Protocoldetails

Smart Charging in Practice



Draft version

Versie	Gewijzigd door	Gewijzigd op	Wijziging
C0.1 - C0.8	Paul Klapwijk	27-05-2013 28-6-2013	Eerste opzet incl. aanpassingen n.a.v. GAP analyse e-laad (Klaas van Zuuren)
C0.84	Paul Klapwijk	26-7-2013	
C0.9	Lennart Verheijen	29-7-2013	Translated the document to English Also changed the priority description (again)
C0.91	Paul Klapwijk	9-8-2013	Made block-size and unit variable within the messages. Adjusted picture in chapter 2.
C0.92	Paul Klapwijk	16-8-2013	Changed UnitOfMeasure into attribute for OSCP messages (similar to OCPP)
C0.93	Paul Klapwijk	20-9-2013	Changed field Primary to primary in message SetChargingProfile. Removed all information irrelevant for the protocol.

Inhoud

1. Introduction 4
   1.1 Goal 4
   1.2 Scope 4
   1.3 Terms and acronyms 4
   1.4 References / sources 4
2. High level overview process / message flow 5
3. Interfaces 7
   3.1 General 7
   3.2 Architecture 7
   3.3 Security 7
   3.4 Messages 8
      3.4.1 General information in header 8
      3.4.2 UpdateCableCapacityForecast 8
      3.4.3 SetChargingProfile (IslandingCapacity) 10
         3.4.3.1 Header (uit OCPP 1.5 specificatie) 10
         3.4.3.2 SetChargingProfile.req (uit OCPP 2.0 specificatie) 10
      3.4.4 GetCapacityForecast 16
         3.4.4.1 GetCapacityForecastRequestMessage 16
         3.4.4.2 GetCapacityForecastResponseMessage 16
      3.4.5 SetChargingProfile (UpdateChargePlan) 17
      3.4.6 RequestAdjustedCapacity 17
         3.4.6.1 RequestAdjustedCapacityRequestMessage 17
         3.4.6.2 RequestAdjustedCapacityResponseMessage 18
      3.4.7 UpdateAggregatedUsage 19
      3.4.8 MeterValues 20
         3.4.8.1 MeterValues.req (uit OCPP 1.5 specificatie) 20
         3.4.8.2 MeterValues.conf (uit OCPP 1.5 specificatie) 23
      3.4.9 Response messages 23
4. Example messages 24
   4.1 UpdateCableCapacityForecast 24
   4.2 SetChargingProfile.req 25
   4.3 RequestAdjustedCapacityRequestMessage 26
   4.4 RequestAdjustedCapacityResponseMessage 26

Draft Version

4.5 UpdateAggregatedUsage 26
4.6 MeterValues.req 27
5. Appendix 1: open points 28
6. Appendix 2: Incorporate OCPP 2.0 in original design 29

Draft Version

1. Introduction

1.1 Goal

The goal of this document is to describe the protocol as defined in the reference model for Smart Charging in Practice (SCiP) in more detail. Since in an innovative concept like this it is impossible to have a clear view on all the details at the start of the project, this document will be finalized at the end of the smart charging project (Q1 or Q2 2014).

1.2 Scope

This document describes the message flow defined in the reference model for SCiP. This documents is limited to the communication between:

- DSO and CSP
- CSP and Local Controller

1.3 Terms and acronyms

Term / acronym*	Meaning
CP Module	**C**ommunication **P**rotocol **Module**. This is a module in the charge spot which enables the charge spot to receive messages from the outside world
CSP	**C**harge **S**ervice **P**rovider. The CSP is defined as the party that pays for the electricity with which the EV is charged
DSO	**D**istribution **S**ystem **O**perator
EV	**E**lectric **V**ehicle
Local Controller (LC)	Component that communicates with the CP modules in the charge spot and is able to change the maximum charge current in the charge spot. The LC is controlled by the CSP (but can also operate in island mode)
SCiP	**S**mart **C**harging **i**n **P**ractice

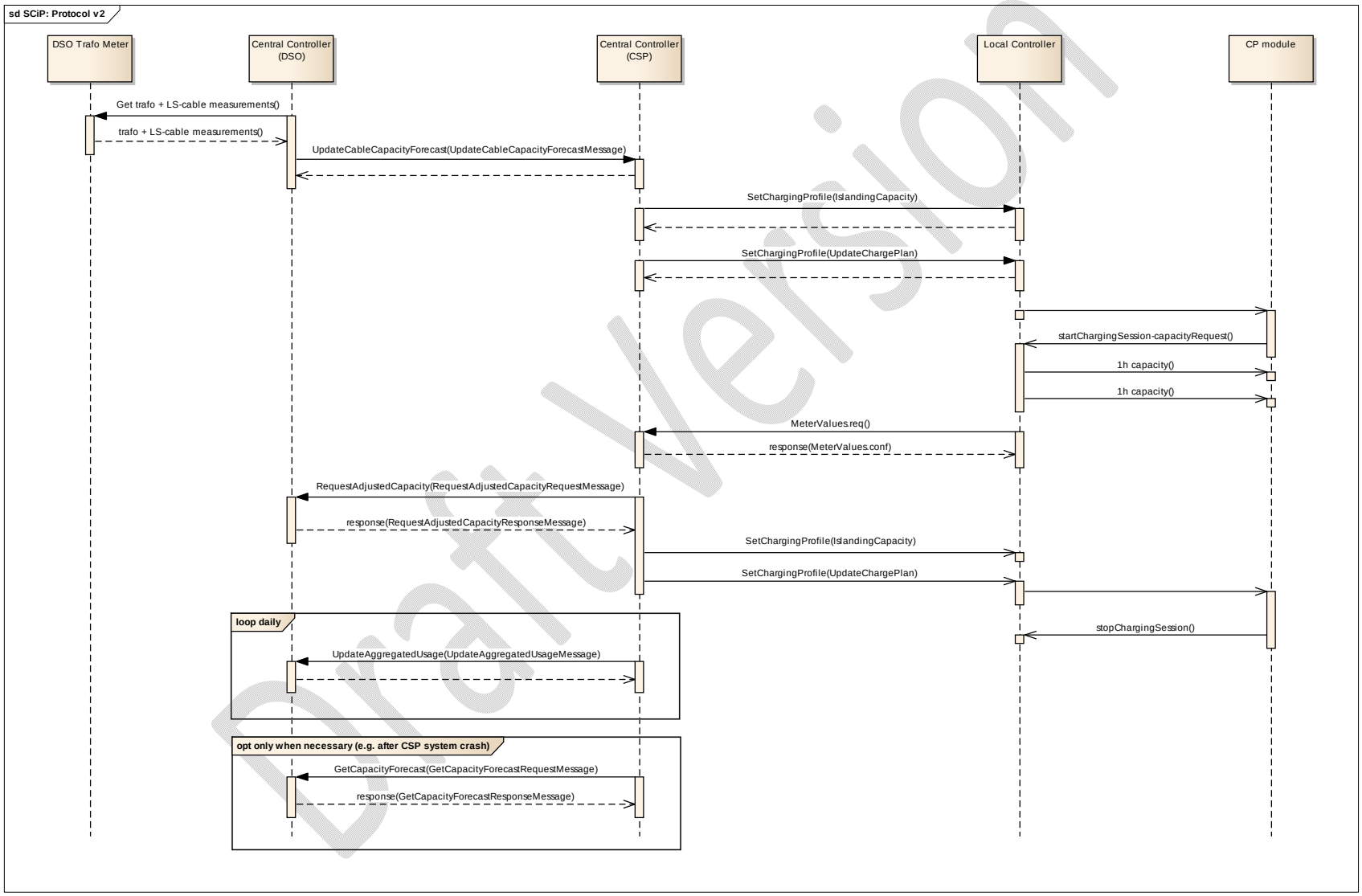
* See “Systeembeschrijving SCiP ReferentieModel” [1] for more details about the functionalities that belong to the different components/parties in the achitecture

1.4 References / sources

No.	Titel	Version	Autor	Date
1	Systeembeschrijving SCiP ReferentieModel	6	GreenFlux / Enexis	4-6-2013
2	Open Charge Point Protocol	1.5	e-laad	8-6-2012
3	Open Charge Point Protocol	2.0 (draft)	e-laad	15-4-2013

Draft Version

2. High level overview process / message flow



Draft Version

Explanation:

In the overview above the message flow between the different parties is depicted. The initial design has been compared with the draft version of OCPP 2.0. Based on this comparison several messages from the original design have been replaced by messages from OCPP 2.0. See “Appendix 2: incorporate OCPP 2.0 in the original design” for more information.

The process as described in the diagram above exists roughly of the following steps/parts:

1. The DSO receives measurements from the transformer station. This is displayed in the diagram as “Trafo + LS-cable measurements”.
2. Based on these measurements a forecast of the cable capacity is made, which is forwarded to the CSP (UpdateCableCapacityForecast).
3. Based on this forecast the CSP can tell the Local Controller what its capacity for the next period of time will be in case the LC needs to go to island mode. This island mode occurs only in case the LC does not receive forecasts from the CSP / DSO (eg. in case of technical problems). For forwarding the forecast with regard to island mode the message (SetChargingProfile) can be used, which comes from the draft of the OCPP 2.0 protocol.
4. Subsequently the CSP (based on its customers wishes) can tell de Local Controller how to charge the EVs connected to the LC. Again the message SetChargingProfile can be used for this. With respect to priority: in general the LC will execute the charge plan it receives from the CSP. This charge plan has priority over the forecast for Island mode.
5. Subsequently the EVs at the charge spot(s) connected to the LC can be charged. To enable this, communication is necessary between de LC and the CP Module. The LC can set a maximum charge current for the charger in the EV, which results in the EV not charging at full capacity.
6. Using the message MeterValues, the LC can indicate that he is using less capacity than he was allowed to according to the CSP
7. If the CSP notices there is not enough capacity available for a LC, there is the possibility to request extra capacity at the DSO. This can be done using the message “RequestExtraCapacity”.
8. Every period of time, the CSP needs to tell the DSO what its actual consumption has been. The DSO needs this information in order to improve the algorithms for both the calculation of the capacity forecast and the division of capacity over the different CSP’s. Both these algorithms need historical data.
9. In case there hase been an issue on the side of the CSP or DSO, the CSP had the possibility to request a new forecast. This is to prevent the CSP having to wait for the next forecast.

Draft Version

3. Interfaces

3.1 General

In this chapter the messages are discussed in more detail. Paragraph 3.2 describes the (technical) architecture choices which have not yet been defined in “Systeembeschrijving SCiP ReferentieModel” [1]. Paragraph 3.3 explores safety aspects. Paragraph 3.4 describes the messages which are necessary to implement the process described in Chapter 2.

3.2 Architecture

Most of the architecture has been discussed in “Systeembeschrijving SCiP ReferentieModel” [1]. What is not yet defined is the manner of communication between the different parties.

From	To	Communication-protocol	Description
DSO TRAFO meter	DSO Backend System	REST-like	The information stream is from transformer meter to the DSO, but the technical communication is the other way around: the DSO polls the meter.
DSO Backend System	CSP Backend System	SOAP / http	This holds both for traffic initiated by the DSO as by the CSP.
CSP Backend System	Local Controller	To be defined	

3.3 Security

Van	Naar	Beveiliging	Toelichting
DSO TRAFO meter	DSO Backend Systeem	??	??
DSO Backend Systeem	CSP backend systeem	https (*2-way-SSL*)	For the development, test and acceptance-environment self signed certificates will be used. For the production-environment an official CA certificate will be used.
CSP backend systeem	Local Controller	??	??

Draft Version

3.4 Messages

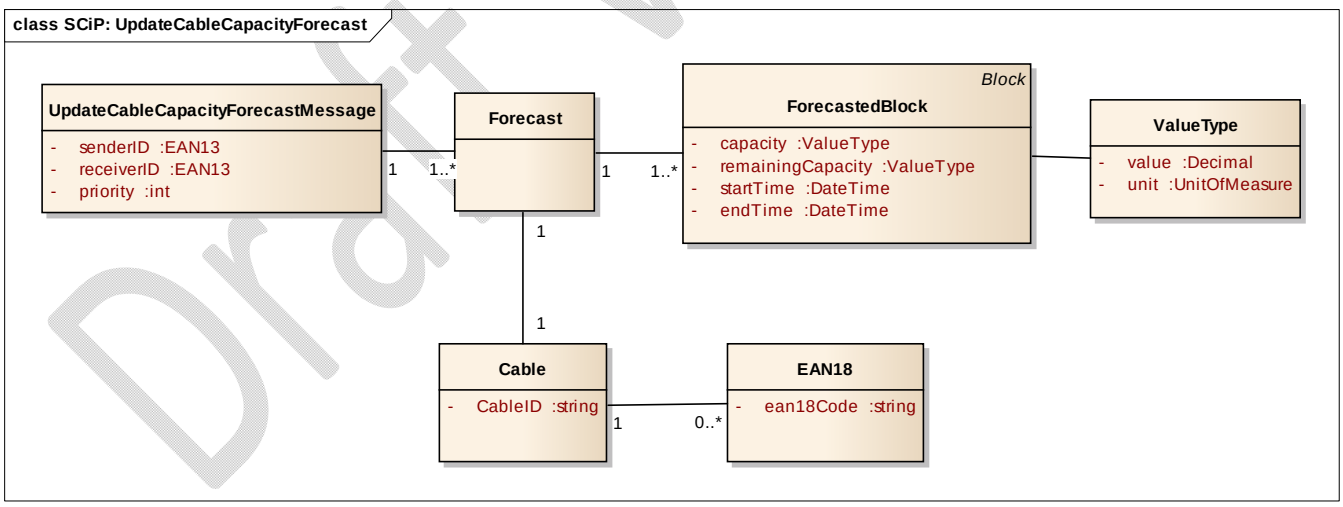
3.4.1 General information in header

For all messages sent between the different parties, the sending party will include in the header both the receiving party and the priority of the message.

For the identification of the sender / receiver it's custom in the energy market to use an EAN code (European Article Number) containing 13 digits. Note: this is not the same as the identifier of the physical connection, as the code for this connection contains 18 digits. For both identifiers holds that the last digit is a control-digit. An example of the information in the header is:

```
<soap:Header>
   <ns:Priority>1</ns:Priority>
   <ns:Identification>
      <ns:DSO>8712423014022</ns:DSO>
      <ns:CSP>8712423222120</ns:CSP>
      <ns:DateTime>2013-06-28T00:00:00.000Z</ns:DateTime>
      <ns:EventId>uuid</ns:EventId>
   </ns:Identification>
</soap:Header>
```

3.4.2 UpdateCableCapacityForecast



Field Name	Field Type	Card.	Description
foreCastedBlock	ForeCastedBlock	1..*	
CableID	String	1..1	Cable where the CSP has connections for EV, connections for households excluded
EAN18	String (18)	0..*	Connections of the CSP, connections for households excluded

Draft Version

ForeCastedBlock

Field Name	Field Type	Card.	Description
capacity	ValueType	1..1	The value is the maximum *forecasted value*
remainingCapacity	ValueType	1..1	The remainingCapacity is the capacity which the DSO has available as backup capacity, to be able to serve parties which indicate to need extra capacity. For dividing the capacity over the different CSPs the DSO uses an algorithm based on total connected power and historical consumption. The backup capacity is to cope with unexpected events. If it turns out that this backup capacity is frequently insufficient the DSO might have to expand its network capacity.
startTime	DateTime	1..1	Start time of this block
endTime	DateTime	1..1	End time of this block

ValueType:

Field Name	Field Type	Card.	Description
value	Decimal	1..1	The value is the maximum *forecasted value*
attribute:unit	UnitOfMeasure	0..1	UnitOfMeasure, unit of the value. Default = “Amp” (= Ampère per phase)

UnitOfMeasure

Enumeration

Value	Description
Wh	Watt-hours (energy). Default.
kWh	kiloWatt-hours (energy).
varh	Var-hours (reactive energy).
kvarh	kilovar-hours (reactive energy).
W	Watts (power).
kW	kilowatts (power).
var	Vars (reactive power).
kvar	kilovars (reactive power).
Amp	Amperes (current).
Volt	Voltage (r.m.s. AC).
Celsius	Degrees (temperature).

Draft Version

3.4.3 SetChargingProfile (IslandingCapacity)

This message concerns information sent to the Local Controller for the case of Island mode.

3.4.3.1 Header (uit OCPP 1.5 specificatie)

The OCPP 1.5 specification, paragraph 8.1, states the following with respect to the identification of a charge point:

To be able for the Central System to identify uniquely a Charge Box, a Charge Box MUST send its identifier in the SOAP header of each request PDU. The header name "chargeBoxIdentity" SHALL be treated as case insensitive. For example:

```
<!-- Header with the identifier of the sending Charge Box -->
<ns:chargeBoxIdentity>CB1234</ns:chargeBoxIdentity>
```

When a Central System needs to send requests to a Charge Box, the Central System MUST send in each request the "chargeBoxIdentity" for which Charge Box the request is intended. If the receiving Charge Box is not the intended one, then the Charge Box MUST send a SOAP Fault Response message, indicating that the identity is wrong (e.g. sub-code is "IdentityMismatch").

Therefore, in the messages from CSP to LC, this needs to be part of the SOAP header!!

3.4.3.2 SetChargingProfile.req (uit OCPP 2.0 specificatie)

The message below has been copied directly from the OCPP 2.0 specification, paragraph 1.7.4.2 ([3]).

Field name	Field Type	Card.	Description
connectorId	integer	1..1	Mandatory. The limits described in the schedule are imposed onto the electric vehicle plugged into the outlet identified by ConnectorId. If ConnectorId = 0, the message contains an overall limit for the charge point.
chargingProfileId	integer	0..1	Optional. Non-Zero ID of the limit schedule. The ID shall be changed whenever the schedule is updated. Mandatory if ConnectorId = 0, shall

Draft Version

			not be used if ConnectorID > 0. Adopted from [5, 6.2.4]
saChargingProfiles	ChargingProfile	1..3	Includes several tuples of limit schedules of secondary actors. If ISO15118-2 is used, multiple schedules with prices might be offered to the car. @@NOTE (FB): Are secondary actors e-mobility providers? The current version of ISO15118-2 limits the number of limit schedules to 3 [3, V2G2-295]. If the schedule is addressed to the charge point, the selection between the limit schedules will be done based on the charging scheduling strategy of the charge point.
saChargingProfileKind	SAChargingProfileKindType	1..1	Mandatory. Indicates the kind of schedule. @@NOTE (FB): Is it correct that there can be only 1 charging profile kind, but 1 to 3 charging profiles? In other words, they must all be of the same kind?
recurrencyKind	RecurrencyKindType	0..1	Optional. Indicates the start point of a recurrence.
validFrom	DateTime	0..1	Optional. Point in time at which the schedule starts to be valid. If empty, the schedule is valid as soon as it is received by the charge point. For an absolute schedule, all periods are relative to this point in time.
validTo	DateTime	0..1	Optional. Point in time at which the schedule stops to be valid. If empty, the schedule is valid until it replaced by another schedule.

SAChargingProfileKindType

Enumeration

Draft Version

Value	Description
ABSOLUTE	Schedule periods are relative to a fixed point in time defined in the schedule.
RELATIVE	Schedule periods are relative to a situation-specific start point (such as the start of a session) that is determined by the charge point.
RECURRING	The schedule restarts periodically at the first schedule period.

RecurrencyKindType

Enumeration

Value	Description
DAILY	Daily: The schedule restarts at the beginning of the next day.
WEEKLY	The schedule restarts at the beginning of the next week (defined as Monday morning)

ChargingProfile

A ChargingProfile consists of a ChargingSchedule, describing the amount of power that can be delivered per time interval, and optionally one or more SalesTariffTables.

Field Name	Field Type	Card.	Description
chargingProfileId	integer	1..1	Mandatory. Unique identifier within a LimitSchedule list.
primary	boolean	1..1	Mandatory. Indicates whether the limit schedule is the primary limit schedule or an alternative limit schedule that can be offered during negotiation using ISO15118. A Change Limit Schedule Request contains exactly one primary limit schedule. If an EV charges controlled by PWM (mode 3) or doesn't support multiple sales tariff tables when charging with ISO15118, the primary limit schedule shall be used. *Note:* The term "alternative limit schedule" is used for limit schedules with primary = false.
chargingSchedule	ChargingSchedule	1..1	Mandatory. Contains limits for the available power over time.
salesTariff	SalesTariffTable	0..1	Optional. Contains indications of the

Draft Version

			relative value of energy over time (for example price or percentage of green energy). Mandatory for an alternative limit schedule.

ChargingSchedule

Field Name	Field Type	Card.	Description
chargingScheduleId	integer	1..1	Mandatory. Unique identifier for a charging schedule.
duration	integer	0..1	Optional. Duration of the charging schedule. If the duration is left empty, the last period will continue indefinitely or until end of the transaction in case startSchedule is absent.
startSchedule	DateTime	0..1	Optional. Start point of an absolute schedule. If absent the schedule will be relative to start of charging.
chargingSchedulePeriods	ChargingSchedulePeriod	0..*	List of ChargingSchedulePeriod defining maximum power usage over time.

ChargingSchedulePeriod

Field Name	Field Type	Card.	Description
startPeriod	integer	1..1	Mandatory. Start of the period, in seconds from the start of schedule. The value of StartPeriod simultaneously defines the stop time of the previous period.
maxPower	decimal	1..1	Mandatory. Maximum power in Watt that may be delivered during the schedule period. For OSCP: Ampere instead of Watt!
numberPhases	integer	0..1	Optional. The number of phases that can be used for charging. If a number of phases is needed, NumberPhase=3 will be assumed unless another number is given. For calculation purposes, the PWM duty cycle shall be PMax/(NumberPhases*Nominal Voltage), where NominalVoltage is between one phase and neutral.

SalesTariffTable

Field Name	Field Type	Card.	Description

Draft Version

salesTariffId	integer	1..1	Mandatory. Identifier used to identify one sales tariff.
tariffDescription	string(32)	0..1	Optional. A human readable short description (for visualization and reporting).
maxEPriceLevel	integer	1..1	Mandatory. Defines maximum of distinct price levels across all provided sales tariff tables (e.g. on-peak, mid-peak, off-peak).
tariffPeriods	TariffPeriod	1..*	Mandatory. List of TariffPeriod

TariffPeriod

Field Name	Field Type	Card.	Description
startPeriod	integer	1..1	Mandatory. Start of the period, in seconds from the start of schedule. The value of StartPeriod simultaneously defines the stop time of the previous period.
periodDescription	string(32)	0..1	Optional. A human readable short description (for visualization and reporting).
ePriceLevel	integer	1..1	Mandatory. Defines the price level of this period. 0 if comparison of price levels shall not be used (e.g. because payment is handled by external means).
eCost	ConsumptionCost	0..*	Optional. List of consumption costs.
pricingBase	PricingBaseType	0..1	Optional. Base for (monetary) prices given in ECost. The following values are supported: • Energy Consumed (in Kwh) • Flat Rate • Session Duration (in Minutes) • Charging Duration (in Minutes), defined as the time span a car is charging

PricingBaseType

Enumeration

Value	Description
ENERGY_CONSUMED	Energy consumed in kWh
FLAT_RATE	Flat rate pricing
SESSION_DURATION	Session duration in minutes
CHARGING_DURATION	Charging duration in minutes, defined as the time a vehicle is actually charging.

Draft Version

ConsumptionCost

Field Name	Field Type	Card.	Description
entryType	EntryKindType	1..1	Mandatory. Defines whether the entry shall be used for ISO15118 communication or for charging by external means.
startValue	integer	1..1	Mandatory. The lowest level of consumption that defines the starting point for this consumption block (in Watth or minutes). The interval extends to the start of the next interval. Is per Watth if the ConsumptionCost is used for ISO15118 communication.
amount	float	0..1	Optional. The estimated or actual cost per kWh. Mandatory, if the ConsumptionCost element is used for ISO15118 communication.
periodDescription	string(32)	0..1	Optional. A human readable short description (for visualization and reporting).
ePriceLevel	integer	1..1	Mandatory. Defines the price level of this period. 0 if the price level mechanism is not used.
costKind	CostKindType	0..1	Optional. The kind of cost given in eAmount according to ISO-15118-2. Mandatory, if the ConsumptionCost element is for ISO15118 communication.
ePrice	float	0..1	Optional. Price per unit of measurement. Mandatory, if the ConsumptionCostElement is used for advice of charge or pricing. Refers to PricingBase.
pricingInterval	integer	0..1	Optional. Size of the interval (in units given at the PricingBase). Mandatory, if the ConsumptionCostElement is used for advice of charge or pricing. Refers to PricingBase.

CostKindType

Enumeration

Value	Description
REL_PRICE_PERC	Relative price, as percentage relative to a common base between all tariffs.

Draft Version

RENEW_GEN_PERC	Renewable generation percentage, as a percentage of overall generation
CO2_EMISSION	Carbon dioxide emission, in grams per unit of measure (kWh)

EntryKindType

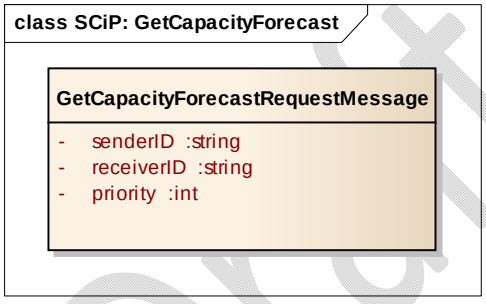
Enumeration

Value	Description
ISO15118	
ADVICE_OF_CHARGE	
EXTERNAL_PRICING	

3.4.4 GetCapacityForecast

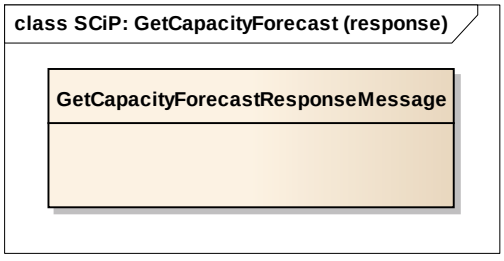
This message is intended to enable a CSP to ask for a new forecast . This service only contains a request to which the answer will be OK. Subsequently the DSO calls the UpdateCableCapacityForecast service for the specific CSP.

3.4.4.1 GetCapacityForecastRequestMessage



See paragraph 3.4.1 for the exact fields.

3.4.4.2 GetCapacityForecastResponseMessage



Draft Version

In this case the reply only contains OK, see 3.4.9 for the standard response fields.

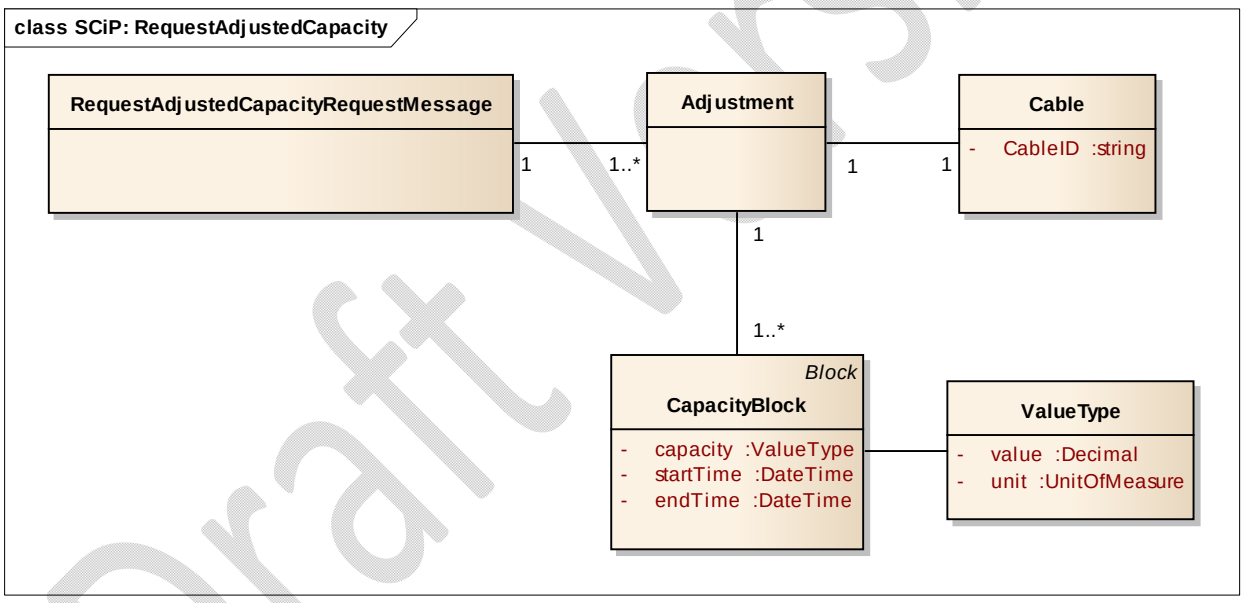
3.4.5 SetChargingProfile (UpdateChargePlan)

For the message to update a charge plan, the message SetChargingProfile is also used.

3.4.6 RequestAdjustedCapacity

3.4.6.1 RequestAdjustedCapacityRequestMessage

This message is intended to enable a CSP to request extra capacity in the case there is not enough



Field Name	Field Type	Card.	Description
CableID	String	1..*	CableID belonging to the connection(s) for which the CSP requests extra capacity.
capacityBlock	**CapacityBlock**	1..*	

Draft Version

CapacityBlock

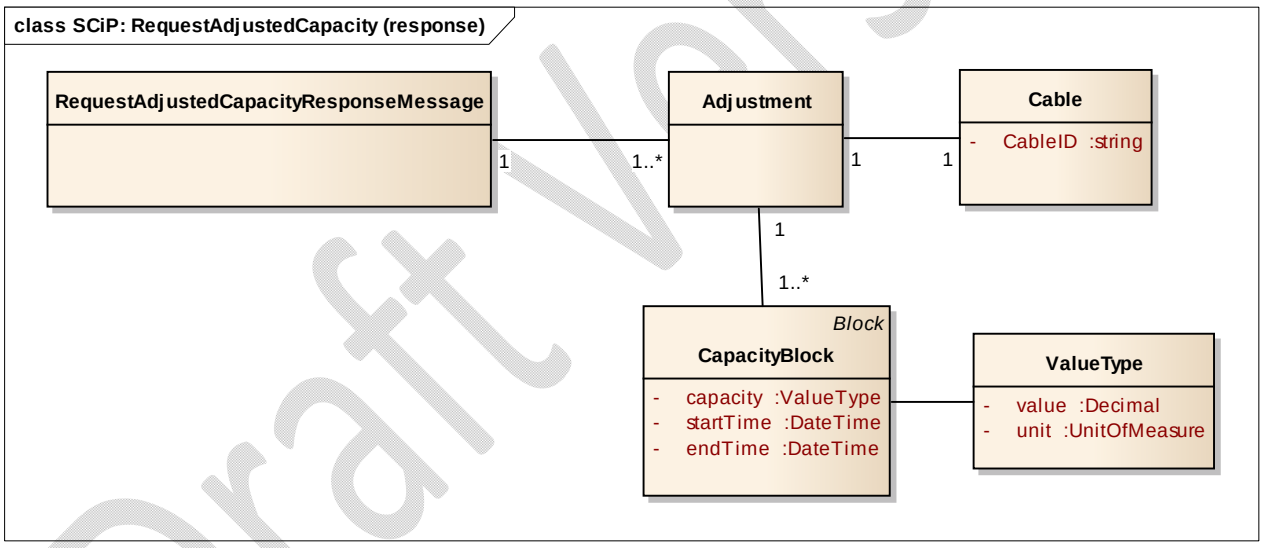
Field Name	Field Type	Card.	Description
capacity	ValueType	1..1	Requested capacity.
startTime	DateTime	1..1	Start time of this block
endTime	DateTime	1..1	End time of this block

ValueType:

See paragraph 3.4.2.

3.4.6.2 RequestAdjustedCapacityResponseMessage

The answer to a request for extra capacity. See 3.4.9 for the standard response fields (not in the picture below).



Field Name	Field Type	Card.	Description
CableID	String	1..*	CableID belonging to the connection(s) for which the CSP requests extra capacity.
capacityBlock	**CapacityBlock**	1..*	

CapacityBlock

Field Name	Field Type	Card.	Description
capacity	ValueType	1..1	Assigned capacity.

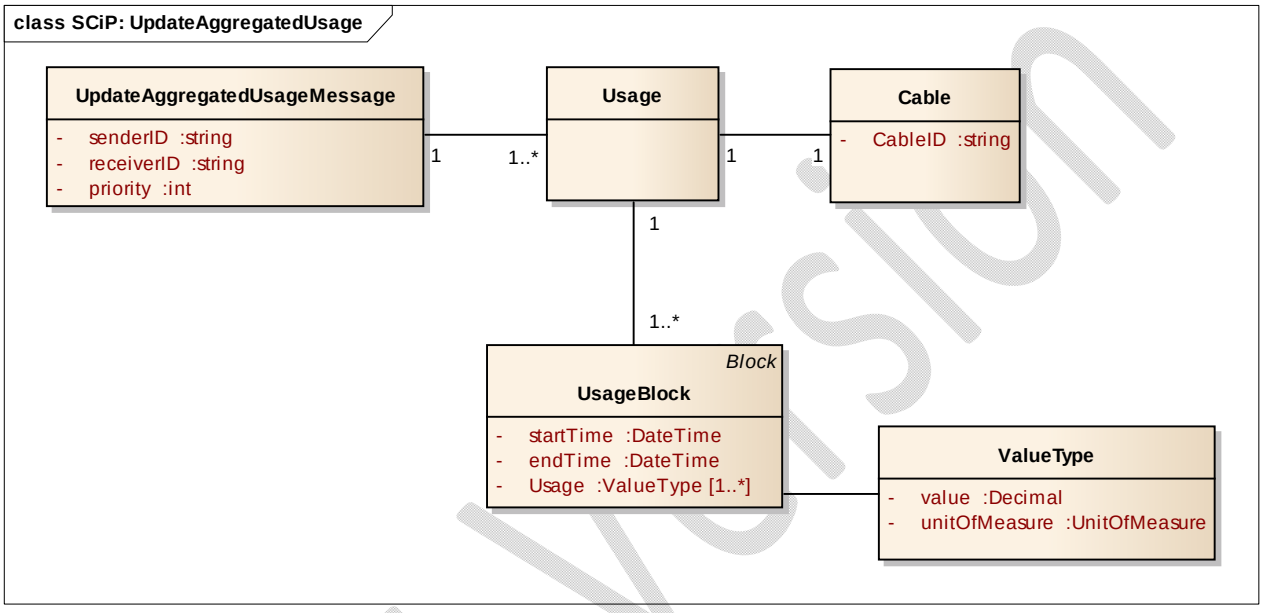
Draft Version

startTime	DateTime	1..1	Start time of this block
endTime	DateTime	1..1	End time of this block

ValueType:

See paragraph 3.4.2.

3.4.7 UpdateAggregatedUsage



Field Name	Field Type	Card.	Description
CableID	String	1..1	CableID beloning to the connection(s) for which the CSP gives the consumed capacity
usageBlock	**UsageBlock**	1..*	

CapacityBlock

Field Name	Field Type	Card.	Description
Usage	ValueType	1..*	Consumed capacity / usage.
startTime	DateTime	1..1	Start time of this block
endTime	DateTime	1..1	End time of this block

Draft Version

ValueType:

See paragraph 3.4.2.

3.4.8 MeterValues

This concerns a message from the LC to the CSP to inform the CSP of the charging status.

3.4.8.1 MeterValues.req (uit OCPP 1.5 specificatie)

This contains the field definition of the MeterValues.req PDU sent by the Charge Box to the Central System.

Field Name	Field Type	Card.	Description
connectorId	integer connectorId >= 0	1..1	Mandatory. This contains a number (>0) designating a connector of the Charge Point.„0" (zero) is used to designate the main power meter.
transactionId	integer	0..1	Optional. The transaction to which these meter samples are related.
values	MeterValue	0..*	Optional. The sampled meter values with timestamps.

MeterValue

Class
Collection of meter values in MeterValues.req. Each value can optionally be accompanied by one or more attributes.

Field Name	Field Type	Card.	Description
timestamp	dateTime	1..1	Mandaroy. Timestamp for measured value(s).
value	string	1..*	Mandatory. Values for one or more measurands. Value as a "Raw" (decimal) number or "SignedData". Field Type is "string" to allow for digitally signed data readings. Decimal numeric values are also acceptable to allow fractional values for measurands such as Temperature and

Draft Version

			Current.
attribute: **context**	ReadingContext	0..1	Optional. Type of detail value: start, end or sample. Default = “Sample.Periodic”
attribute: **format**	ValueFormat	0..1	Optional. Raw or signed data. Default = “Raw”
attribute: **measurand**	Measurand	0..1	Optional. Type of measurement. Default = “Energy.Active.Iimport .Register”
attribute: **location**	Location	0..1	Optional. Location of measurement. Default=”Outlet”
attribute: **unit**	UnitOfMeasure	0..1	Optional. Unit of the value. Default = “Wh” if the (default) measurand is an “Energy” type.

ReadingContext

Enumeration
Values of the context attribute in of a value in MeterValue.

Value	Description
Interruption.Begin	Value taken at start of interruption.
Interruption.End	Value taken when resuming after interruption.
Sample.Clock	Value taken at clock aligned interval.
Sample.Periodic	Value taken as periodic sample relative to start time of transaction.
Transaction.Begin	Value taken at end of transaction.
Transaction.End	Value taken at start of transaction.

ValueFormat

Enumeration
Format that specifies how the value element in MeterValue is to be interpreted.

Value	Description
Raw	Data is to be interpreted as integer/decimal numeric data.
SignedData	Data is represented as a signed binary data block, encoded as hex data.

Draft Version

Measurand

Enumeration
Allowable values of the optional "measurand" attribute of a Value element, as used in MeterValues.req and StopTransaction.req messages. Default value of "measurand" attribute of Value is always "Energy.Active.Import.Register"

Value	Description
Energy.Active.Export.Register	Energy exported by EV (Wh or kWh)
Energy.Active.Import.Register	Energy imported by EV (Wh or kWh)
Energy.Reactive.Export.Register	Reactive energy exported by EV (varh or kvarh)
Energy.Reactive.Import.Register	Reactive energy imported by EV (varh or kvarh)
Energy.Active.Export.Interval	Energy exported by EV (Wh or kWh)
Energy.Active.Import.Interval	Energy imported by EV (Wh or kWh)
Energy.Reactive.Export.Interval	Reactive energy exported by EV. (varh or kvarh)
Energy.Reactive.Import.Interval	Reactive energy imported by EV. (varh or kvarh)
Power.Active.Export	Instantaneous active power exported by EV. (W or kW)
Power.Active.Import	Instantaneous active power imported by EV. (W or kW)
Power.Reactive.Export	Instantaneous reactive power exported by EV. (var or kvar)
Power.Reactive.Import	Instantaneous reactive power imported by EV. (var or kvar)
Current.Export	Instantaneous current flow from EV
Current.Import	Instantaneous current flow to EV
Voltage	AC RMS supply voltage
Temperature	Temperature reading inside charge point.

Location

Enumeration
Allowable values of the optional "location" attribute of a value element in MeterValue.

Value	Description
Inlet	Measurement at network (“grid”) inlet connection
Outlet	Measurement at an EVSE connection outlet (Connector). Default value
Body	Measurement inside body of charge point (e.g. Temperature)

UnitOfMeasure

See paragraph 3.4.2.

Draft Version

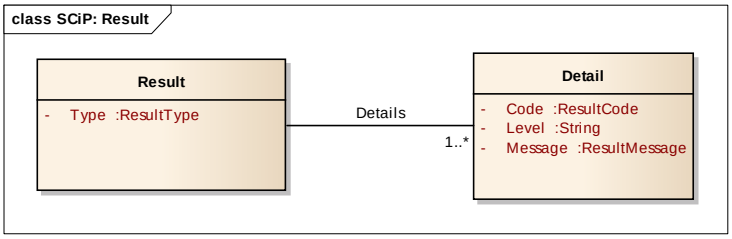
3.4.8.2 MeterValues.conf (uit OCPP 1.5 specificatie)

This contains the field definition of the MeterValues.conf PDU sent by the Central System to the Charge Box as response to a MeterValues.req PDU.

No fields are defined.

3.4.9 Response messages

In response messages the following fixed part is used:



Field Name	Field Type	Card.	Description
Type	**ResultType**	1..1	Indicates success or error response
Details	**Detail**	1..n	Identifier of the response result

ResultType

Enumeration of the following values

Value
SUCCES
ERROR

Detail

Field Name	Field Type	Card.	Description
Code	**ResultCode**	1..1	Identifier of the response result
Level	**String enumeration**	1..1	Loglevel: INFO/WARN/ERROR
Message	**ResultMessage**	1..1	Response message corresponding with response Code

ResultCode

A String code with maximum length of 15.

ResultMessage

A String message with maximum length of 254.

Draft Version

4. Example messages

4.1 UpdateCableCapacityForecast

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://OSCP/CSP/2013/06/">
   <soap:Header>
      <ns:Priority>2</ns:Priority>
      <ns:Identification>
         <ns:DSO>8712423014022</ns:DSO>
         <ns:CSP>8712423222120</ns:CSP>
         <ns:DateTime>2013-06-28T00:00:00.000Z</ns:DateTime>
         <ns:EventId>uuid</ns:EventId>
      </ns:Identification>
   </soap:Header>
   <soap:Body>
      <ns:UpdateCableCapacityForecastRequest>
         <!--1 or more repetitions:-->
         <ns:Forecast>
            <ns:Cable>
               <ns:CableId>kabel-1234</ns:CableId>
               <!--1 or more repetitions:-->
               <ns:Connection>
                  <ns:Ean>871098765432123456</ns:Ean>
               </ns:Connection>
            </ns:Cable>
            <!--1 or more repetitions:-->
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Amp">200</ns:Capacity>
               <ns:StartTime>2013-08-09T12:00:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:15:00.000Z</ns:EndTime>
               <ns:RemainingCapacity UnitOfMeasure="Amp">4</ns:RemainingCapacity>
            </ns:CapacityBlock>
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Amp">250</ns:Capacity>
               <ns:StartTime>2013-08-09T12:15:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:30:00.000Z</ns:EndTime>
               <ns:RemainingCapacity UnitOfMeasure="Amp">8</ns:RemainingCapacity>
            </ns:CapacityBlock>
         </ns:Forecast>
         <ns:Forecast>
            <ns:Cable>
               <ns:CableId>kabel-7890</ns:CableId>
               <!--1 or more repetitions:-->
               <ns:Connection>
                  <ns:Ean>871098765432987654</ns:Ean>
               </ns:Connection>
            </ns:Cable>
            <!--1 or more repetitions:-->
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Amp">100</ns:Capacity>
               <ns:StartTime>2013-08-09T12:00:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:15:00.000Z</ns:EndTime>
               <ns:RemainingCapacity UnitOfMeasure="Amp">8</ns:RemainingCapacity>
            </ns:CapacityBlock>
         </ns:Forecast>

      …

      </ns:UpdateCableCapacityForecastRequest>
```

Draft version

```
   </soap:Body>
</soap:Envelope>
```

4.2 SetChargingProfile.req

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <soap:Header>
            <chargeBoxIdentity
xmlns="urn://Ocpp/Cp/2013/09/">NL_GFX_1068KK7</chargeBoxIdentity>
      </soap:Header>
      <soap:Body>
            <setChargingProfileRequest xmlns="urn://Ocpp/Cp/2013/09/">
                  <connectorId>0</connectorId>
                  <chargingProfileId>698163906</chargingProfileId>
                  <saChargingProfiles>
                        <chargingProfileId>542473912</chargingProfileId>
                        <primary>true</primary>
                        <chargingSchedule>
                              <chargingScheduleId>550741720</chargingScheduleId>
                              <startSchedule>2013-08-07T12:00:00</startSchedule>
                              <chargingSchedulePeriods>
                                    <startPeriod>0</startPeriod>
                                    <maxPower>2000.00</maxPower>
                              </chargingSchedulePeriods>
                        </chargingSchedule>
                  </saChargingProfiles>
                  <saChargingProfileKind>ABSOLUTE</saChargingProfileKind>
                  <recurrencyKind>DAILY</recurrencyKind>
            </setChargingProfileRequest>
      </soap:Body>
</soap:Envelope>
```

Draft Version

4.3 RequestAdjustedCapacityRequestMessage

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://OSCP/DSO/2013/06/">
   <soap:Header>
      <ns:Identification>
         <ns:DSO>8712423014022</ns:DSO>
         <ns:CSP>8712423222120</ns:CSP>
         <ns:DateTime>2013-06-28T00:00:00.00Z</ns:DateTime>
         <ns:EventId>uuid</ns:EventId>
      </ns:Identification>
      <ns:Priority>2</ns:Priority>
   </soap:Header>
   <soap:Body>
      <ns:RequestAdjustedCapacityRequest>
         <!--1 or more repetitions:-->
         <ns:Adjustment>
            <ns:Cable>
               <ns:CableId>kabel-4321</ns:CableId>
            </ns:Cable>
            <!--1 or more repetitions:-->
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Amp">16</ns:Capacity>
               <ns:StartTime>2013-08-09T12:00:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:15:00.000Z</ns:EndTime>
            </ns:CapacityBlock>
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Amp">16</ns:Capacity>
               <ns:StartTime>2013-08-09T12:15:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:30:00.000Z</ns:EndTime>
            </ns:CapacityBlock>
         </ns:Adjustment>
         <ns:Adjustment>
            <ns:Cable>
               <ns:CableId>kabel-7890</ns:CableId>
            </ns:Cable>
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Amp">64</ns:Capacity>
               <ns:StartTime>2013-08-09T12:15:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:30:00.000Z</ns:EndTime>
            </ns:CapacityBlock>
         </ns:Adjustment>
      </ns:RequestAdjustedCapacityRequest>
   </soap:Body>
</soap:Envelope>
```

Draft version

4.4 RequestAdjustedCapacityResponseMessage

See 4.3. Messages are almost identical.

4.5 UpdateAggregatedUsage

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://OSCP/DSO/2013/06/">
   <soap:Header>
      <ns:Priority>1</ns:Priority>
      <ns:Identification>
         <ns:DSO>8712423014022</ns:DSO>
         <ns:CSP>8712423222120</ns:CSP>
         <ns:DateTime>2013-08-09T00:00:00.000Z</ns:DateTime>
         <ns:EventId>uuid</ns:EventId>
      </ns:Identification>
   </soap:Header>
   <soap:Body>
      <ns:UpdateAggregatedUsageRequest>
         <!--1 or more repetitions:-->
         <ns:Usage>
            <ns:Cable>
               <ns:CableId>kabel-1234</ns:CableId>
            </ns:Cable>
            <!--Zero or more repetitions:-->
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Wh">32</ns:Capacity>
               <ns:StartTime>2013-08-09T12:00:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:15:00.000Z</ns:EndTime>
            </ns:CapacityBlock>
            <ns:CapacityBlock>
               <ns:Capacity UnitOfMeasure="Wh">100</ns:Capacity>
               <ns:StartTime>2013-08-09T12:15:00.000Z</ns:StartTime>
               <ns:EndTime>2013-08-09T12:30:00.000Z</ns:EndTime>
            </ns:CapacityBlock>

         …

         </ns:Usage>

         …

      </ns:UpdateAggregatedUsageRequest>
   </soap:Body>
</soap:Envelope>
```

4.6 MeterValues.req

<to be added later>

Draft Version

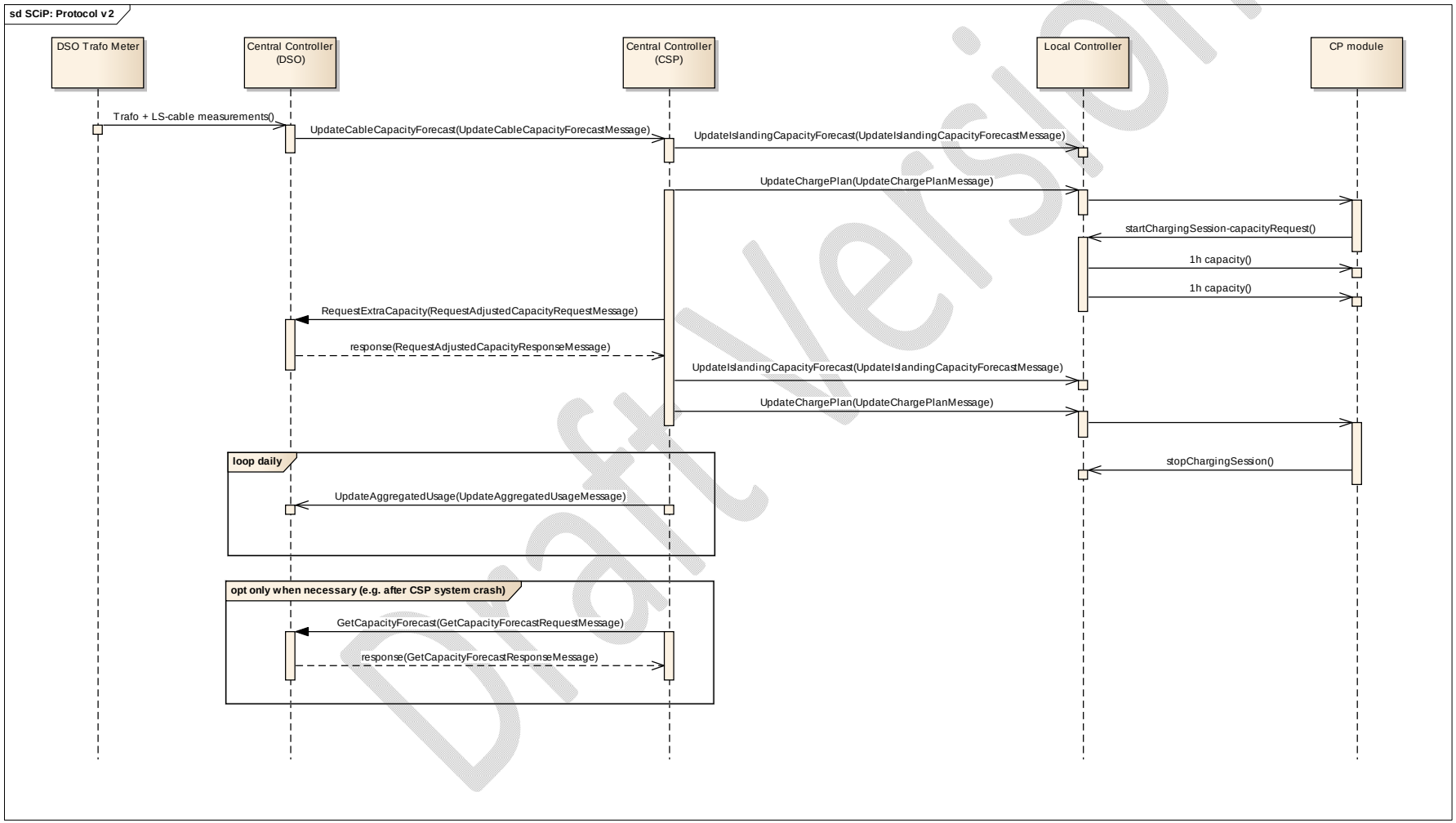
5. Appendix 1: open points

Wat?	Actiehouder	Datum	Status
Is it ok for GreenFlux to use webservices for the communication between CSP en DSO?? Answer: yes	Hans de Boer	6-6-2013	Closed
What is an easy way to aggregate?	All	11-6-2013	OPEN

Draft Version

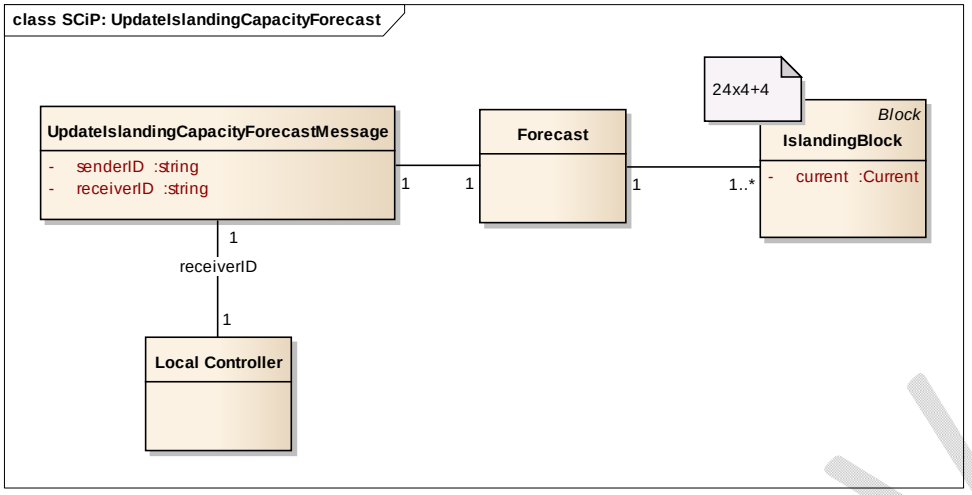
6. Appendix 2: Incorporate OCPP 2.0 in original design

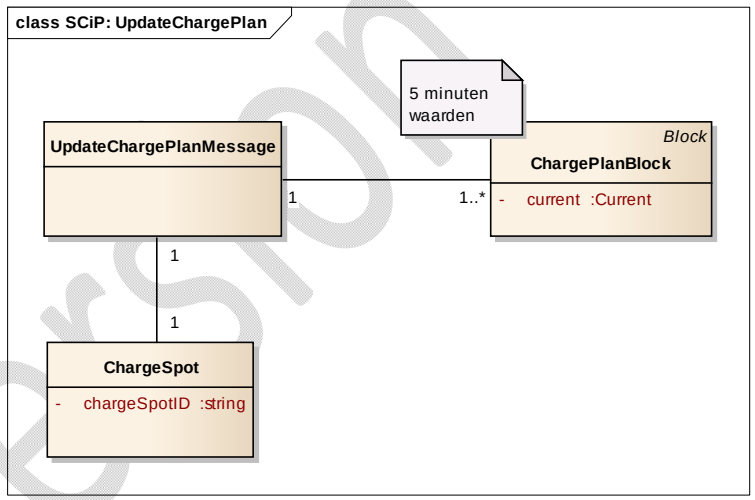
Original design (as a reference):



Draft Version

The message UpdateIslandingCapacity used to look like the figure on the left.





Since the LC functions as a (sometimes intelligent) in this case, this information can also be sent using the message SetChargingProfile from the draft of the OCPP 2.0 protocol. The same holds for the original UpdateChargePlan message. This one can therefore also be replaced by the message SetChargingProfile.

Draft Version