7th Framework Programme

INFSO-ICT 285285

Specifications for the enhancement of existing smart metering protocols

Deliverable no.	4.2	Specifications for the enhancement of existing smart metering protocols	
Work package	4	Specifications	
Editor(s)	Ankou, Auguste (IFS) Schmidt, Robert (DNDE) Caldevilla, Andrés (DNDE) Andras Kovacs (Broadbit) Evangelia Portouli (ICCS)		
Status	Final		
Distribution	Confidential (CO)		
Issue date	2012-07-04	Creation date	2012-01-09
European Commission Information Society and Media	Project co-funded by the European Commission DG-Information Society and Media in the 7th Framework Programme	SEVENTH FRAMEWORK PROGRAMME	

TABLE OF CONTENTS

Figures

LIST OF ABBREVIATIONS

ABBREVIATION	DESCRIPTION
A-V2G	Automotive-side Vehicle to Grid
AMI	Advanced Metering Infrastructure
AMR	Automatic Meter Reading
BMS	Battery Management System
CS	Charging Spot
DSO	Distribution System Operator
ECU	Electronic Control Unit
EEM	Electric Energy Meter
EV	Electric Vehicle
EVCC	Electric Vehicle Communication Controller
EVSE	Electric Vehicle Supply Equipment
GW	Gateway
HMI	Human Machine Interface
I-V2G	Infrastructure-side Vehicle to Grid
LAN	Local Area Network
LBC	Load Balancing Controller
OEM	Original Equipment Manufacturer
OSI	Open Systems Interconnection
PLC	Power Line Communication
PWM	Pulse Width Modulation
SECC	Supply Equipment Communication Controller
SoC	State of Charge
V2G	Vehicle to Grid
WP	Work Package

REVISION CHART AND HISTORY LOG

REV	SVN-REV	DATE	REASON
0.1	179	2012-01-09	Initial structure
0.2		2012-03-13	First draft
0.3		2012-04-27	Rework and Integration of Broadbit contribution.
0.4		2012-06-12	integration of ICCS contribution
0.5		2012-06-21	Final draft
0.6		2012-07-03	Integration of comments after review

EXECUTIVESUMMARY

This deliverable reviews existing smart metering protocols and performs a gap analysis, according to the requirements as defined in D4.1. For the two charging scenarios, private and public, extensions are specified. The two situations are addressed with specific processes. These processes and their involvement on the protocol and data carried out by the protocol,are the scope of this document.

Itconsiders the smart metering protocols and comes to the conclusion that they can be used for V2G application as they are, by defining additional objects needed by V2G application.

This document is made of 3 parts: the first one related to the EVSE and its environment with the LBC and the EV, and the role of the smart meter. After that, the process of the vehicle charging is considered with data related to each phase.

The next part is related to smart metering protocol in term of services. The conclusion reached is that the needs of the communication between the LBC and theEVSE can be supported by the services currently available in DLMS/COSEM protocol services.

The last part concerns the objects requiredto be carried out during the communication between the LBC and the EVSE. Most of them are already available in DLMS/COSEM,some of them not. Each time a gap is identified, solutions are proposed. These solutions are in line with the protocols as they are defined for Smart metering.

1 SCOPE

The scope of this document is to make use of smart metring protocol in the context of EV, by defining the necessary topics which are currently not addressed in these protocols so that they can match all the identified requirements in the EV domain concerning the communication with the Electricity meter.

Furthermore, functionalities which are the scope of the SECC are highlighted here according to the IEC 15118-2.

2 REFERENCED DOCUMENTS

1	IEC 62056-46 Ed 1.1:2007	Electricity Metering – Data exchange for meter reading, tariff and load control - Part 46: Data link layer using, HDLC protocol
2	IEC 62056-47 Ed 1.0:2006	Electricity Metering – Data exchange for meter reading, tariff and load control - Part 47: COSEM Transport layer
3	IEC 62056-5-3 Ed 2.0:2006	Electricity Metering – Data exchange for meter reading, tariff and load control - Part 53: COSEM Application layer
4	IEC 62056-6-1	Electricity Metering – Data exchange for meter reading, tariff and load control - Part 61 Object identification System (OBIS)
5	IEC 62056-6-2	Electricity Metering – Data exchange for meter reading, tariff and load control - Part 62. Interface classes
6	IEC 61334-6 Ed 1.0 2000	Distribution automation using distribution line carrier systems – Part 6 A-XDR encoding rule
6	ISO CD 15118-1	Road vehicles- Vehicles to Grid Communication Interface-Part 1: Technical protocol description and Open System Interconnections (OSI)layer requirements
7	ISO CD 15118-2	Road vehicles- Vehicles to Grid Communication Interface-Part 2: Technical protocol description and Open System Interconnections (OSI)layer requirements
8	ISO CD 15118-3	
9	Power up D3.1	PowerUp Preliminary V2G Architecture
10	Power up D4.1	V2G Interface specifications between the electric vehicle, the local smart meter, and ITS service providers.
11	ERDF G3 PLC Profile Specification	ERDF G3 PLC Profile Specification

3 INTRODUCTION

An analysis of available protocols used in Smart Metering domain, according to some criteria predefined by the European standardization organization has led to the selection of DLMS/COSEM as the protocol to be used in Smart metering.

DLMS/COSEM protocol can be used over several media, by defining the proper lower layers matching the concerned purpose, so that it can be used on distribution line, Ethernet or wireless communication…etc. The main point is that all these defined protocol stacks share the same application layer and object model so that data can be exchanged in a consistent way, using the proper bridge.

One of the approaches followed within the scope of PowerUp project has identified the DLMS/COSEM as the protocol to be used between the Grid and the EVSE, and IEC 15118-2, the one to use for the communication between the EVCC and the EV, as it is shown in the figure below:

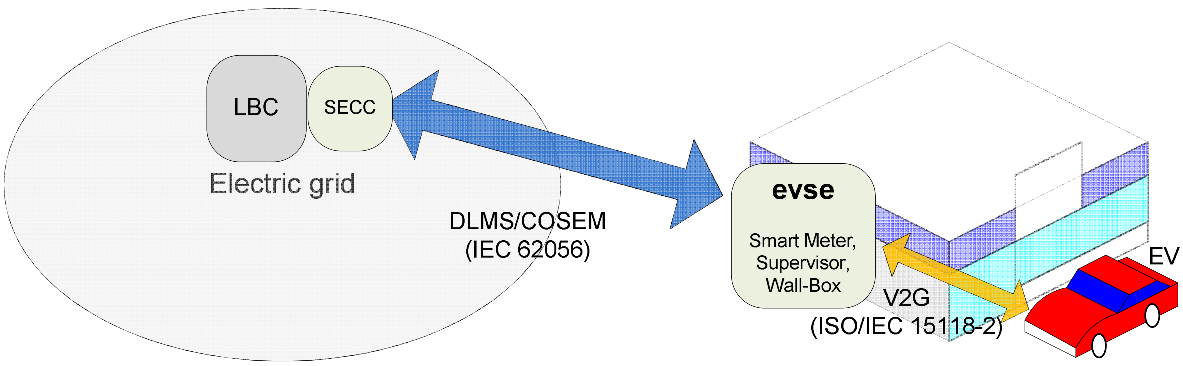


Figure 1 - Electric vehicle charging architecture

Internally, inside the Grid and the EVSE, the communication takes place between the Load balancing Controller (LBC) which is responsible for the grid constraints management, request prioritization and charging sessions scheduling, and at the EVSE side, the Electricity meter, which has to be considered at the Grid point of view as the metering Point of Delivery where ultimately, the supply is provided.

The DLMS/COSEM protocol is used between the LBC and the Electricity meter. In this configuration, the LBC always plays the role of the client when the Electricity meter plays the role of the server.

3.1.1. Meter situation

It is important to note that the Electricity meter is embedded in the Supply equipment and becomes part of this device. It does not directly communicate with the EV, but is able to get any information coming from the EV through the proper interface in the EVSE.

The way the data has to be exchanged internally in the EVSEneeds to be defined. It may also use COSEM protocol. It is worth noticing that the communication with the EV uses the ISO/IEC 15118 protocol only.

4 THEEVSE

The structure of the EVSEis made of at least 3 components:

1- The smart meter
2- The SECC
3- The Wallbox

4.1. Role of the Smart meter

The smart meter is responsible when authorized, to allow the energy transit between the LBC and the EV at the energy Point of Delivery.

At the beginning of the charging sequence there is an agreement between the EV and the LBC regarding the quantity of energy to provide and the duration of the charging (quality of service).

This agreement takes into account the EV identification, the contract identification… etc verification, as performed by the SECC. The duration of charging has to take into account the different rates of the energy during the time interval the charging has to take place, the demand level and the grid constraints.

During this energy flow the smart meter monitors the demand according to the profile defined during the setup of the charging session, and takes the proper action accordingly.

At the end of the charging session, the smart meter elaborates the billing data related to the charging sequence. The smart meter must have prepayment functionalities. In such a case, the charging authorization and billing data provided at the end of the charging session take into account the credit available for the concerned client.

4.2. Role of the SECC

The SECCcan be embedded in the Smart meter or not. Its role is to manage all the process not directly linked to energy measurement and delivery, as requested by the Electric Vehicle charging, mainly process related to:

- Electric vehicle identification,
- Contract identification with the utility or the charging station company,

- The payment mode, references and parameters

In addition, the SECChas the responsibility of the monitoring of the EV during all the charging lifetime, meaning the control of the

- Line lock process,
- Power delivery
- Welding detection
- Metering status

And any other action to be enterprise according to their occurrence.

4.3. Role of the Wall-box

The role of the Wall-box is the monitoring of safety issues. As the Wall-box uses for its communication with the EV the Pilot line and not the same communication protocol than the SECC, the Wall-box will not the scope of this document.

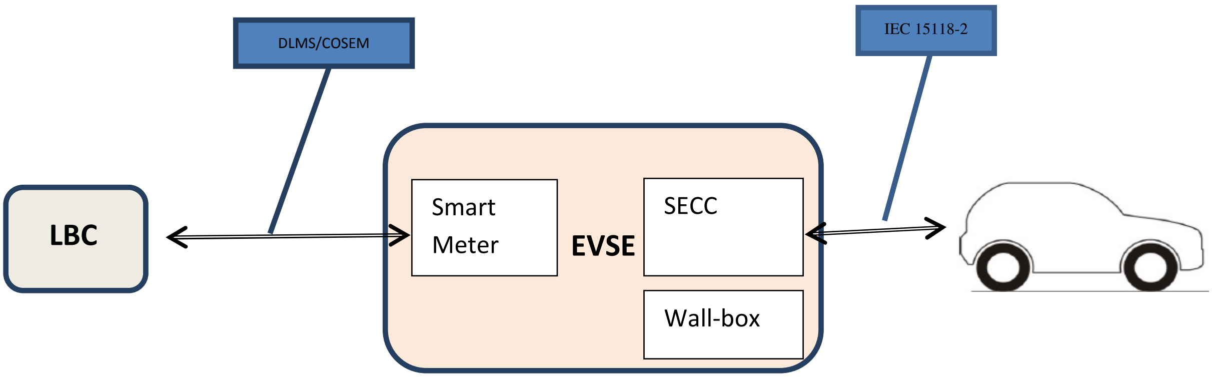


Figure 2- EVSE architecture

The previous figure presents the internal and external entities participating to the charging process. From a practical device point of view, the SECC may be embedded in the smart meter or it may be a distinct device.

All the communications between the LBC and the EVSEare performed using the DLMS/COSEM protocol. The relevant interface in the EVSEfor this communication is the Smart meter.

All the communications between the EVSEand the EVCC are performed using the ISO 15118-2 protocol. The relevant interface for this communication is the SECC.

5 PROCESS

5.1. Connection establishment phase

The SECC is able to handle the EV recharge process and has locally stored information about the LBC (IPv6 address, applied transport protocol etc.).

The first step is that the plug is connected to the EV. When the link is established, the EV begins the search for EVSE. In this case the role of EVSE (from the EV side is the smart meter).

The EV connects to the smart meter and sends the supported app. request.

The SMART METER receives the *support app protocol request*, which contains the session id, the supported protocol versions identified by schema IDs and their priority.

The smart meter stores these parameters in the V2G transaction object (V2G request). *When the SMART METER receives the support app protocol request in the V2G transactions object and stores the session id parameter in a data object called* ***identification object****.*

When the former stepis successful then the SMART METER sends the data in V2G transactions object (V2G request)to the LBC.

When the LBC receives this data, it opens an application association and uses DLMS/COSEM service forsending the V2G response (sent by LBC) to the V2G transaction data object (V2G response). This encapsulated message contains the V2G *supported application protocol response*. After receiving the message the SMART METER sends the *V2G supported app. response* to EV (using V2G protocol) containing the schema ID.

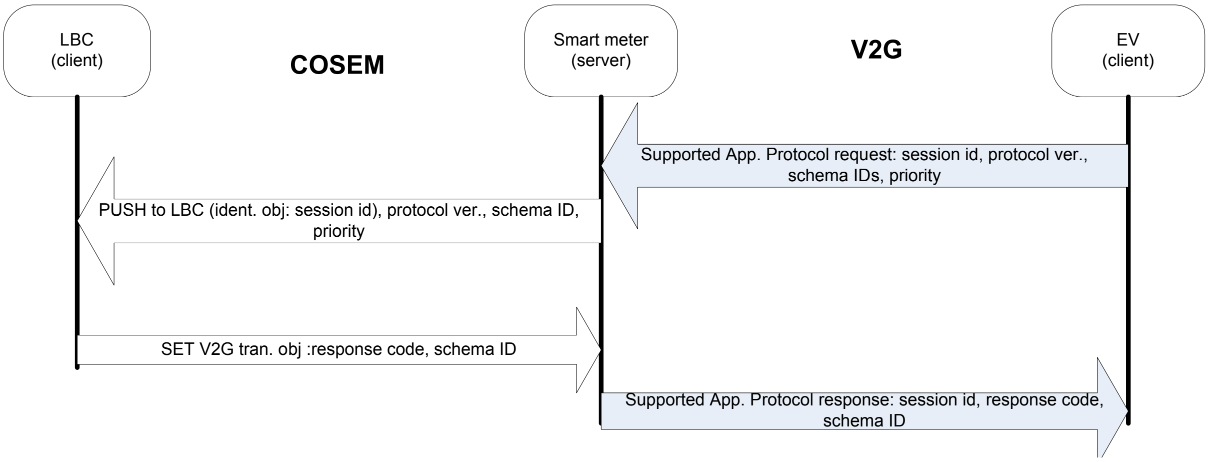


Figure 3 – MSC of the connection Establishment Phase

5.2. Session set up

If the *supported app. protocol responseis* received at EV level, the EV sends the *session setup request*. This V2G message contains the EVCC_ID. The SMART METER receives the session setup request and stores the EVCC_ID in the context management entity. After that the SMART METER transfers the data from V2G transaction object to LBC. After receiving the message, the SMART METER sends the *V2G session setup response* to EV containing the EVSE_ID, a timestamp and response code (OK). In the response message (that the smart meter sends to EV), only the EVSE_ID comes from the LBC, other parameters (response code, timestamp) are generated by the smart meter locally.

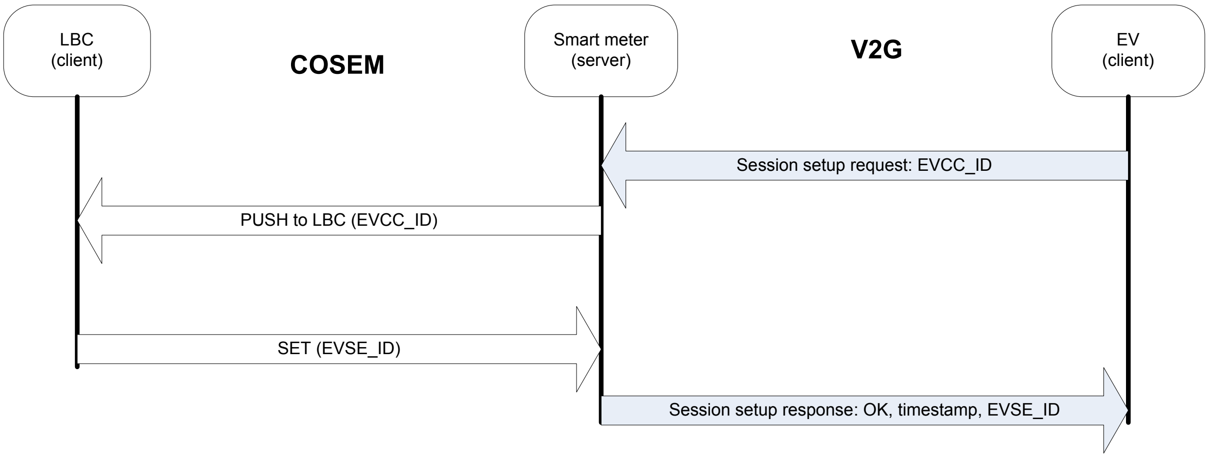


Figure 4 – MSC of the session set up phase

Objects identified at this phase are all identification objects.

5.3. Service and payment selection phase

During this sequence, the EV sends the service discovery request to SMART METER. This request contains the service scope string and the service category (AC 3 phase recharge).
The SMART METER has previously defined objectsallowing service and Payment management. Theseobjects have static values and contain the services the SMART METER can handle (for example EV recharge) and the payment types (contract, prepaid, etc.).
When the service discovery request message is received from EV, the SMART METER sends a V2G the response to the EV. This response contains all the payment option available in the Smart meter for the current transaction.
The payment selection phase can be done between the Smart meter and the EV without the contribution of the EV.

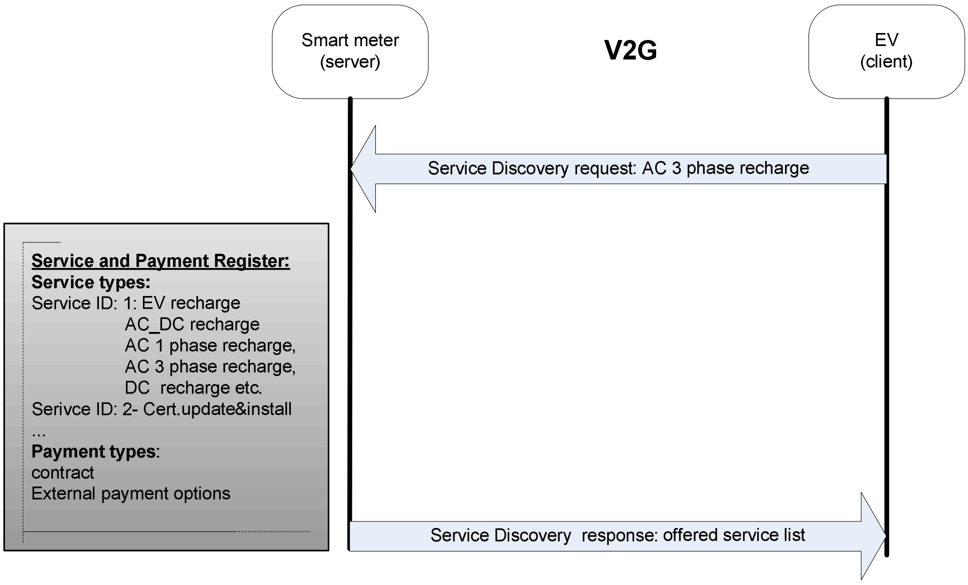


Figure 5 – MSC of the Service discovery phase

The EV chooses the EV recharge service and contract for payment type and sends the service and payment selection V2G request to SMART METER. The SMART METER after receiving this V2G request reads the Service&Payment parameters. If the chosen service and payment types are correct, the SMART METER sends the V2G response message with response code (OK). Otherwise the meter sends a V2G response containing response code error: FAILED_ServiceSelectionInvalid or FAILED_PaymentSelectionInvalid).

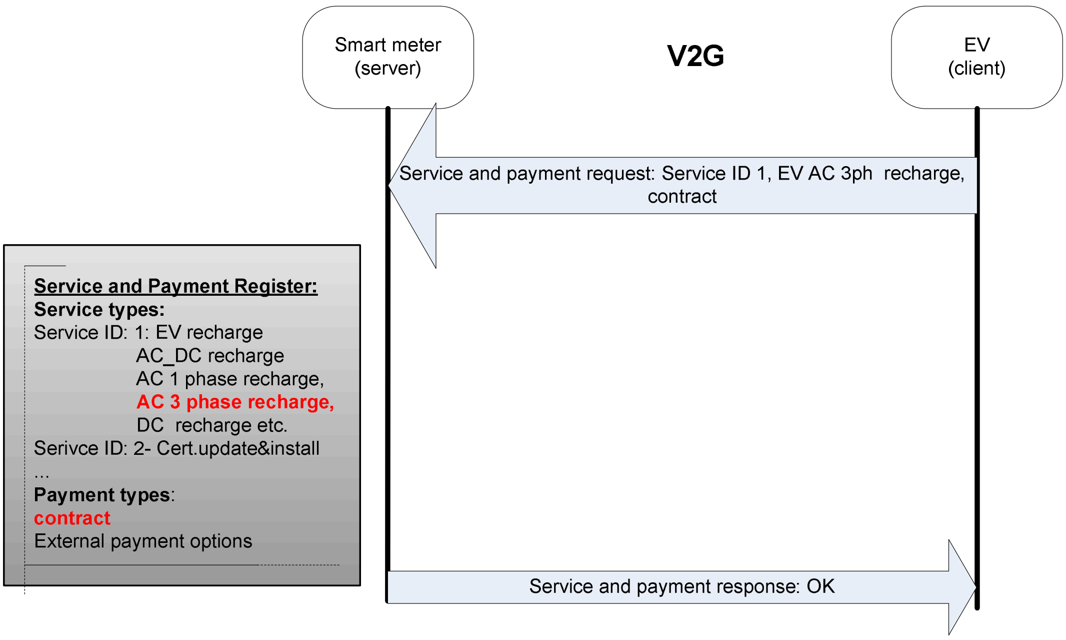


Figure 6 – MSC of the Payment selection Phase

5.4. Payment detail and Authentication phase

If the chosen payment type is contract, the EV sends a payment details request (with contract id and Contract Signature Cert. Chain) to the SMART METER. The SMART METER sends these data to the LBCwhich responds with process to the challenge. After this, the SMART METER sends the power delivery response message to the EV containing the response code (OK), a timestamp from the SMART METER and the Genchallenge.

After the EV receives payment details response message with response code (OK) sends the Contract Authentication Request to SMART METER with the Genchallenge from the previously message and the ChallengeSignature. The SMART METER sends this message to LBC. The LBC authenticates the user. If the authentication was successful the entity sends a positive response (OK), and the SMART METER forwards the message to EV in Contract Authentication with response code OK.

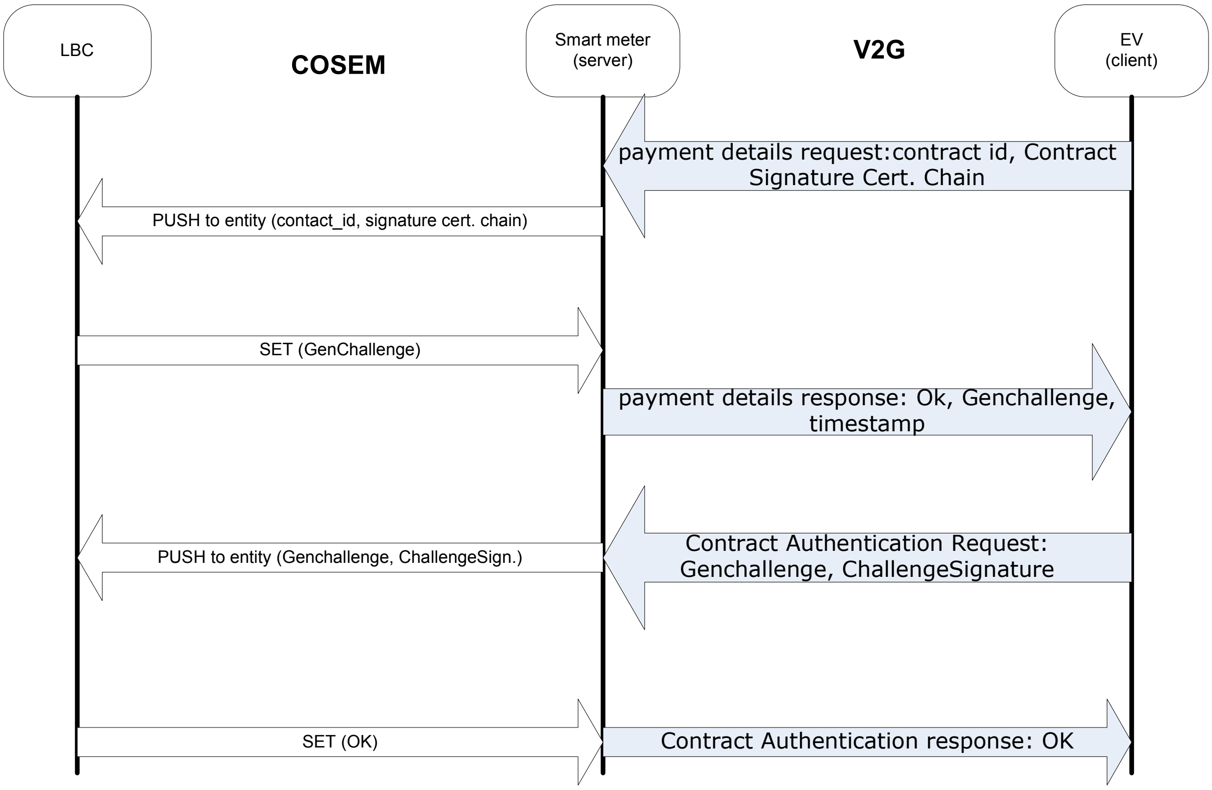


Figure 7 – MSC Payment detail and Authentication

5.5. Charge setup phase

Knowing that the authentication was successful, the EV sends the charge parameter discovery message to SMART METER. This message contains the requested energy transfer type (AC 1 phase, 3 phase or DC), the requested amount energy, the departure time, the priority and some EV limitation parameters during the charge (maximum voltage and current level, minimum current level).

The SMART METER stores these data. After this, the SMART METER sends the Energy amount, Maximum current, priority and the departure time data to LBC.
The LBC uses SET to send the schedule plan(s) for the given EV. The SMART METER stores the schedule plan(s) in an object called ***charging profile object***. The schedule plan(s) contains time and power pairs. Every schedule plan is identified with SAScheduleTuple ID number.
The SMART METER sends the charge parameter discovery response message to EV containing the following parameters:

- response code (OK),
- schedule plan(s) for EV.
- tariff information for the schedule plan(s)
- the local limitations for current and voltage level and the EVSE ready indicator (EVSE ready= true).

The EV received the charge parameter discovery response message with response code OK, schedule plan(s), tariff information and local limit (maximum, minimum current, voltage) parameters.

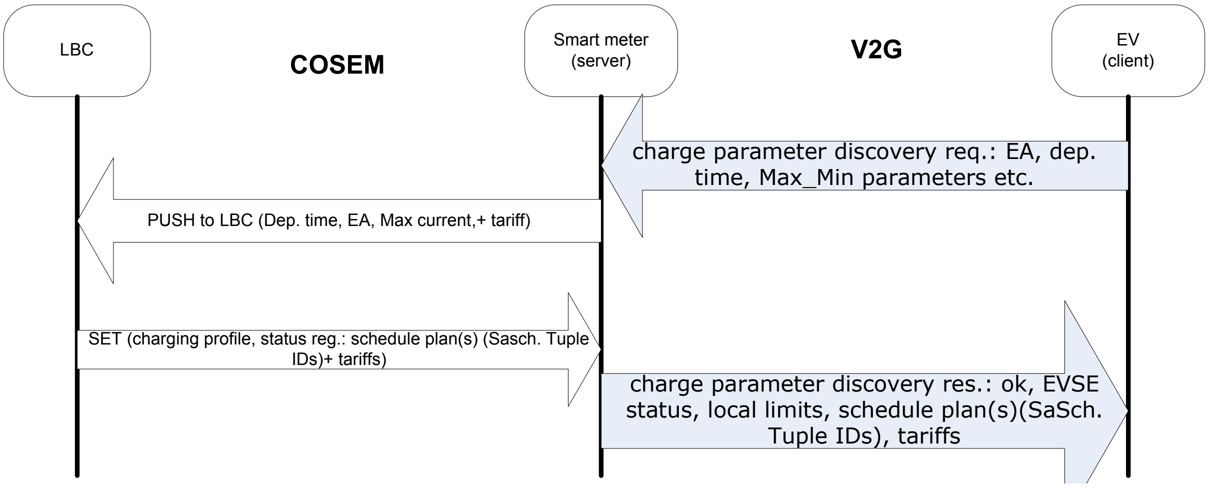


Figure 8 – Charge Set up Phase

5.6. Metering phase

If the charge parameter discovery response has a positive (OK) response code then the EV sends the power delivery request which contains the ready to charge state (true), the chosen schedule plan (identified by SAscheduleTuple ID) and the charging profile.
The SMART METER compares the charging profile received from the powerdelivery message and the stored data (the stored schedule plan with Saschedule Tuple ID=1 from LBC). If there are the same, the SMART METER sends a power delivery response with the EVSE status (from status object) and a response code (OK).

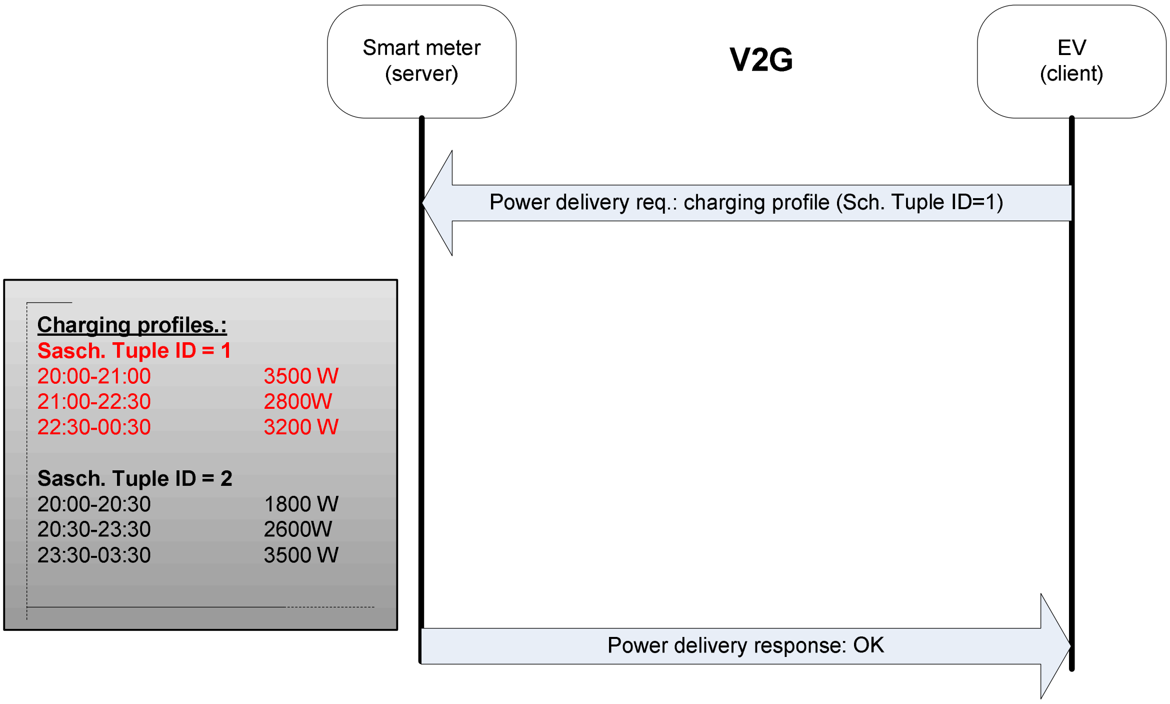


Figure 9 – Metering Phase

At the metering loop the EV sends the charging status request to SMART METER. The SMART METER reply message contains the actual meter status, some additional information about the meter, receipt requirement and the EVSE status form (status obj.).

Then the EV sends a request for the receipt with the data represented on the figure. The meter receives the request and sends a response message with response code OK.

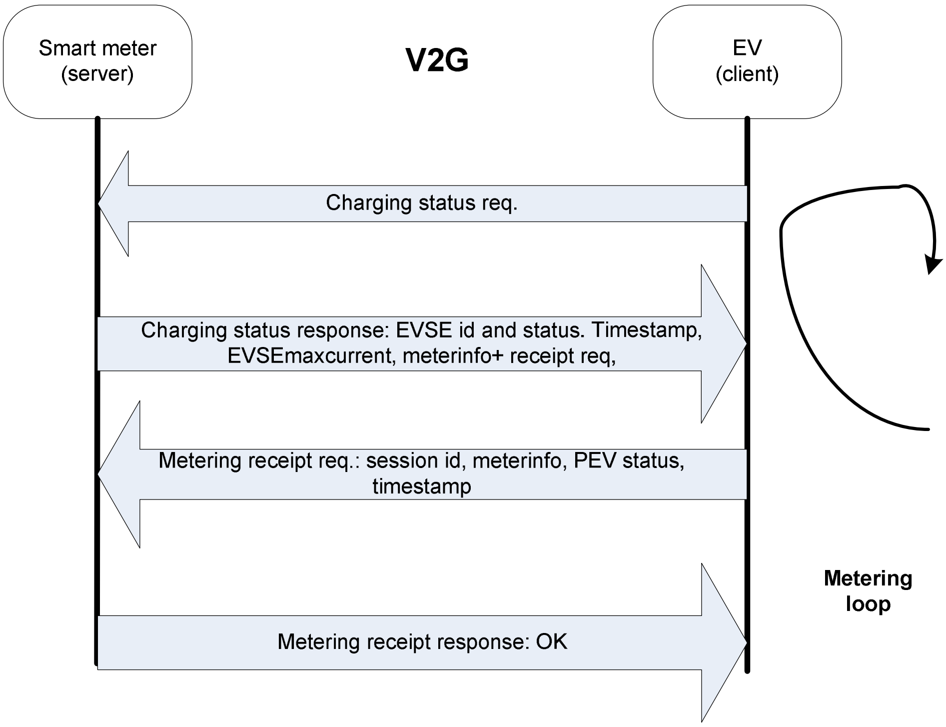


Figure 10 – MSC Charging Loop

When the EV finishes the charge,it sends a session stop message to the smart meter. The smart meter replies with a session stop response containing response code (Ok).

The smart meter can also end the charge process by setting the EVSE status ShutDownTimeStopCharging from false to true. This indicates the EV to stop the charge process.

5.7. Charge modification phase

If needed, for example due to the connection of another EV with more imminent needs or if the user requests a change in the schedule, the LBC may modify the charging profile of already connected EVs.
For each already connected vehicle, already charging, the SMART METER sends the Energy amount, Maximum current, priority, the departure time and the actual meter status data to LBC.
The LBC uses SET to send the modified schedule plan(s) for a specific EV, if this is required.
Below is a sum up of all the charging session, actor involved and message exchanged:

EV ↔ EVSE	EVSE ↔ LBC	EVSE ↔ Auth.&billing entity

	Reg/Res	Information (V2G)	Req/Res	Information (COSEM)	Req/Res	Information (COSEM)	
Session Setup	→ Req	EVCCID	→ Req	EVCCID	→ Req		seq Communication Setup
	← Res	ResponseCode: OK/FAILED EVSEID DateTimeNow	← Res	EVSEID	← Res		
Service Discovery	→ Req	ServiceScope ServiceCategory	→ Req		→ Req		seq Identification, Authentication and Authorization
	← Res	ResponseCode: OK/FAILED PaymentOptions ChargeService ServiceList	← Res		← Res		
ServiceDetail (opt)	→ Req	ServiceID	→ Req		→ Req		
	← Res	ResponseCode: OK/FAILED ServiceID ServiceParameterList	← Res		← Res		
Service and Payment Selection	→ Req	SelectedPaymentOption SelectedServiceList	→ Req		→ Req		
	← Res	ResponseCode: OK/FAILED	← Res		← Res		
Certificate Installation (opt)	→ Req	Id OEMProvisioningCert ListofRootCertificateIDs	→ Req		→ Req		
	← Res	Id ResponseCode: OK/FAILED ContractSgnatureCertChain ContractSignatureEncryptedPrivateKey ContractEncryptionEncryptedPrivateKey DHParams ContractID	← Res		← Res		
Certificate Update (opt)	→ Req	Id ContractSignatureCertChain ContractID ListOfRootCertificateIDs	→ Req		→ Req		
	← Res	Id ResponseCode: OK/FAILED ContractSignatureCertChain ContractSignatureEncryptedPrivateKey ContractEncryptionEncryptedPrivateKey DHParams ContractID RetryCounter	← Res		← Res		
Payment details	→ Req	ContractID ContractSignatureCertChain	→ Req		→ Req	ContractID ContractSignatureCertChain	
	← Res	ResponseCode: OK/FAILED GenChallenge DateTimeNow	← Res		← Res	GenChallenge	
Contract Authenticati	→ Req	Id GenChallenge	→ Req		→ Req	GenChallenge	

on	← Res	ResponseCode: OK/FAILED	← Res		← Res	ResponseCode: OK/FAILED	
Charge Parameter Discovery	→ Req	EVRequestedEnergyTransfer - AC_single_phase_core - AC_three_phase_core - DC_core - DC_extended - DC_combo_core - DC_unique AC_EVChargeParameter - DepartureTime - EAmount - EVMaxVoltage - EVMaxCurrent - EVMinCurrent DC_EVChargeParameter - DC_EVStatus - EVMaximumCurrentLimit - EVMaximumPowerLimit - EVMaximumVoltageLimit - EVEnergyCapacity - EVEnergyRequest - FullSOC - BulkSOC	→ Req	AC_EVChargeParameter - DepartureTime - EAmount - EVMaxCurrent **+ SalesTariff**	→ Req		seq Target setting and charge scheduling

	← Res	ResponseCode: OK/FAILED SAScheduleList - SAScheduleTupleID - PMaxSchedule + PMaxScheduleID + PMaxScheduleEntry - SalesTariff + SalesTariffID + SalesTariffDescription + NumEPriceLevels + SalesTariffEntry AC_EVSEChargeParameter - AC_EVSEStatus - EVSEMaxVoltage - EVSEMaxCurrent - EVSEMinCurrent DC_EVSEChargeParameter - DC_EVSEStatus - EVSEMaximumCurrentLimit - EVSEMaximumPowerLimit - EVSEMaximumVoltageLimit - EVSEMinimumCurrentLimit - EVSEMinimumVoltageLimit - EVSECurrentRegulationTolerance - EVSEPeakCurrentRipple - EVSEEnergyToBeDelivered	← Res	SAScheduleList - SAScheduleTupleID - PMaxSchedule + PMaxScheduleID + PMaxScheduleEntry - SalesTariff + SalesTariffID + SalesTariffDescription + NumEPriceLevels + SalesTariffEntry	← Res		
Power Delivery	→ Req	ReadyToChargeState ChargingProfile - SAScheduleTupleID - ProfileEntry + ChargingProfileEntryStart + ChargingProfileEntryMaxPower DC_EVPowerDeliveryParameter - DC_EVStatus - BulkChargingComplete - ChargingComplete	→ Req		→ Req		

	← Res	ResponseCode: OK/FAILED AC_EVSEStatus - PowerSwitchClosed - RCD - ShutDownTime - StopCharging DC_EVSEStatus - EVSEIsolationStatus - DC_EVSEStatusCode	← Res		← Res		
Charging Status	→ Req		→ Req		→ Req		seq Charge control and Re-scheduling (AC-Messages)
	← Res	ResponseCode: OK/FAILED EVSEID SAScheduleTupleID EVSEMaxCurrent MeterInfo ReceiptRequired AC_EVSEStatus	← Res		← Res		
Metering Receipt (opt)	→ Req	Id SessionID SAScheduleTupleID MeterInfo	→ Req		→ Req		
	← Res	ResponseCode: OK/FAILED AC_EVSEStatus	← Res		← Res		
Session Stop	→ Req		→ Req		→ Req		seq End of charging process
	← Res	ResponseCode: OK/FAILED	← Res		← Res		
Cable Check	→ Req	DC_EVStatus	→ Req		→ Req		(DC-Messages)
	← Res	ResponseCode: OK/FAILED DC_EVSEStatus	← Res		← Res		
Pre Charging	→ Req	DC_EVStatus EVTargetVoltage EVTargetCurrent	→ Req		→ Req		
	← Res	ResponseCode: OK/FAILED DC_EVSEStatus EVSEPresentVoltage	← Res		← Res		
Current Demand	→ Req	DC_EVStatus EVTargetCurrent EVMaximumVoltageLimit EVMaximumCurrentLimit EVMaximumPowerLimit BulkChargingComplete ChargingComplete RemainingTimeToFullSoC RemainingTimeToBulkSoC EVTargetVoltage	→ Req		→ Req		
	← Res	ResponseCode: OK/FAILED DC_EVSEStatus EVSEPresentVoltage EVSEPresentCurrent EVSECurrentLimitAchieved EVSEVoltageLimitAchieved EVSEPowerLimitAchieved EVSEMaximumVoltageLimit EVSEMaximumCurrentLimit EVSEMaximumPowerLimit	← Res		← Res		
Welding	→ Req	DC_EVStatus	→ Req		→ Req		

Detection	← Res	ResponseCode: OK/FAILED DC_EVSEStatus EVSEPresentVoltage	← Res		← Res		

6 RELATED METERING PROTOCOLS –DLMS/COSEM

6.1. General

DLMS/COSEM protocol is used between the EVSEand the LBC. The following section describes the application of DLMS/ COSEM protocol to the EV charging and then identifies related objects necessary for that.

Three steps are considered in the use of DLMS/COSEM in V2G environment:

- Use of Lower layers
- Use of the DLMS/COSEM Application layer
- Metering device modeling

All these 3 steps are considered below.

6.2. Lower layers

All the lower layers which can be used (Wireless, G3, Broadband PLC or any other…) are considered as IP compatible. Therefore the lower layers will not be part of the scope of this document. The main point is that at the top of these lower layers, the interface presented is either TCP, either UDP or SCTP.

All these transport layers can correctly interface the DLMS/COSEM wrapper in order to provide the relevant parameters need by DLMS/COSEM application layer at its service access point.

6.3. Application layer

DLMS/COSEM is a connection oriented protocol meaning that before any data exchange takes place, an application association must be established between the client and the server. This application association is important because providing the identification, authentication and correct credentials related to the client and the server which want to participate to the application association. In the context of V2G this application association will be used to verify the authentication of the EV driver and owner authentication in regard to their subscription in the grid or other is not.

A given amount of tools are already available currently in DLMS/COSEM for this purpose. Some other one, related to asymmetric algorithm management will be added very soon. V2G requirements have to be checked in regard to these available tools if ever they are sufficient or not, and if ever there is a need to have an additional one for addressing V2G needs.

6.4. xDLMS services

The xDLMS services to be used are not only those currently available in the Green book, but also those under definition meaning,

- GET,
- SET,
- ACTION
- EVENT NOTIFICATION,
- DATA NOTIFICATION,
- ACCESS SERVICE

The GET service allows the data retrieving from the metering device. This GET service includes also GET with list and GET with Data block management.

The SET service allows data programming inside the metering device. This SET service includes also SET with list and SET with Data block management.

The ACTION service allows method execution inside the metering device. This ACTION service includes also ACTION with list and ACTION with Data block management.

EVENT NOTIFICATION service is a non-client/server service. It allows a metering event reporting up to the LBC. This service is always used as an unconfirmed service.

DATA NOTIFICATION service is also a non-client/server service allowing information reporting to the LBC.

ACCESS service allows mixing GET, SET, ACTION and EVENT NOTIFICATION service allowing performing in one shoot, all the operations required.

These six services are complete enough so that there is no need to add any additional service for V2G specific purposes. Any use case process will make use of one of these 6 services unless during some application a gap is accidently identified.

6.5. Encoding

The encoding used is ISO CD 15118 is XML/EXI not available where in DLMS/COSEM the a-xdr encoding is used. Communication between the LBC and the metering device will use a-xdr encoding rules. A one to one relationship, based on the ASN1 can be made between XML/EXI encoding and a-xdr encoding.

7 OBJECT MODELING

7.1. General

The objective of the present section is to model the meter meaning the information which can be exchanged from the meter to the LBC, and information flowing in the other direction.

Internal information regarding the meter or the LBC does not have any interest in the data modeling. The reference document for EV meter modeling is the ISO CD 15118, from which the following data can be extracted:

7.2. Identification

This information is provided by the EVCC to the meter. It is created at the beginning of a charging session, during the session set up and discovery phase, and identifies any charging transaction. It will be part of the payment or reading.

The information is retrieved by the SECC using IEC 15118-2 protocol and made available for the Smart meter via the SECC.

The LBC can access these information when it requests them from the smart meter.

Name	Class Id	Logical name	Data type	length
session_id	Data	1-b:95.2.0.255	octetstring	8
EV_id	Data	1-b:95.2.1.255	octetstring	32
evse_id	Data	1-b:95.2.2.255	octetstring	32
Service_id	Data	1-b:95.2.3.255	octetstring	8
Contract_id	Data	1-b:95.2.4.255	octetstring	128
Energy_provider_id	Data	1-b:95.2.5.255	octetstring	256
Identification	Profile	1-b:95.2.255.255	Array	6 elements

Each identification data can be accessed individually. The complete identification is accessible in a profile. In such a case, the capture object of the related profile has to precise the data currently captured inside it.

7.3. Contract identification

This parameter is need during the service and payment selection phase.

This data comes from the EVCC. It defines a unique contract identification number used for charge payment. It identifies a given subscription process regarding the Utility or the charging station and is used by the EVCC to identify the concerned contract with the EP.

The contract is identified by a contract Id

Name	Class Id	Logical name	Data type	length
contract_id	Data	1-b:95.2.4.255	octetstring	128bytes

7.4. Payment

This section will be held with the security extension of DLMS/COSEM for the use of the asymmetric key and certificate management and prepayment objects. See Annex A.

7.5. Charge parameters

This also is addressed by payment objectsdefined in DLMS/COSEM. See Annex A.

7.6. Charging profile

The charging profile provides tariffication information (price of the cost of the energy, depending on the instant the charging is performed). It is based on

- the duration of the charging required by the EVCC,
- the level of energy needed.
- The quality of service requested

The charging profile is the result of a negotiation between the LBC and the EV, and managed after that by the smart meter.

At the beginning the EVCC provide to the Smart meter, the quantity of energy need and the whished duration of the charging, then the smart meter send these parameters to the LBC.

Based on these parameters, the LBC calculates and proposes, depending on its capabilities, the type of contract subscribed by the vehicle user, several charging profiles which are depending on

- The level of demand need for complying to the requested charge,
- The tariff rate to apply during the different section of the day profile in application during the concerned charge and the estimated cost of the charge,

The vehicle user has to agree on one of these profiles or even deny all of them and repeat his request with other parameters. He can even modify the maximum power inside each section of day profile, so that the customer, for very high price can select low demand or significantly decrease it, in order the supply being cut off during high price rate day profile section.

As conclusion when the negotiation is concluded, then the effective duration of the charge will be calculated. All these information constitute the charging profile negotiated between the two parties.

In the metering point of view, price of each tariff is part of the activity calendar. This interface class is already available in DLMS/COSEM.

Maximum Demand request profile can be built from the Register monitor interface class.

Providing price information can be done with prepayment objects under definition.

At final the charging profile will look like the following structure

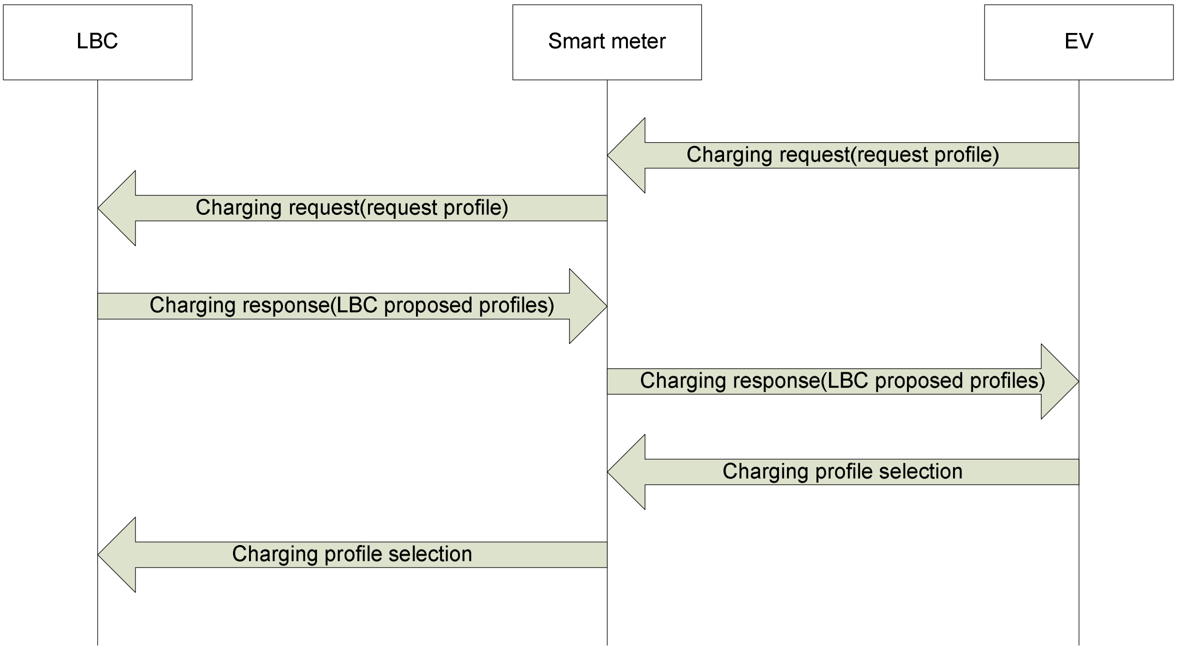
```
Charging_profile array  ofstructure

{
        Start_time      time
        Rate            rate_type
        Max demand      demand_type
}
```

For each rate, a maximum demand which must not overtake is set up.

Such a process can be performed using activity calendar, script and register monitor interface classes.

For DLMS/COSEM modeling point of view a data IC is used. It contains the same pattern than the daily profile action structure (attribute 5 of the activity calendar). This data interface class is directed to a register monitor who manages the Maximum demand as a threshold.



The register monitor array must contain the same element than the day profile.

Name	Class Id	Logical name	Data type
charge_profile	Data	1-b:95.2.0.255	Array ofstructure
Max_demand	Register monitor	1-b:16.0.40.255	Array unsigned32
Ev_script	script	0-b:10.0.110.255	Method
Activity calendar	Activity calendar	0-b:13.0.0.255	array
Special days table	Special days table	0-b:11.0.0.255	array
Tariffication script	dcript	0-0.10.0.106.255	Method

During the charging, the charging profile can be renegotiated. This needs an agreement of the 2 parties prior.

7.7. Tariff

The tariff is defined from the agreed charging profile. The parameters constituting the tariff table are the following:

Parameter name	type	Cosem object
Currency	Octetstring (3)	Accounting Control IC Att 11
tariff	Structure	Activity calendar
Tariff Id	Long unsigned	Activity calendar Attr 5 (script Id
Tariff description	octetstring	
EPrice unit	structure	Charge IC Attr2

EPriceMultiplier	structure	Charge IC Attr2
TariffEntries	structure	
TariffStart	octetstring	
TariffPmax	Array of unsigned 32	Register monitor
EPrice		Charge IC Attr2

All the parameters needed are already available or on the way to be included in the Blue book already.

7.8. Charging

Once the charging profile is accepted, the charging can start. It will be ruled by the parameters defined during the charging profile selection. During this phase, the smart meter gets knowledge of the EVCC and EV status and if ever everything is OK, process to the following actions:

1- Reading of all the energy register versus tariff and timestamp them

2- Reading of the start date of the charging

3- Allows the energy flow by closing the breaker.

Then periodically, the smart meter get knowledge of status of the EVCC and the EV. Data related to these statuses are below.

7.9. Line lock

The line lock request and response exchange is between the EVSE and the EV. This does not involve the SECC. However, the smart meter needs to be keep informed at the beginning of the charging and during the its whole lifetime in order to be in position to stop the energy flow in case of problem.

The only information the smart meter needs to know is the EVStatus which is provided by the EVSE to the smart meter. This last one does not need the SECC contribution for stopping the charging. However, the EVStatus is available inside the smart meter and by this way in the SECC in order to know the reason of the charging stopping.

EVSEStatus will be the status of the EVSE and the one of the EV.

Parameter name	type	Logical name	Cosem object
EVSE status	Bitstring	1-b:95.2.10.255	Extend register IC

Fatal_error	Boolean	Attr 2	B0
EVSE_standby	Boolean	Attr 2	B1
Connector_locked	Boolean	Attr 2	B2
Powerswitch_status	Boolean	Attr 2	B3
Rcd_type	Boolean	Attr 2	B4
Shutdown_time	octetstring	Attr 5	
Charger_standby	Boolean	Attr 2	B5
Evse_malfunction	Boolean	Attr 2	B6
Stop_charging	boolean	Attr 2	B7

Two objects are needed here. One which is a bitstring describing the status. It is a data IC

Another one which is a dynamic object also holding the date and time when the EVSE will turn off the power if the current power consumption level is not reduced by the EV. In case the EVSE detects the need for a shutdown, it defines this value by adding an offset in seconds to the current point in time. During normal operation the value of ShutDownTime is set to “0”.

Both can be set in an extended register IC. Then the shutdown time is the one of the Powerswitch status, in the case this one is TRUE. Otherwise it must have a default value (ex 1.1.2000 at 00:00);

7.10. Metering status

The data concerned are

Name	Class Id	Logical name	Data type
Evse_id	data	0-1:95.2.2.255	octetstring
evse_status	data	0-1:96.5.0.255	bitstring
current_date_time	Clock object	0-0:1.0.0.255	octetstring
maximum_power	Extended register	1-0.8.6.0.266	Unsigned32
current power	Extended register	1-0.8.24.0.255	Unsigned32
meter_ status	data	0-0:96.5.0.255	bitstring
status_profile	Profile	0-0:98.d.e.255	array

This object is a profile. All the data this profile is made of are already available. Just take them and build the profile.

7.11. Metering receipt

Data	Cosem object	LN	Class Id
EV_id	id	1-b:95.2.1.255	Data
evse_id	status	1-b:95.2.2.255	Data
session_id	Id	1-b:95.2.0.255	Data
meter_ status	status	0-0:96.5.0.255	Data
EV_status	status	0-2:96.5.0.255	Data
current_date_time	Clock object	0-0:1.0.0.255	Clock
tariff	See tariff		
receipt_profile	profile		profile

Here also, all the data this profile is made of are already available. Just take them and build the profile. The tariff will be excluded from the profile in order to constitute a specific data.

7.12. Power delivery

Data	Cosem object	Class Id	LN
EV_status	status	Data	0-2:96.5.0.255
switch_status	Disconnect control attribute 3	70	0-b:96.3.10.255
tariff	See tariff		
Charging profile	See charging profile	7	1-b:98.d.e.255

Here also, all the data this profile is made of are already available. Just take them and build the profile

7.13. Current demand

Data	Cosem object	Class Id	LN	Data type
EV_status	status	Data	0-2:96.5.0.255	bitstring
voltage	voltage	register	1-0.12.24.0.255	long-unsigned
current	current	register	1-0:11.24.0.255	long-unsigned
demand	demand	register	1-0:1.4.24.0.255	unsigned32

7.14. End of charging

The end of charging is materialized byproviding to the customer,

1- The amount of energy delivered per tariff rate,
2- The duration in each tariff rate,

3- Maximum demand reached in each tariff rate
4- And the cost.

If ever the payment mode is prepayment, the amount of available credit after chargfin is also provided;

Objects:

Data	Cosem objects	Class Id	LN	Data type
Metering receipt	profile	Profile IC	1-0:98.1.0.255	Array of structure
voltage	voltage	register	1-0.12.24.0.255	long-unsigned
current	current	register	1-0:11.24.0.255	long-unsigned
demand	demand	register	1-0:1.4.24.0.255	unsigned32

8 PRIVATE CHARGING METERING

The configurations described in the chapters above are mainly related to public charging. However they can also be applied to private charging with specific situations.

During private charging, the energy used for charging the EV is the one providing from the premises of the customer. The energy billing is therefore included in the global customer billing. In the LBC management point of view, it is the scope of the customer to correctly schedule the charging in order to comply with its subscribed energy profile, taking into account the other devices available in his premises. This is particularly the case when the customer owns several Electric vehicles which all need to be charged. The charging schedule must be defined by him.

8.1. EVSE

For security reasons the Wallbox must be available on the same way than for the public charging.

Although the energy used for the EV charging comes from the premises, a specific Smart meter for V2G purposes is needed. Additional functionalities for this meter are:

8.1.1. Follow up of the consumption of the EV

On the same way thanfor public charging, a bill is elaborated after each charging session. The same process can apply to the private charging. Therefore the customer can follow up day by day, the cost of its vehicle. Every time a charging session takes place, the amount of energy provided, max power, the duration of the charging and the time stamp are registered.

8.1.2. Load management

The load management is available in the Smart meter currently but this can be enhanced, allowing the schedule of several V2Gsessions.

It can also provide to the customer a dynamic way to schedule the charge depending on the power demand available in the premises.

8.1.3. Other EV charging

The objective of this last feature is to allow vehicles which are not owned by the owner of the premises to be charged, without the owner of the premises has to pay the consumed energy.

For this purpose, the process to follow is the same than for the public charging, meaning before the charging start,

The vehicle identification, the contract identification, the owner of the vehicle service, and the energy service provider identification must be entered in the smart meter. Then a connection with the service provider is performed, for the verification of the validity of the concerned contract.

If these verifications are successful, then the user of the vehicle enters the amount of energy to be provided and the expected duration of the charging.

When this step is successful, the charging can start and continue up to it completion. At the end of the charging, a bill is elaborated mentioning the amount of energy used. This bill has to be signed by the vehicle user, and sent to the vehicle owner energy service provider for payment elaboration.

Prepayment can also take place. In such a case,the electric vehicle owner must enter a token on which there must be a positive available credit. After the token identification and verification, the vehicle user can set the amount of energy expected. This amount of energy cannot exceedthe amount of energy allowed by the available credit.

9 LOAD BALANCING PARAMETERS AND ENHANCEMENTS

9.1. Load Balancing Controller

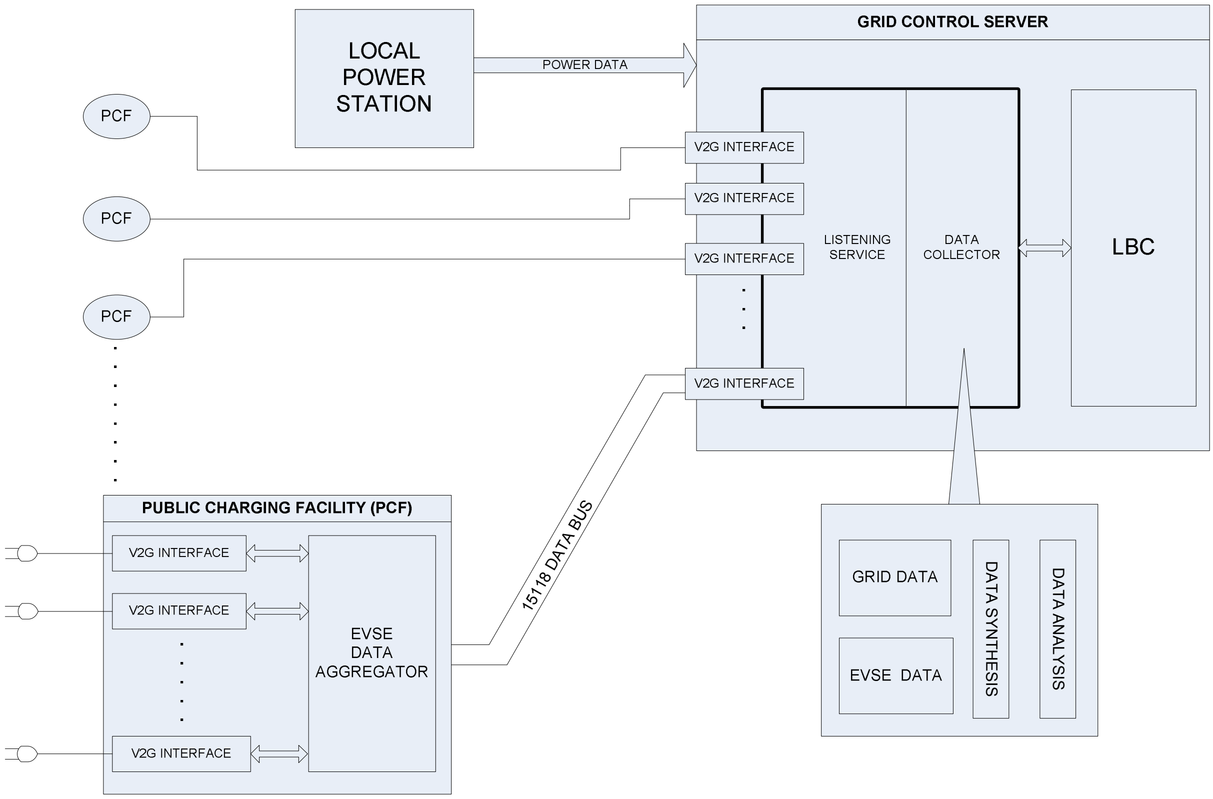
According to D3.1 (pp. 11-13), two different topologies are for consideration within PowerUp:

- The first one is based on V2G communication both in the local and end-to-end contexts, in accordance to the ISO/IEC 15118-2. This is selected as a simpler solution, since already defined messages may be employed even in different context, with the possible addition of elements according to the requirements deriving from the PowerUp use cases.
- The second topology is employing the V2G communication between the EV and EVSE, and the use of the DLMS/COSEM protocol, according to the IEC 62056 series, extended adequately for EVs. This topology is also relevant, since the DLMS/COSEM protocol is already widely used and favoured by the Utility Metering Industry, so the consortium believes that it will be the future market trend for smart meters.

This discussion is initiated with respect to the actual software components that will implement the specified services. Therefore the following topologies are serving the purpose of evaluating the best, most efficient, most effective technical software data exchanges.

8.1.1 LBC Topology 1

In this topology, only one session needs to be established between the EVSE and the EV. The EVSE Data Aggregator is used to transmit to the Grid Control Server (GCS) all the necessary data.



This topology is assuming the use of an EVSE Data Aggregator at the public charging facilities.

Components:

V2G Interface : Contains the necessary components for the communication between the EVSE and the EV. Data needs to be collected by the EVSE Data Aggregator via the established V2G Interfaces (e.g. openV2G).

EVSE Data Aggregator: Collects data from every EVSE in the facility and provides it to the GCS. It can provide data for each EVSE separately, or aggregate data for groups of EVSEs.

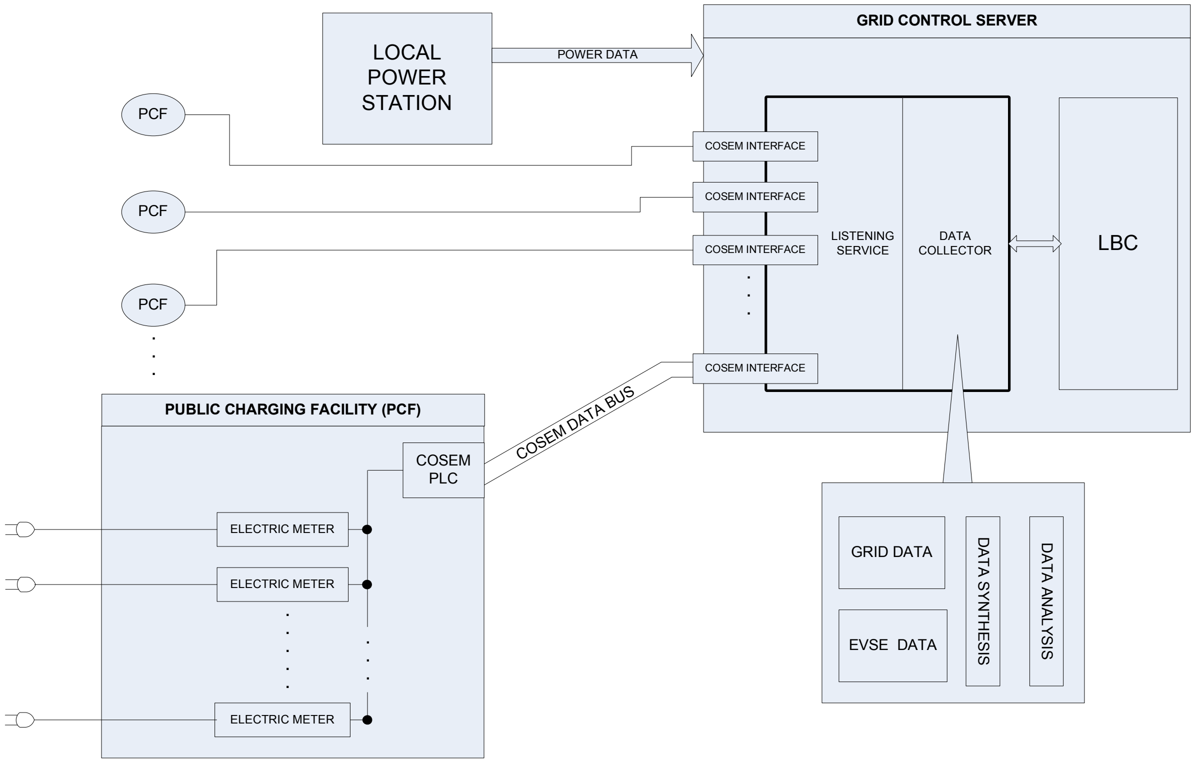
Grid Control Server	This is the name for the software infrastructurethat will implement the grid control algorithms and the information flow described in T 5.3.

This server will require at least the following components:

Listening Service	Component that waits for data from the localPCFs in GCS’s network.
Data Collector	Is a group of components that monitor the stream of incoming data (both data from theLocal Power Station and PCF EVSE data) andapply the grid control algorithms providing thenecessary outputs to the Load Balance Controller component.

8.1.2 LBC Topology 2

In this topology, the EV will connect to the EVSEusing the V2G protocol according to ISO/IEC 15118-2. Between the EVSEand the GCS the extended/modified DLMS/COSEM protocol according to the IEC 62056 series standard may be used.



This topology assumes the use of DLMS/COSEM protocol at the charging facilities, to communicate with the GCS based on the specifications of WP4.

<u>Components:</u>

Further to the components already presented during the previous scenario, the following will also be required.

DLMS/COSEM PLC : This is a DLMS/COSEM profile over PLC implementation to transmit COSEM messages from the electric meters to the Grid Control Server.

DLMS/COSEM Interface	*This component is used to adapt messages coming from the DLMS/COSEM link to the Grid Control Server listening service.*
PCF	Local Public Charging Facility may connect to the Grid Control Server using either the V2G protocol (15188 data bus) or the DLMS/COSEM protocol. In the case of V2G interfacing, bidirectional

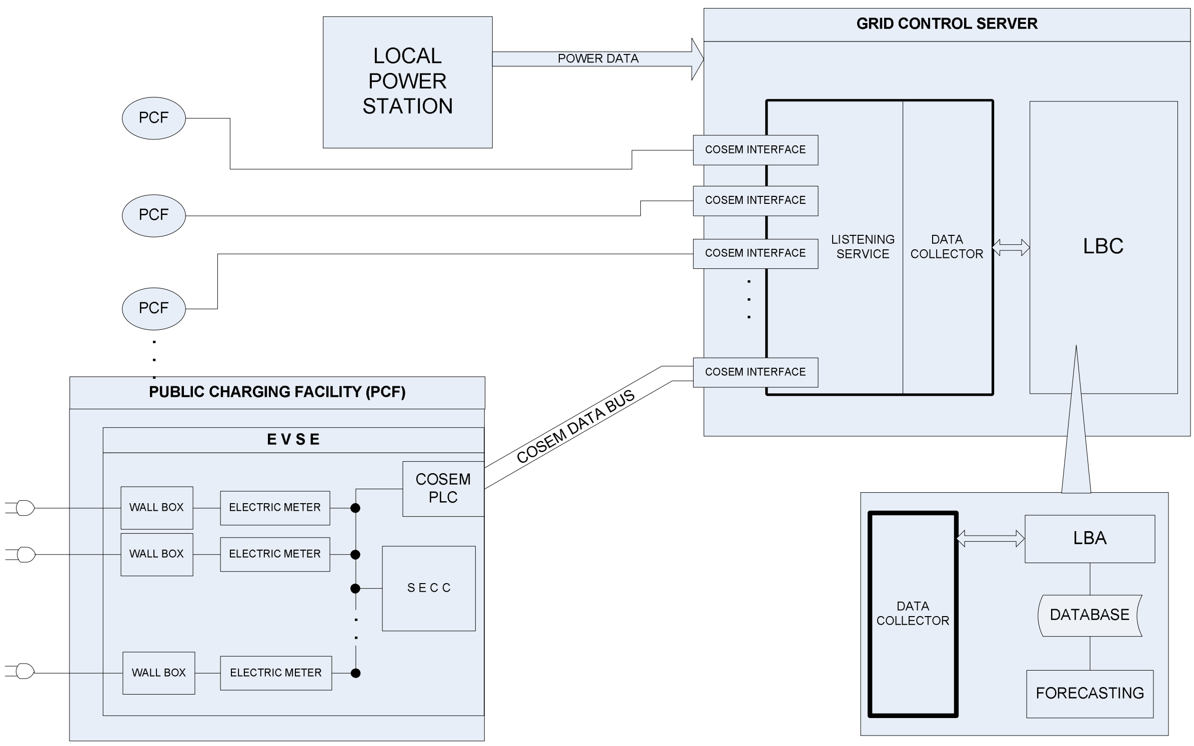
communication with the Grid Control Server can be provided through the EVSE Data Aggregator application

(LBC Topology 1).

9.2. Integrated Load Balancing Topology

According to D4.2, the following topology is considered as the integration of the Load Balancing within PowerUp.

This topology is employing the V2G communication between the EV and EVSE, and the use of the DLMS/COSEM protocol, according to IEC 62056, extended adequately for EVs. This topology is also relevant, since the DLMS/COSEM protocol is already widely used and favoured by the utility metering industry, so the consortium believes that it will be the future market trend for smart meters.



In this topology, the EV will connect to the EVSE through the Wallbox,using the V2G protocol according to ISO/IEC 15118-2. Between the EVSEand the GCS the extended/modified DLMS/COSEM protocol according to IEC 62056 will be used.

<u>Components:</u>

Further to the components already presented during the previous section, the following will also be required.

DLMS/COSEM PLC	This is a PLC implementation to transmit COSEM messages from the electric meters to the Grid. Control Server.
SECC	Manages all the process not directly linked to energy measurement and delivery. It can be embedded to the smart meter or not. It controls the communicationbetween EV and the EVSE, and all of the charging process.
COSEM Interface :	This component is used to adapt messages coming from the COSEM Data Bus to the Grid Control Server listening service.
PCF	Local Public Charging Facility may connect to the Grid Control Server using either the V2G protocol (15188 data bus) or the DLMS/COSEM protocol. In the case of V2G interfacing, bidirectional communication with the Grid Control Server can be provided through the EVSE Data Aggregator application (previous section, LBC Topology 1).
LBA	The algorithm which controls the Load Balancing
Forecasting	The algorithm which performs the forecasting of the maximumavailable power, provided by the Local PowerStation.

This topology is assuming that historical and real time data will be provided by the Electricity Operator.

9.3. Required LBC Parameters

Input variables to the Load BalancingController.

Charge Parameter Discovery Request

- DepartureTime
- EAmount
- EVPriority

Metering Receipt Request (in every loop)

- MeterReading.Value

Maximum Available Power from the Electricity Operator.

- PGrid

Output variables of the Load BalancingController.

Charge Parameter Discovery Response

- PMax

9.4. Parameters' Mapping based on Existing & Enhanced Protocols

The following table summarizes the needed information exchanges between EV and EVSE during all implementation steps.

Step	Implementation Method Name	Data Requirements	LBC Required Parameters
1	Handshake	Protocol Entry # Protocol Namespace Version SchemaID Priority	-
2	sessionSetup	Header SessionID EVCCID	-
3	serviceDiscovery	Header SessionID ServiceCategory	-
4	servicePaymentSelection	Header SessionID SelectedPaymentOption ServiceID	-
5	chargeParameterDiscovery	EVRequestedEnergyTransferType DepartureTime EAmount EVMaxCurrent EVMaxVoltage EVMinCurrent	DepartureTime EAmount EVMaxCurrent EVPriority
6	powerDelivery	ReadyToChargeState	-
7	chargingStatus	-	-
8	meteringReceipt	SAScheduleTupleID MeterInfo.MeterStatus MeterInfo.MeterID MeterInfo.isused.MeterReading MeterReading.Value MeterInfo.TMeter	MeterReading.Value
9	stopSession	-	-

In the table above an example implementation of the communication between EV and EVSE is assumed. This communication is based on the ISO/IEC 15118-2 and the example implementation is openV2G.[1] Against each method of this implementation, the parameters required by LBC are indicated, using so to secure compatibility with ISO/IEC 15118-2.

[1] http://openv2g.sourceforge.net/

10CONCLUSIONS

With objects identified above, a Smart Meter must be able to manage a V2G charging session as defined in the ISO/IEC 15118-2 process, using DLMS/COSEM protocol. Regarding the DLMS/COSEM Application Layer, services available allow performing these processes at a Metering point of view.

Objects already available in the DLMS/COSEM Blue Book, in addition with those defined in the present document cover all the need as identified during this work. They may be needed to be adjusted during the WP5.

11REFERENCES

Khalifa, T., Naik, K. & Nayak, A.A Survey of Communication Protocols for Automatic Meter Reading Applications. *IEEE Communications Surveys Tutorials* **13**, 1-15 (2010).

Smart Meter Requirements: Dutch Smart Meter specification and tender dossier. By order of: EnergieNed. Date: February 4th, 2008. Version: 2.1 final

OPEN meter - Open Public Extended Network metering: D2.3 - IDENTIFICATION OF RESEARCH NEEDS FROM BOTTOM-UP APPROACH - KNOWLEDGE GAPS

12ANNEX A: PAYMENT OBJECTS

12.1. Interface classes Accounting control

The accounting control interface class manages all the necessary elements related to the balance between the credit purchased by the consumer and the amount of energy consumed.

Accounting control IC holds all the attributes related to accounting process and configuration whilst the meter is configured in prepayment mode.

12.1.1. Scenarios Supported

This list of scenarios is not exhaustive, but will act as a starting point for ensuring suitability of the later interface classes, outlined in this document.

1) Current prepay customer steady state
2) Current prepay customer steady state moving to new tariff (does not affect account but may affect tariff charge)
3) Current prepay customer steady state moving house (CoT)
4) Current prepay customer steady state changing supplier (CoS)
5) Addition and consumption of credit
    a. Payment of standing charges
    b. Payment of debts
    c. Energy consumption charges
6) Use of EMC
7) Management of non-disconnection periods (Friendly Credit)
8) Control of supply (delivery)
9) Load limiting (delivery)
10) Taking of snapshots for event recording

Accounting control	**1...n**	**classic = 30, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max***	***Def.***	**Short name**

1. Logical_name	(static)	octet-string				x
2. account_mode	(dyn)	structure				x + 0x8
3. current_credit_in_use	(dyn)	unsigned				x + 0x10
4. status	(dyn)	enum				x + 0x18
5. available_credit	(dyn)	double_long				x + 0x20
6. amount_to_clear	(dyn)	double_long				x + 0x28
7. credit_reference	(static)	erray				x + 0x30
8. charge_reference	(static)	erray				x + 0x40
9. account_activation_time	(static)	octetstring				x + 0x48
10. account_closure_time	(static)	octetstring				x + 0x50
11. currency	(static)	octet-string				x + 0x58
12. low credit threshold	(static)	double-long				x + 0x60
13. next_credit_available_threshold	(dynamic)	double-long				
14. max_proportion	(static)	long unsigned				x + 0x68
15. max_proportion_period	(static)	double_long				x + 0x70
Specific methods		**m/o**				
1. activate_account(data)		O				x + 0x68
2. close_account(data)		O				x + 0x70
3. reset_account(data)		O				x + 0x78
4. update_credit (data)		O				x + 0x80

12.1.2. Attribute description

logical_name	Identifies the “Accounting control” object instance.
account_mode	This attribute defines the accounting mode which is credit or prepay, and also the status of the account.

	structure
	{
	payment_mode: enum,
	(1) Credit mode
	(2) Prepay mode
	
	account_status: enum,
	(1) New account
	(2) Account active
	(3) Account closed
	}
	This mode indication is not necessarily a parameter that will force some other actions in the meter, or cause a change of behaviour. However it is designed to be used as an indication of a notional prepay, or credit mode. The actual operation of the meter will be defined within the configuration of charge, credit, accounts functions and disconnection rules. There is a possibility that this indicator can switch from credit to prepay, once a prepayment configuration has been correctly set up.

current_credit_in_use	This attribute is an index into the array of credit_reference indicating which credit is in use at the point of query.
status	This attribute provides the situation of the current credit used. Bit Map (0) in credit, meaning that the available credit level is above the low credit threshold (1) low credit, meaning that the available credit level is below the low credit threshold (2) out of credit, meaning that the credit in use has been exhausted, and an interaction is needed in order to resolve the situation. (3) EMC Available (4) EMC Accepted (5) EMC In use (6) RESERVED (7) RESERVED
available_credit	Holds the current balance of credit in the meter that is in use. This value can be positive or negative and may or may not include EMC depending on whether it has been invoked.

	When it is negative this means that the account is in debt. This figure represents the real time credit in the meter and does not consider the reconciliation of future debt payments scheduled for the future. Future debt payments are considered in the charge objects.
amount_to_clear	This attribute is used when the meter has accumulated a temporary debt when in emergency credit, and/or an anti-social non-disconnect period. This attribute contains the minimum amount of credit that a consumer must add to the meter in order to re-enable supply. For example in the situation when emergency credit has been invoked this amount to clear would be the amount of credit needed to raise the available credit to an amount above the threshold where emergency credit is available (low_credit_threshold) for selection.
credit_reference	This attribute holds the profile of all types of credit managed by the device. Its data type is a array of logical names. All the instances identified by the logical name inside the array, are of type credit class Id. From IEC62055-21 clause 13.7.2 The type of the attribute is an array of logical names credit_reference ::= array of octet-string
charge_reference	This attribute data type is an array of logical names. All the instances identified by the logical name inside the array, are of type charge class Id. From IEC62055-21 clause 13.7.2 charge_reference ::= array of octet-string This array relates to a number of charge references that are separate objects and the object IDs are octect-strings.
account_activation_time	Date & time at which the account will be valid, and as such the supply enabled. When this date & time is in the past, this field serves as a

	record for when the account was first activated
account_closure_time	Date & time from which the account will be invalid therefore deactivated. The supply must at this date disabled in prepay mode. When this date & time is in past, it indicates at which date the account has been closed.
currency	Currency unit expressed using the 3 character international symbol (i.e. GBP, USD,etc)
low_credit_threshold	This attribute applies to the available_credit attribute within the accounting class and is derived from the warning_threshold attribute of the current credit in use. . This threshold can be used to generate a warning to the consumer, for example.
next_credit_available_threshold	This attribute applies to the available_credit attribute in the accounting class and is the threshold in credit terms at which the next priority credit is available. This threshold is derived from the credit_available_threshold attribute in the next priority credit object. If there is no next priority credit object then this becomes zero.
max_proportion	This attribute holds the limiting amount per unit time (eg per week) that can be taken as a proportion of top-up amounts. Related to "proportion" attribute of Charge class
max_proportion_period	This attribute holds the time period applicable for the Max_proportion field.

12.1.3. Method description

activate_account(data)	This method allows the activation of the supply at the service delivery point data ::= 0
close_account(data)	This method allows the closure of the delivery at the concerned service delivery point due to a change of supplier or change of tenancy or any other reasons. data ::= 0.
reset_account(data)	Reset all the data related to credit and charge. This reset is performed when a new tenancy comes in the premise or a new supplier has to manage the meter. A given account can only be reset after its closure. Resetting an active account is not permitted.

	data ::= date and time when the service must be reset
update_credit(data)	This is done by a credit token. The result of this process can be a credit addition or credit removal. data ::= double_long

The functionality of Emergency Credit is illustrated in section **Hiba! A hivatkozási forrás nem található.**.

12.2. Credit interface class

This IC allows the management of a given credit type inside the credit profile. The related credit type can be configured or not.

There are several types of credit listed in IEC62055-21 and these are the credit types that are available for selection and description within this interface class:

token credit: Credit that is transferred to the payment meter in the form of Credit Tokens. Examples are: purchased credit, free issue credit and token replacement credit.

reserved_credit: Credit that is held in reserve, which is released under specified conditions.

In some instances reserve credit is left on the token carrier for later release. One use of this feature is to ensure that consumers budget some reserve for use at night or during winter.

emergency_credit: Credit that is released only under emergency situations. Usually theamount of emergency credit used is recovered from subsequent purchased credit token.

time_based_credit: Credit that is released on a scheduled time basis. Examples of these are:

Indigent (poverty or social)

Grant of credit that is made available on a monthly basis, for example: £20 per month.

consumption_based_credit: Credit that is released on the basis of a schedule of consumption levels. For example indigent grant of credit that is made available, as long as the consumer’s monthly consumption is below 300 kWh.

Credit	0...n	class_id = 31, version = 0			
Attribute(s)	***Data type***	***Min.***	***Max***	***Def.***	**Short name**

1. logical_name	(static)	octet-string				x
2. type	(static)	enum				
2. selectable	(static)	boolean				x + 0x8
3. priority	(static)	enum				x + 0x18
4. warning_threshold	(static)	integer				x + 0x20
5. credit_available_threshold	(static)	integer				x + 0x28
6. period	(static)	date-time				x + 0x30
7. limit	(static)	long_unsigned				x + 0x38
8. amount	(static/dynamic)	octet-string				x + 0x40
9. credit_configuration	(static)	unsigned				x + 0x48
Specific methods		**m/o**				
1. update_ (data)		O				x + 0x50

12.2.1. Attribute description

logical_name	Identifies the "credit" object instance.
type	This identifies the type of credit that this object refers to. The types available are as IEC62055-21: 1. token credit: 2. reserved_credit: 3. emergency_credit: 4. time_based_credit: 5. consumption_based_credit:
selectable	This attribute defines if the credit concerned by this instance requires some consumer interaction after it becomes available but before if comes into use. If false, this means no consumer interaction is needed in order to bring it into use. and if true then. For example in the case of emergency credit this flag is false when emergency credit is not available for selection, and is set true by the meter when the low credit

	threshold has been reached and emergency credit is made available.
priority	This attribute describes the priority of each credit type, inside the credit profile. The range is from 1 to 5. Value 0 is to indicate never used Value 1 is then highest priority and value 5 the lowest. The credit management inside the credit profile uses always to start with the highest priority is ever it is enabled. When the highest priority credit is exhausted, the account object makes use of the priority immediately below, if ever enabled and so on…
Warning_thresold	This attribute holds the threshold which if it is ever reached, triggers a warning for the consumer that credit is low It may also trigger Emergency Credit to be offered..
Credit_available_threshold	This attribute specifies a limit at which the credit becomes available and can be selected by the customer. For example if this credit object was configured to be Emergency Credit then it would be available when the total credit reached this value. If this is set to "0" then the credit is always available subject to its priority.
limit	This attribute holds the credit amount which if ever it is reached, leads the concerned credit to be considered as exhausted. (Normally set to zero)
Amount	This attribute provides the value of the amount of credit that is added to the available_credit attribute in the account object when this particular credit is activated This value is static and set by configuration. For token credit, the amount is always 0, meaning that for having knowledge of its value, the user has to refer to available credit when the token credit is used (decrypted).
Credit_configuration	This attribute allows the user to configure a number of items regarding the behaviour of the particular credit object. 0: Credit allows charge collection 1: requires visual indication 2: Requires confirmation when invoked 3: RESERVED If bit 0 is set then all charge objects that are set for collection

	from certain credits are allowed to collect from this credit as per their configurations. For example some charges are not collected when EMC is invoked. If bit 1 is set then the credit item requires a visual indication on the meter display. If bit 3 is set, confirmation is required from the customer before moving into this type of credit (such as EMC, when the customer is required to press a button before EMC is invoked)

12.2.2. Method description

update_credit(data)	This method allows the update of the credit situation. It allows credit addition and subtraction. data ::= double_long Token credit, reserved credit and emergency credit are updated via the account instance. Time_based and consumption based credit are updated via the credit instance

12.3. Charge interface class

An instance of this interface class allows charge management. Depending on the configured amount per price and the period, the concerned charge is taken from the available credit.

Each charge object identifies itself by its attributes. There are several charge types available as described below and these are referenced in the charge_reference attribute of the Accounting Class. Belos is an explanation of the different types of charge:

tariff_rate_charge object is time based tariff rate charge, consumption based tariff rate charge, or both. A price per unit is assigned to each consumed energy rate inside the object identified by the logical name in the structure (See charge class Id).
time_based_auxiliary_charge is the result of all the time based charges to be applied to the meter. Eg standing charges, or debt charge to be paid off over a period of time..
consumption_based_charge is the result of all the charge which which have a dependency of the amount of energy consumed (Eg VAT…) i.e. with every vend the consumer pays say 20% of the vend amount towards this charge.
Percentage_based_charge is a charge that is applied at the time a new vend amount hits the credit bank, i.e. the consumer pays £2 out of every vend regardless of the vend amount.

Charge		**0...n**	**class_id = 32, version = 0**			
Attribute(s)		***Data type***	***Min.***	***Max***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. active_price_per_unit_charge	(static)	Structure				x + 0x8
3. passive_price_per_unit	(static)	Structure				x + 0x18
4. period	(static)	Array				x + 0x28
5. proportion	(static)	long-unsigned				x + 0x30
6. total _amount	(dynamic)	double_long				x + 0x38
7. last payment _date	(dynamic)	octet-string				x + 0x40

8. activation_date (static)	octet-string				x + 0x48
9. Priority (static)	Enum				x + 0x50
10. special_charge_configuration (static)	unsigned				x + 0x58
Specific methods	**m/o**				
1. update (data)	O				x + 050

12.3.1. Attribute description

logical_name	Identifies the "charge" object instance.

active_price_per_unit_c harge	This attribute defines the active price reference
	. active_price_per_unit_charge structure
	{
	energy_reference octet-string
	price_unit integer
	amount_per_price array of long
	}
	- energy_reference is the logical name of the energy used as reference for the charge computing. This must be one of the energy registers, (class Id 3). It should be possible to configure the service kind to specify the number of decimal places for the charge relating to the fuel type, for example a charge of £0.2345/kWh would be configured as -4.
	- amount_per_unit is the charge amount per unit. It can be positive or negative. For example, if the end consumer generates energy then the charge should be negative, means that the utility has to pay the end consumer for the energy he produces.
	For service_fix_charges and percentage_charges the array must necessary have one element applying for the global total charge that is not dependent on the tariff.

passive_price_per_unit_ charge	This attribute holds the charge to be applied at the occurrence of the activation date. Its data type is the same than the active_price_per_unit one.
Period	This attribute period is related to the charge cycle. - For tariff rate charges and percentage based charge, it is the periodicity of the available credit updating, by subtracting the value of the charge.

	For tariff rate charges and percentage based charges this value specifies the period (in seconds)at which the specified amount of charge is deducted from the available credit register. - For Time based charge, it defines the period at which the charge will be taken. There are as many elements in the array as in the amont_per price, allowing the definition of the charge periodicity for each element in the array. Period ::= array of double_long_unsigned The period is expressed in seconds
Proportion	This attribute is related to charges that take a proportion of each top-up amount. Related to "max_propotion" in Account class
total_amount	This attribute holds the total charge taken for this type of charge.
last payment _date	Holds the date and time when the last payment took place
price_activation_date	Holds the date when the passive_charge_per_unit_price becomes the active_price_per_unit_charge.
Special_Charge_configuration	This attribute allows this particular instance of charge to be collected in all credit types. There is a corresponding attribute associated with each Credit type which specifies that charges can be collected or not. This attribute also allows certain other configuration items to be set or not by use of a bitmap. 0: Collect only in specified Credit Types (see Credit IC) 1: % based charge 2: RESERVED 3: RESERVED If bit 0 is set then this means that this charge type can only be collected in specified credit types. Refer to credit type for the corresponding attribute. If this bit is not set then the meter shall collect this charge type in all credit types. If bit 1 is set then this charge is a percentage based charge and will collect the appropriate amount as a % of a vend when a token credit is applied.

12.3.2. Method description

update(data)	This method allows the update of the the amount_per_price inside the passive charge.

	data is of type array of structure
	
	data ::= array of structure
	
	{
	
	index unsigned
	
	amont_per_unit_element long
	
	}
	
	Index is the order of the amount_per_price_unit_element inside amount_per_price unit.

12.4. Relation to OBIS

Pre-payment is valid for any type of energy.

An instance of the "Accounting control" IC holds, all the configuration needed for a given energy type.

Several instances of credit IC allow the different credits management.

Several instance of charge IC allow different charge management.

An instance of limiter interface class allows reducing the delivery in case of credit exhaustion.

An instance of script interface class allows the management of prepayment scripts.

An instance of extended register holds charge or debt. Value identifies diverse types of debt. When e=0, this means that the attribute 2 of the related extended register hold the total amount of charge or debt.

An instance of extended register holds the last credit added.

An instance of profile generic holds the historic of the top up of the credit.

An instance of profile generic holds the historic of the charges.

Prepayment objects	IC	OBIS identification

		A	B	C	D	E	F
Account	30, Accounting control	0	*b*	30	0	0	255
credit	31, credit	0	*b*	30	1	e	255
charge	32 charge	0	*b*	30	2	e	255
Charge _register	4, Extended register	0	*b*	30	4	e	255
Last_added_credit register	4, Extended register	0	*b*	30	5	0	255
Limiter	71, Limiter	0	*b*	17	0	e	255
Disconnect control	70, Disconnect control	0	*b*	96	3	10	255
Prepayment script table	09, Script table	0	*b*	10	0	108	255
Charge_profile_history	07, profile generic	0	*b*	99	13	e	255
To_up_profile_history	07, profile generic	0	*b*	99	14	0	255

13ANNEX B: OBJECTS IDENTIFIED:

Below are the up object identified for the Electric vehicle charge management. They are listed into two parts: Objects which are already available and new objects created for this specific purpose. The reader has to be aware that during the prototyping phase additional objects may be brought.

13.1. Available objects

Name	Class Id	Logical name	Data type
Max_demand	Register monitor	1-b:16.0.40.255	Array unsigned32
Ev_script	script	0-b:10.0.110.255	Method
Activity calendar	Activity calendar	0-b:13.0.0.255	array
Special days table	Special days table	0-b:11.0.0.255	array
Tariffication script	script	0-0.10.0.106.255	Method
Switch_status	Disconnect control attribute 3	0-b:96.3.10.255	enum
Evse_id	data	0-1:95.2.2.255	octetstring
evse_status	data	0-1:96.5.0.255	bitstring
current_date_time	Clock object	0-0:1.0.0.255	octetstring
maximum_power	Extended register	1-0.8.6.0.266	Unsigned32
Current power	Extended register	1-0.8.24.0.255	Unsigned32
meter_ status	data	0-0:96.5.0.255	bitstring
Status_profile	Profile	0-0:98.d.e.255	array
EV_status	Data	0-2:96.5.0.255	enum
voltage	register	1-0.12.24.0.255	Long-unsigned
Current	register	1-0:11.24.0.255	Long-unsigned
Demand	register	1-0:1.4.24.0.255	Unsigned32
Metering receipt	Profile IC	1-0:98.1.0.255	Array of structure

voltage	register	1-0.12.24.0.255	Long-unsigned
Current	register	1-0:11.24.0.255	Long-unsigned
Demand	register	1-0:1.4.n.255	Unsigned32
Energy register	register	1-0.1.8.n.255	Unsigned32

13.2. New objects

Name	Class Id	Logical name	Data type	length
session_id	Data	1-b:95.2.0.255	octetstring	8
EV_id	Data	1-b:95.2.1.255	octetstring	32
evse_id	Data	1-b:95.2.2.255	octetstring	32
Service_id	Data	1-b:95.2.3.255	octetstring	8
Contract_id	Data	1-b:95.2.4.255	octetstring	128
Energy_provider_id	Data	1-b:95.2.5.255	octetstring	256
Identification	Profile	1-b:95.2.255.255	Array	6 elements
charge_profile	Data	1-b:95.3.0.255	Array	
EVSE status	Extend register IC	1-b:95.2.10.255	Bitstring	8bits
Receipt_profile	profile		array	