionResponse>
</v:Body>
```