Deliverable 3.6

Core services and transactions design specification Release 2

Prepared by:

Betterplace, BMW, Bosch, Daimler, Endesa, EDF, Enel, RWE, SAP, Siemens
Under the lead of IBM

Date: September 30th, 2013

Version: 1.1

Document Information

Authors

	Name	Company
Key author Key author	Andreas Herdt, Jan Laczay, Martin Rapos Chapter 3 – Thomas Gereke	IBM Siemens
Key author	Chapter 4 – Tom Kiemes	SAP
Key author	Chapter 5 – Teresa Casacchia,	Enel
Key author	Chapter 6 – Andreas Herdt, Thomas Stiffel	IBM, Bosch
Additional author	Giovanni Coppola	Enel
Additional author	Volker Fricke, Norbert Reschauer, Ruth Schilling	IBM
Additional author	Stephan Cater	RWE

Distribution

Dissemination level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision history

Version	Date	Author	Description
Draft 0.1	11-7-2013	Andreas Herdt	Initial version based on D3.5
Draft 0.2	19-7-2013	Martin Rapos	Version for WP internal review
Draft 0.3	2-8-2013	Andreas Herdt	Version for WP external review
Final 1.0	30-8-2013	Andreas Herdt	Version for Project Coordinator review
Final 1.1	30-9-2013	Andreas Herdt	Version for EU review
Final 1.2			
Final 1.3			

Status	
For Information	
Draft Version	
Final Version (Internal document)	
Submission for Approval (deliverable)	x
Final Version (deliverable, approved on)	

List of Abbreviations

B2B	Business to Business
B2C	Business to Consumer (aka: Business to Customer)
BC	Business Component
BIN	Battery Identification Number
BO	Business Object
BSS	Battery Switch Station
CDR	Charge Detail Record
CEE	Commission on the Rules for the Approval of the Electrical Equipment
CH	Clearing House
CHAdeMo	CHArge de Move
CLM	Charging Location Management
CM	Congestion Management
CMS	Charging Management System
CP	Charging Point
CSMS	Car Sharing Management System
DC	Direct Current
DIN	Deutsches Institut für Normung
DM	Domain Model
DSG	Dynamic Service Gateway
DSO	Distribution System Operator
E2E	End to End
EMM	Electro-Mobility Management
ENUM	Enumeration
ES	External Stakeholder
EV	Electric Vehicle
EVCOID	Electric Vehicle Contract ID
EVSE	Electric Vehicle Supply Equipment (Charge Point)
EVSEID	Electric Vehicle Supply Equipment ID (Charge Point / Unit)
EVSP	Electric Vehicle Service Provider
FD	Functional Decision
GeM	Green eMotion
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
IT	Information Technology
kWh	Kilowatt hour
LA	Load area
LM	Load Management
LMDR	Load Management Detail Record
LV	Low voltage
M2M	Machine to Machine
MP	Marketplace
MV	Medium Voltage
OEM	Original Equipment Manufacturer

POD	Point of Delivery
POI	Point of Interest
REN	Renewable energies
RFC	Request for Comments
RRC	Rational Requirements Composer
SAE	Society of Automotive Engineers
SDR	Service Detail Record
SKT	Sketch
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQD	Sequence Diagram
SRV	Service Interface
TSO	Transmission System Operator
UC	Use Case
UI	User Interface
UML	Unified Modeling Language
URI V2G	Uniform Resource Identifier Vehicle to Grid
VDE	Verband der Elektrotechnik Elektronik Informationstechnik e.V.
VIN	Vehicle Identification Number
WP	Work Package
WS	Web Service
WSDL	Web Service Definition Language
XML XSD	Extensible Markup Language XML Schema Definition

Table of Contents

List of Figures

1 Executive summary

The genesis of the Green eMotion requirements description started with a comprehensive stakeholder analysis and high level business scenarios in deliverable D3.1 and were with the use of applied meta-model transformed with increasing granularity to features and use cases described in the D3.3 deliverable. Selection of these requirements was further detailed by the IT developers in the specification phase in deliverable D3.5, leading to the first release of the Green eMotion Marketplace. The D3.4 deliverable incorporated feedback from demo regions and the executive board as well as the lessons learned from the first Release implementation. Carefully selected use cases from the D3.4 are specified in this deliverable. This deliverable thus comprises the complete specification of the key business components, service interfaces and user interfaces for the Green eMotion Marketplace eco-system and reflects the Phase 2 and final implementation of this ecosystem. All mentioned deliverables are available on the Green eMotion web site http://www.greenemotion-project.eu/dissemination/deliverables-ict-solutions.php.

Work on the D3.6 commenced with Business components identification. Business components are describing the selections of systems that need to communicate in order to implement a particular use case. In a second step the service interfaces between these systems have been specified. By the business to business nature of the Green eMotion project this specification considers solely the service interfaces between the back-end systems of the involved partners. Business objects are further describing the implementation details of the interfaces. As the primary goal of D3.6 is the provision of detailed requirements to the software development teams working on the Green eMotion marketplace and provided services, the applied methodology leverages the previously used structure and the Rational Requirements Composer (RRC) to extend the previously used meta-model. Practical implementation decisions describing the transition between Use cases and Business components are documented as “Functional Decision” and their aim is to enhance traceability between the high and low levels of requirements description.

Deliverable D3.6 is an export from the RRC, which was used as collaborative tool to create this document. Over one hundred use cases described in the previous deliverables outline the scope of business interactions and are in detail described around the central IT information platform. Components, interfaces and business objects of this platform, the Green eMotion Marketplace, are described in chapter 6, featuring all core Marketplace components and also interconnection between marketplaces. Chapter 3 describes technical components of the General electric mobility services such as Search and reservation of EVSE, Chapter 4 describes Roaming provided by a contractual Clearing house, and Chapter 5 encompasses the Energy and smart grid perspective, including congestion management for DSO.

This deliverable comprises those service interfaces, which have been agreed by major e-mobility partners. Some of these artifacts are already demonstrated by the demo regions of Green eMotion utilizing the first release of the Green eMotion Marketplace (e.g. Search EVSE, Roaming via Clearinghouse or Load management), others are to be developed and demonstrated in the second release of the Marketplace (e.g. Reservation of EVSE). These interfaces will enable the selected use cases and will build the technical foundation for further services and innovation in the emerging e-mobility market. This specification is vital for the implementation of the components that make up the Green eMotion Marketplace eco-system.

The following picture illustrates the scope of this deliverable, which are the components within the highlighted area as well as the interfaces from, to and between those components.

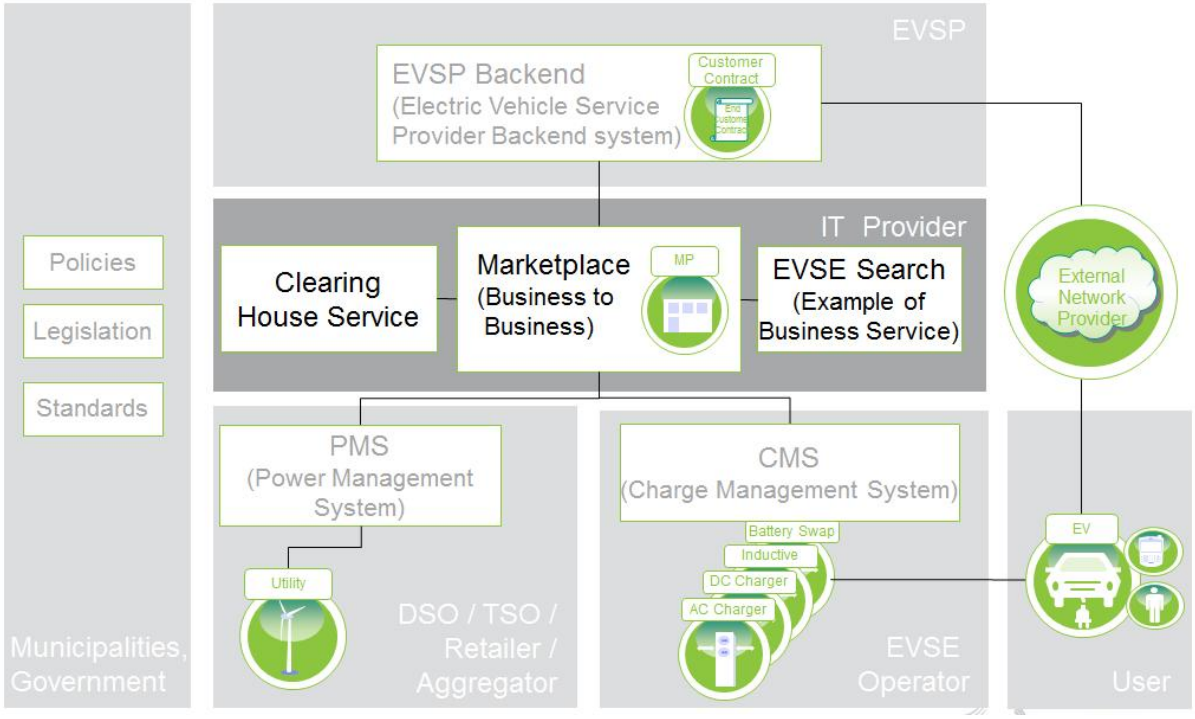


Figure 1.1: Scope of D3.6 Service specification for GeM Marketplace

The larger grey boxes represent domains under the control of certain roles (e.g. the highlighted domain is controlled by IT providers). The smaller boxes inside represent logical system components, which means that they do not need to have a one to one counterpart in the world of real hardware and software components. The lines between the logical components represent communication paths between those components. Services offered by the components and business objects exchanged between them are being described in detail in the corresponding sections.

While the focus of this document stays within the highlighted scope, in many cases the backend systems attached to the marketplace and their interaction with other components are being described briefly in order to provide a better understanding of the various scenarios.
The following deliverables (especially demo prototype and system release) will build upon this deliverable in the same way that D3.6 builds upon the previous work.

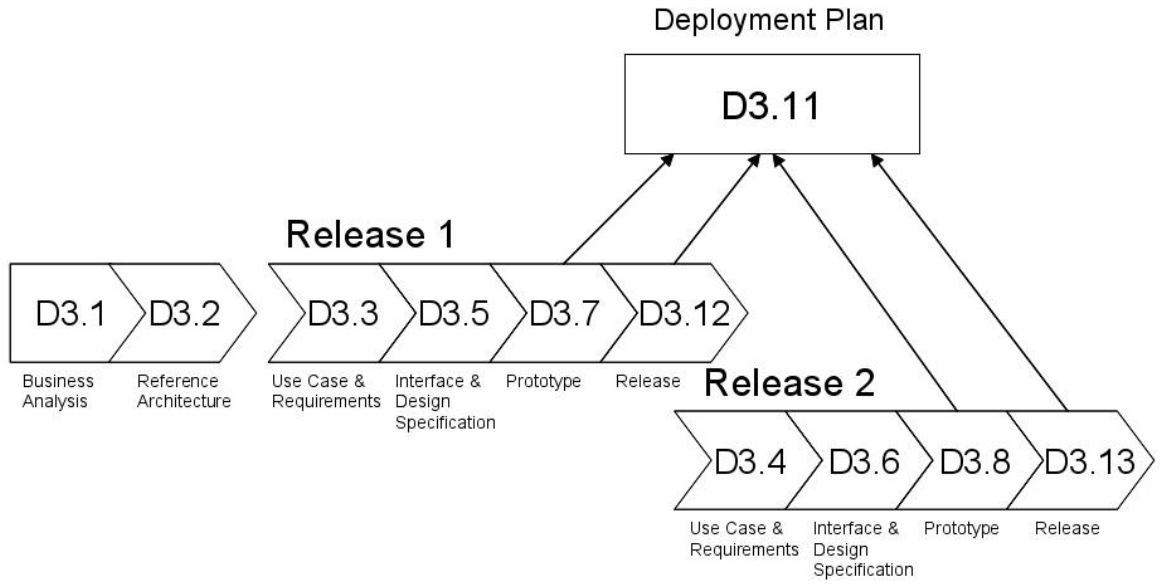


Figure 1.2 Sequence of deliverables in GeM

As D3.6, like all other release 2 documents, is designed as a full update to the preceding release 1 version, it is not required to read D3.5 in order to understand this deliverable. However, it is strongly recommended to read D3.4 before, as it contains the foundation D3.6 builds up upon. Both deliverables are linked together through the use cases contained in each of them.

Note: To achieve efficient retrieval of desired content from this comprehensive deliverable please refer to section “How to read this document” at the end of the Introduction chapter.

2 Introduction

This deliverable provides a thorough specification for both releases of the Green eMotion Marketplace eco-system. The basis for this specification is a subset of Use Cases from D3.4 that was selected based on the value it offers for the e-mobility mass market preparation supported by the involved partners and finally for the end-users. Additionally it was critical to identify dependencies between the existing Use Cases to define a scope that is self-contained and can be fully realized and demonstrated in the first release.

The creation of this deliverable uses the same process that was established for the Business Scenario and Use Case definition. This process is enabled by the use of the IBM Rational Requirements Composer Tool as central tool for the collaborative management of requirements and specification. This document is an export of the information maintained in this tool and requires the reader to understand its structure to be able to retrieve desired information efficiently. For detailed information, and for hints on efficient retrieving of content please refer to the section “How to read this document” at the end of this Introduction chapter. The methodology used for the specification builds upon the Structured Requirements Management Method that was already used in the requirements gathering process. The meta-model that was previously used was extended to enable the identification and specification of Business Components and Service Interfaces which make up the building blocks for a Service Oriented Architecture (SOA).

The D3.6 deliverable comprises functional aspects of e-mobility under the assumption of the use of shared information platform(s), the GeM Marketplace eco-system. Partners from relevant e-mobility related industries participating in this work package (WP3) have specified detailed interfaces for defined services, which are from their perspective most likely to be used in the demo regions already during the first release. These interfaces will enable the demo regions to decide which Use Cases they will be demonstrating and to identify the tasks that are required from their sides for the implementation of the Service Interfaces.

2.1 Division of requirements into functional domains

The input for this deliverable was created in four working groups working on separate Functional Domains. These domains have already been defined for the Use Case definition in D3.3 and D3.4. Each Domain involved multiple stakeholders but was led by a party with strong background in the specific domain.

- General Electric mobility working group was led by Siemens
- Roaming Domain working group was led by SAP
- Energy Domain working group was led by Enel
- Marketplace and multi marketplace domain working group was led by IBM

Although these domains are inseparable for EV, and contain substantial overlap, this approach made it possible to create substantial progress in a short time frame. The coherence was achieved by joint working sessions and by the participation of stakeholders across all working groups. This structure is also clearly visible in the design of this document.

The **General Electric mobility domain** is described in Chapter 3. The Charging Domain describes the Service Interfaces for two separate scenarios: “Search for a suitable Electric Vehicle Supply Equipment (EVSE)” and “Reservation of an EVSE”. These scenarios have been defined in several Use Cases that contained some overlaps. For the specification this was resolved and resulted in one Service Interface

specification. To maintain the traceability, all Use Cases in this domain that were selected are listed in this specification even if they reference the same Business Components and Service Interfaces.

The **Roaming Domain** is described in Chapter 4. The Roaming services that enable the basic functionality of contractual clearing by the Clearing House are described in this chapter. All required Service Interfaces for the contractual clearing are present. This Domain also relies on Service Interfaces described in the Marketplace Domain. For easier readability and to make this chapter self-contained these Interfaces are also listed in this Domain.

The **Energy Domain** is described in Chapter 5. The Use Cases that have been included are concerned with the load and congestion management scenarios between a DSO and an EVSE operator. Furthermore the services that make available current and historical usage data of EVSEs are specified. These services might be used in several business scenarios that therefore have not been specifically defined yet. This means that there is currently no Business Component that will require these Service Interfaces and therefore act as a client.

The **Marketplace Domain** is described in Chapter 6. The Core Marketplace Service Interfaces enable the use of the Marketplace to call services that have been contracted between B2B partners. In addition to that the Marketplace is closely linked with the Clearing House for the validation of Roaming agreements that will be managed within the Marketplace. Multi marketplace service interface has been a new addition to this chapter.
Apart of these Service Interfaces, the Marketplace offers functionality through User Interfaces for the offering and contracting of Services.

2.2 Methodology

In D3.3 and D3.4 the WP3 team developed a series of use cases that realize the business scenarios defined for the Green eMotion project.

These use cases were already defined as structured content within the Rational Requirements Composer (RRC) tool. Therefore it was easy to build upon the existing content and to take the next step to define a specification that links to the existing use cases. For this purpose the WP3 team used the Component Modeling Methodology to identify and specify the components and interfaces required for the realization of the previously defined use cases.

In order to support this methodology, the previously defined meta model had to be extended by the following new artifacts:

- **Business Component** – Logical part of an IT system that aggregates similar functionality and has a clear responsibility. Measures for a good definition of components are a high level of cohesion, a weak coupling with other components and a high level of isolation.
- **Service Interface** – formal description of services provided or required by a business component in order to fulfill its responsibilities.

Furthermore the previously defined business objects were adapted and new business objects were defined, in order to unambiguously define the data structures used in the service interfaces. The business objects that are referenced in this specification use a more formalized notation commonly used in data modeling. For more complex structures domain models were created using the Unified Modeling Language (UML) notation, to illustrate the relationships between business objects.

Before beginning the work on the specification, the WP3 team decided which use cases they wanted to focus on for the development releases, in order to have a manageable scope. These decisions were documented in the RRC tool. Only the use cases that were selected to be relevant for either release 1 or release 2 were subsequently considered for this specification.

In the following specification process the WP3 team had to make sure that the use cases have a consistent level of granularity. Use cases that describe a more general perspective of activities were linked with use cases that describe certain parts of these activities in more detail. Furthermore overlaps between use cases had to be identified, to avoid duplicate efforts in the specification.

In the next step business components were defined which will implement the use cases by offering services and by interacting with each other. The identified services had to be defined with service interfaces that exactly define which data is needed as input and what output will be created. This was a very time intensive task, as all the data had to be complete and described in an unambiguous way. In order to ease the integration of several services, common domain models were defined that describe the data structures either for a specific domain or for the overall project.

For the implementation the business components can now be assigned to one or more partners that will provide the described functionality for the demonstration. The service interfaces will be used to derive technical interfaces that allow the integration of the applications that will be developed.

The following pictures illustrate the extensions that have been made for the existing requirements management meta model.

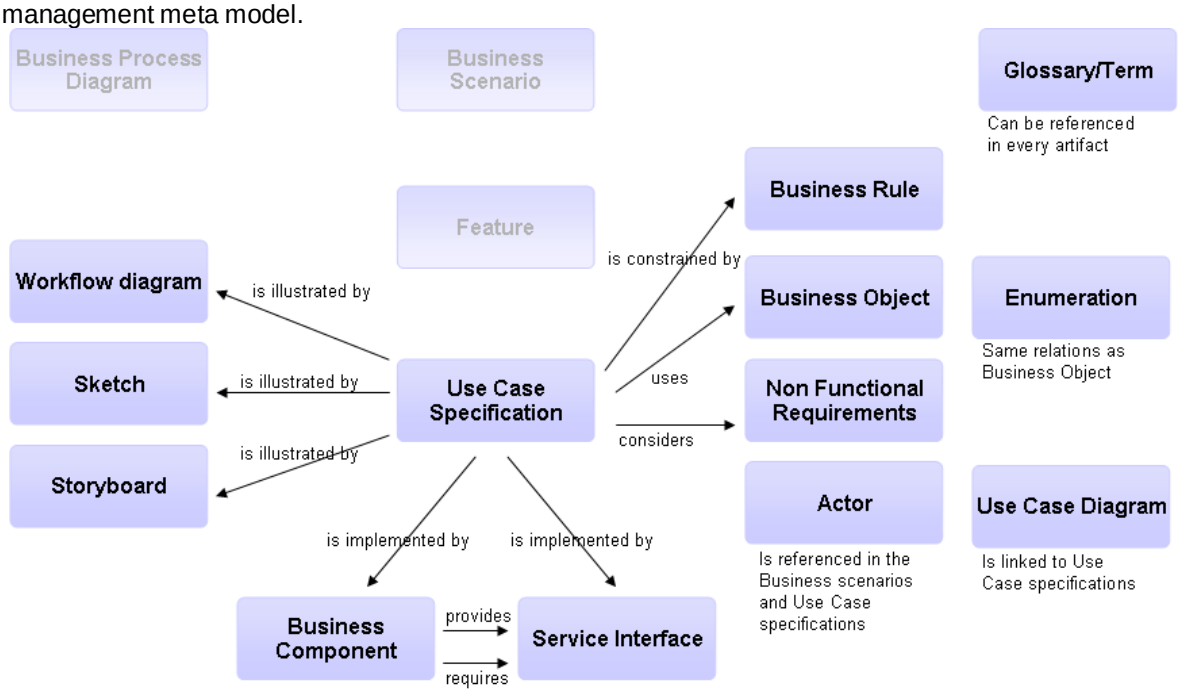


Figure 2.1 Extending the existing artifacts and relations

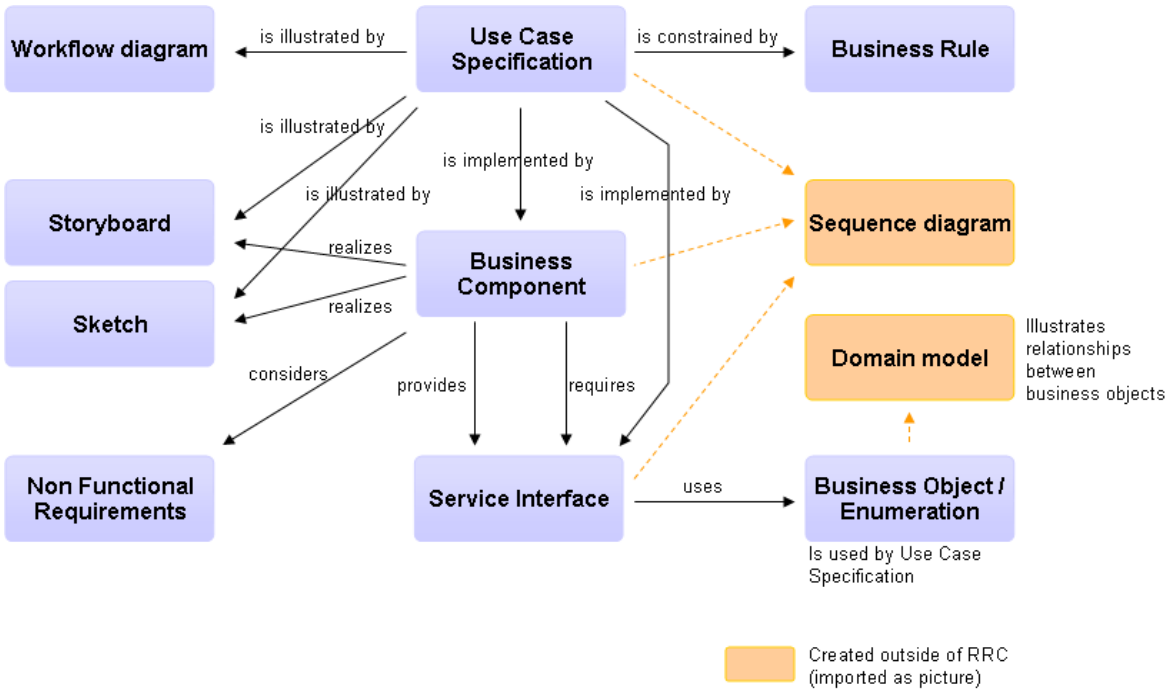


Figure 2.2 Overview of relevant artifacts for the specification

2.3 How to read this document

The main goal of this deliverable is to provide a reference for the implementation of the selected use cases for all partners. It is therefore assumed that the use cases defined in D3.4 are already known before reading this document. In order to create the link to D3.4, this document is divided into the same functional domains, which are further subdivided into the previously defined use cases.

The WP3 team suggests the following approach to reading this document:

- Read the introduction first
- Select the use case of interest (if required reference to D3.4 to understand the scenario described by the use case)
- Read the descriptions of the involved business components and service interfaces. If available, the included sequence diagrams can give you a quick overview of the interactions defined for the use case.
- To go into more detail you can read the attribute definitions for each service interface. You may need to refer to the domain model chapter of the functional domain if the service interface references common business objects defined for this domain.

Each functional domain starts with the chapters “Domain Model” and “Functional Decisions”, if those were defined for the domain. These contain a short introduction as well as information about the data structures and functionality of the domain and are relevant for several use cases.
The chapters “Service Specification” and “User Interface Specification” are both divided into chapters for each use case. Service interfaces and business components that are relevant for several use cases are only included once and are only referenced in subsequent use cases.

The service interface and business object definitions use a similar format to define the data structures:

Attributes	Datatype	Constraints	Description
Name of an attribute of either a business object or a service interface	Definition of the data type of the attribute. This can be either a primitive type (as defined in XSD standard: http://www.w3.org/TR/xmlschema-2/#built-in-primitive-datatypes) or a reference to a business object.	Constraints those are relevant for this attribute. This usually includes multiplicities, min and max values or other validation logic that can be formally described. Constraints can be defined for business objects or service interfaces. If a service interface references a business object, all constraints defined for the business object also apply for the service interface. A multiplicity of 0..1 (zero or one) or 0..* (zero or many) is equivalent to optional while 1 (exactly one) or 1..* (one or many) is equivalent to required.	This describes what this attribute represents and may also clarify the meaning of specific values, or of constraints that were described in the column before.

For service interfaces it is also required to list business objects that are only indirectly referenced by another business object but that will be part of the request or response data structure. This is required because the business objects that are referenced in service interfaces may be part of a complex domain model with many relations to other business objects that again might reference even more business objects. For a service interface you have to define which of these indirectly referenced business objects are relevant and which ones are not included in the request or response. Therefore business objects which are only indirectly referenced are included in the table, but instead of an attribute name it is stated by which business object they are referenced. The constraints column might list additional constraints (e.g. a multiplicity that is defined in the domain model as 1..* might be defined for a service interface as 1), however the original constraints still apply.

3 General E-Mobility

3.1 Domain Model

DM EVSE Domain

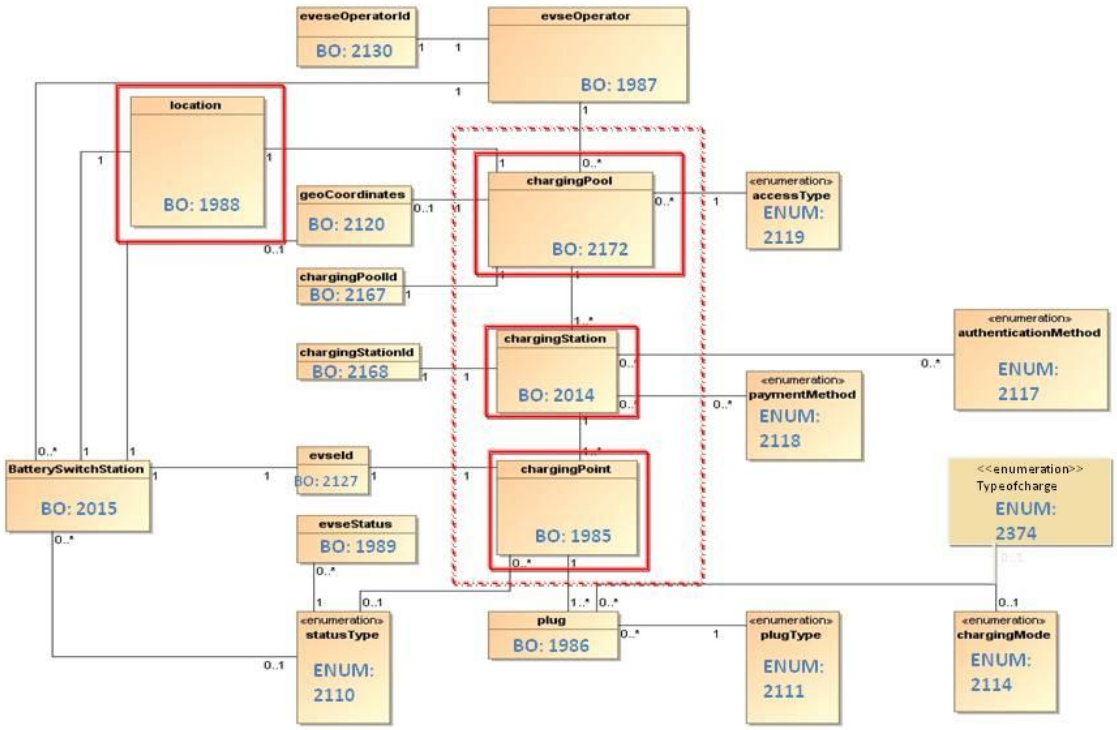


Figure 3.1: High level EVSE Domain model

The Charging UCs described in D3.4 are implemented by the Service Interfaces (SVR), Business Components (BC) and Business Objects (BO) described in this chapter.
The Charging services shall enable EVSP and the related end-user (EV-driver) to use the charging-infrastructure provided by the EVSE-operator.
In this chapter the charging services “Search of EVSE” (UC 1527) and “Reservation of EVSE” (UC 1528) are described. For this purpose the information flow between the Business Components, the attributes of the Business Objects and the relationship to the Service Interfaces are depicted.
The purpose of the charging end-to-end process execution is to support the EV-driver with a B2C charging services in order to allow an easy and comfortable EV-charging-process.
Release 1 specifications of charging services, described in D3.5, have already provided a comprehensive set of BCs, BOs and SRVs that will be further validated by D3.6.

E2E demonstrations for charging services in particular, two main BCs have been identified in the Charging Domain:

BC 1527 Search of EVSE (incl. advertising)
Within the EVSP-backend-system (e.g. Customer portal) the integration of the service Search of EVSE is necessary. On the other side the Search of EVSE will have service interfaces to get the master and actual status information from the EVSE-operator-backend and a service interface to get the relevant advertising data for the selected geo-coordinates.
On the other hand it could be also possible to have two different services, placed on the marketplace to archive this goal. This means one service for pure search of EVSE and on service for advertising based on the geo-search-criteria's.

BC 1528 Reservation of EVSE
The service reservation of EVSE can be used in order to support the EV-driver to reserve/book a EVSE in advance in order to charge the EV. This enables extended-range journeys with increased confidence.
For the reservation of EVSE it is necessary that the service placed on the marketplace will get the actual EVSE master-data, actual status data and charging-session data from the EVSE-operator backend system (CMS). The EVSP can use the service reservation of EVSE by integration of the relevant graphical user interface for configuration, search and manage the reservation of EVSEs.
The reservation of EVSE can be done in two different modes, point in time or time period. Furthermore the EVSP have the possibilities to do a short term (acute) or long term reservation.
Through these business components the EVSP and the EVSE Operator will be able to implement the charging services in order to increase the confidence of the EV-driver and increase the customer-relationship to the EVSP and also increase the degree of capacity of the charging infrastructure of the EVSE-operator.

The following specified SRVs respond to such use case:
- 2012: SRV Search EVSE
- 2045: SRV EVSE master data
- 2046: SRV EVSE status data
- 2065: SRV EVSE status data (push)
- 2370: SRV EVSE advertising data
- 2379: SRV EVSE session data
- 2366: SRV Reservation request
- 2053: SRV Reservation to CMS
- 2381: SRV Reservation status

A detailed description of these SRVs follows in the document.

2130: BO EVSEOperatorID

This object describes the unique identifier of an EVSE operator.
The EVSEOperatorID consists of the first two parts of the EVSE ID (Country Code - 2 alpha, two character country code according to ISO-3166-1 (Alpha-2-Code), Spot Operator ID - 3 alpha / digits). For further information on the EVSE ID, see 2127: BO EVSEID.
The Spot Operator IDs will be managed by the consortium during the demonstration phase. Afterwards, it is mandatory that Governments or the European Commission nominate one authority or a system of authorities which manage the Provider IDs as well as the Spot Operator IDs.

Attributes	Datatype	Constraints	Description
id	string	• max. length: 255 • required	Unique identifier of the EVSE or spot operator

1987: BO EVSEOperator

This object describes a EVSE operator which provides charging pool of EVSE.

Attributes	Datatype	Constraints	Description
evseOperatorId	BO EVSEOperatorId	• required	Unique identifier of the EVSE or spot operator
name	string	• max. length: 255 • optional	Name of the EVSE operator
web	string	• max. length: 255 • optional	Home page address
eMail	string	• max. length: 255 • optional	Contact e-Mail address
phone	string	• max. length: 20 • optional	Contact phone number

2172: BO ChargingPool

This object describes a Charging Pool. A charging pool consists of one or more charging stations. The EVSE operator is able to group a set of his charging points with the same location to a charging pool.

Attributes	Datatype	Constraints	Description
poolId	BO ChargingPoolId	• required	Identifier of a single charging pool
evseOperator	BO EVSEOperator	• required	Identifies the operator that operates this charging pool, e.g. spot operator
stations	BO ChargingStations	• Multiplicity: 1..*	A set of the charging stations the pool consists of.
location	BO Location	• required	Address of the charging pool
geoCoordinates	BO GeoCoordinates	• required	Geo coordinates of the charging pool
openHours	BO MultilingualText	• Multiplicity: 0..*	General information about opening hours and days in different languages

Attributes	Datatype	Constraints	Description
time zone	string	• Max length: 50 • optional	Identifies the timezone of the POI, see the zone names defined in IANA Time Zone Database for valid values, e.g."Europe/London", "Europe/Berlin"
accessType	ENUM AccessType	• optional	General access type of the charging stations within the pool

2120: BO GeoCoordinates

This object provides geo coordinates.

Attributes	Datatype	Constraints	Description
latitude	double	• required	Geo position: latitude
longitude	double	• required	Geo position: longitude

1988: BO Location

This object provides address of a POI.

Attributes	Datatype	Constraints	Description
country	string	• max. length: 2 • required	country code according to DIN EN ISO-3166-1
region	string	• max. length: 50 • optional	Region where the POI is located
city	string	• max. length: 50 • required	name of the city in national language
zipCode	string	• max. length: 10 • optional	Zip code / postal code, required, if country specific address format requires zip code
street	string	• max. length: 100 • required	Name of the street
houseNumber	string	• max. length: 10 • optional	House number

Attributes	Datatype	Constraints	Description
floorLevel	string	• max. length: 10 • optional	Indicates the floor level in which the POI is located.
timezone	string	• max. length: 50 • optional	Identifies the timezone of the POI, see the zone names defined in IANA Time Zone Database for valid values, e.g."Europe/London", "Europe/Berlin"

2167: BO ChargingPoolID

This object describes the unique identifier of a Charging Pool. A charging pool describes one or more charging stations at the same location, provided by one EVSE operator. The EVSE operator is responsible to set the ChargingPoolId and to ensure, that the ID is unique within all of his charging pools.

Attributes	Datatype	Constraints	Description
id	string	• max. length: 255 • required	The ChargingPoolId identifies a concrete charging pool, which is unique per EVSE operator.

2119: ENUM AccessType

The Object ENUM AccessType containes the possible values for the access types of a charging point.

Values	Description
PRIVATE	Restricted to residents use only
COMPANY	Restricted to company use only
PUBLIC	Freely acccessible to the public

2014: BO ChargingStation

This object describes a Charging Station. A charging station consists of one or more charging points (EVSE) and belongs to a charging pool.

Attributes	Datatype	Constraints	Description
stationId	BO ChargingStationId	• required	Identifier of a single charging station.
chargingPoints	BO ChargingPoint	• Multiplicity: 1..*	A set of the charging points the station consists of.
paymentMethod	ENUM PaymentMethod	• Multiplicity: 0..*	Methods to pay at the station
authenticationMethod	ENUM AuthenticationMethod	• Multiplicity: 0..*	Methods to authenticate at the station
restrictions	BO MultilingualText	• max. length: 255 • Multiplicity: 0..*	Special restrictions like max. car dimensions or battery type in different languages

Attributes	Datatype	Constraints	Description
description	BO MultilingualText	• Multiplicity: 0..*	Textual description of this station
model	BO MultilingualText	• Multiplicity: 0..*	Information about manufacturer and type e.g. "Siemens CP500" in different languages
bookable	boolean	• required • default: false	True, if the charging points of this station can be reserved

2168: BO ChargingStationID

This object describes the unique identifier of a Charging Station. A charging station belongs to a charging pool and provides one or more charging points (EVSE). The EVSE operator is responsible to set the ChargingStationId and to ensure, that the ID is unique within the charging pool.

Attributes	Datatype	Constraints	Description
id	string	• max. length: 255 • required	The ChargingStationId identifies a concrete charging station, which is unique per charging pool.

2117: ENUM AuthenticationMethod

The Object ENUM AuthenticationMethod containes the values for the possible authentication methods at an EVSE. This list will need to be extended in the future as the market develops and new standards arise.

Values	Description
RFID	Via RFID card
SMS	Via sms to EVSE operator
CALL	Via call to EVSE operator
V2G_CI	Vehicle to grid communication
PARKNCH	Park & charge
NONE	None authentication necessary
APP	Via smart phone app
PLUGNCH	Via Plug&Charge

2118: ENUM PaymentMethod

The Object ENUM PaymentMethod contains the values for the possible payment methods at an EVSE. This list will need to be extended in the future as the market develops and new standards arise.

Values	Description
PNCH	Park & Charge
CREDIT	Credit card
CASH	Cash
DEBIT	Debit

Values	Description
NONE	Free charging
CONTRACT	Pay via contract

1985: BO ChargingPoint (EVSE)

The object ChargingPoint (EVSE) describes a single charging point as part of a charging station.
A charging station (EVSE) can be equipped with one or more charging points.
A charging point can be equipped with one or more plugs.

Attributes	Datatype	Constraints	Description
evseId	2127: BO EVSEID	• required	Globally unique identifier of a single EVSE (charging point)
plugs	1986: BO Plug	• Multiplicity: 1..*	Available connection plugs
EVSE max Amp	string	• optional	Maximal on EVSE available ampere
Voltage	string	• optional	Maximal on EVSE available voltage
KW	string	•	Maximal on EVSE available KW
Phase	string	• optional	Maximal on EVSE available Phase
description	BO MultilingualText	• Multiplicity: 0..*	Textual description of this CP and/or the location e.g.number "5" or "CP left to entry" ' in different languages
Bookable	Bookable	• optional	Flag, if charging point ca be reserved (=true - default) or not (false). If this parameter is obsolete the charging point may be reserved.

2127: BO EVSEID

This object describes the unique identifier of single EVSE. An EVSE can represent a Charging Point.
For the demonstration it will not be possible to fully implement one common standard for the EVSE ID, because the existing systems and infrastructure cannot be changed in this short timeframe. As a first step towards harmonization of the EVSE ID it was agreed to adopt the format specification, which is outlined in D3.9 (chapter 3.4; see http://www.greenemotion-project.eu/dissemination/deliverables-ict-solutions.php), for the first two parts of the EVSE ID (Country Code - 2 alpha, two character country code according to ISO-3166-1 (Alpha-2-Code), Spot Operator ID - 3 alpha / digits). The existing EVSE IDs will have to be mapped to the common format by the EVSE operator backend.
In the end all involved partners will strive for one common standard. The format that is currently discussed is outlined in D3.9 (chapter 3.4) and was submitted for standardization by RWE.
The Spot Operator IDs will be managed by the consortium during the demonstration phase. Afterwards, it is mandatory to create one authority or a system of authorities which manage the Provider IDs as well as the Spot Operator IDs.

Attributes	Datatype	Constraints	Description
Id	string	• max. length: 255 • required • <EVSEID>=<Country Code> "*" <Spot Operator ID> "*" <Power Outlet ID>	The EVSEID identifies a concrete charging point. If a charging station has multiple charging points, multiple EVSEIDs have to be used.

2110: ENUM StatusType

The Object ENUM StatusType contains the possible values for the state of an EVSE.

Values	Description
Charging	EVSE is charging
Free	EVSE is free
Offline	EVSE is offline
OutOfOrder	EVSE is out of order
Reserved	EVSE is reserved
Planned	EVSE is in planning
Unknown	EVSE status is unknown

1989: BO EVSEStatus

This object describes the current status of a EVSE (Charging Point or Battery Switch Station).

Attributes	Datatype	Constraints	Description
evseId	BO EVSEId	• required	Identifier of the related EVSE
status	ENUM StatusType	• required	Information about the current status of the EVSE

1986: BO Plug

This object describes a single plug of a BO ChargingPoint. Each charging point can have one or more plugs, but only one plug can be used per time.

Attributes	Datatype	Constraints	Description
plugType	ENUM PlugType	• required	Type of the plug
maxAmpere	double	• optional	Maximum load in Ampere
chargingMode	ENUM ChargingMode	• optional	Mode of charging as defined in IEC 61851

2111: ENUM PlugType

The Object ENUM PlugType contains the possible values for the type of a charging point plug. This list will need to be extended in the future as the market develops and new standards arise.

Values	Description
Household	Household connection is available, CEE 7/4, Shuko, 230V16A (CEE_7_4)
Type_1	SAE J1772, 5 pin, Type1 in IEC62196
Type_2	VDE-AR-E 2623-2-2 (Mennekes), 7 pin, Type 2 IEC62196
Type_3	EV plug alliance, Type 3 in IEC62196
Combo_Type_2	Fast charging, DC & AC
CHAdeMO	Fast charging, DC
Camping	Camping plug, 230V/16A, 3 pin
CEE_3_Phase	CEE, 400V/16A, 5 pin
CEE_7_4	CEE 7/4, Shuko, 230V/16A
CEE_7_7	CEE 7/7, 230V/16A
Inductive	Inductive charging

2114: ENUM ChargingMode

The Object ENUM ChargingMode containes the possible values for the modes of charging provided via a charging point plug. This list will need to be extended in the future as the market develops and new standards arise.

Values	Description
Mode_1	Mode 1 defined in IEC 62196
Mode_2	Mode 2 defined in IEC 62196
Mode_3	Mode 3 defined in IEC 62196
Mode_4	Mode 4 defined in IEC 62196
CHAdeMO	In addition to Mode_4
Inductive	Inductive charging, not yet standardized
AC	Alternating current
DC	Direct current

2374: ENUM TypeOfCharge

The enumeration ENUM TypeOfCharge contains the possible values used in the roaming domain.

Values	Description
AC	Alternate Current
DC	Direct Current

2015: BO BatterySwitchStation

This object describes a single Battery Switch Station, which represents an EVSE.

Attributes	Datatype	Constraints	Description
evseId	BO EVSEId	• required	Globally unique identifier of a single EVSE (BatterySwitchStation)
evseOperator	BO EVSEOperator	• required	Identifies the operator that operates this BatterySwitchStation
location	BO Location	• required	Address of the BatterySwitchStation
geoCoordinates	BO GeoCoordinates	• optional	Geo coordinates of the BatterySwitchStation
status	ENUM StatusType	• optional	Information about the current status of the BatterySwitchStation
openHours	BO MultilingualText	• Multiplicity: 0..*	General information about opening hours and days in different languages
restrictions	BO MultilingualText	• max. length: 255 • Multiplicity: 0..*	Special restrictions like max. car dimensions or battery type in different languages
model	BO MultilingualText	• Multiplicity: 0..*	Information about manufacturer and type in different languages

Common Business Objects

This domain model includes business objects that are not specific to a certain functional domain but should be used throughout the project in order to enhance the interoperability of service interfaces by using the same data types for common elements.

2389: BO EVSPID

This object describes the unique identifier of an EVSP.
The EVSPID consists of the first two parts of the first part of the EVCOID (Country Code - 2 alpha, two character country code according to ISO-3166-1 (Alpha-2-Code), EVSPID - 3 alpha / digits).
The EVSPIDs will be managed by the consortium during the demonstration phase. Afterwards, it is mandatory to create one authority or a system of authorities which manage the Provider IDs.

Attributes	Datatype	Constraints	Description
Country Code	2 alpha	• required	Country code according to ISO-3166-1
Provider-ID	3 alpha / digits	• required	Unique EVSP provider-ID with the prefix country-code

2115: BO MultilingualText

Represents a human readable text in a specific language. To support multiple languages an attribute of a BO can reference this BO as data type with the multiplicity 1..* or *

Attributes	Datatype	Constraints	Description
language	string	IETF language tag (defined in RFC 5645 and RFC 4647)	Language of the text
text	string		Any text in the defined language

2165: BO EVCOID

The EVCOID describes the unique identifier of the contract between an EVSP and an EV customer.
For the demonstration it will not be possible to fully implement one common standard for the contract ID, because the existing systems and infrastructure cannot be changed in this short timeframe. As a first step towards harmonization of the EVCOID it was agreed to adopt the format specification, which is outlined in D3.9 (chapter 3.4), for the first two parts of the contract ID (Country Code - 2 alpha digits, Provider ID - 3 alphanumeric digits). In cases where the Provider ID of existing EVCOIDs exceeds three digits, only the first three digits will be used and the remaining digits will be ignored. The format of the customer ID that is contained in the EVCOID will not be harmonized for the demonstration.
In the end all involved partners will strive for one common standard. The format that is currently discussed is compliant to the ISO IEC 15118. The Provider IDs will be managed by the consortium during the demonstration phase. Afterwards, it is mandatory to create one authority or a system of authorities which manage the Provider IDs as well as the Spot Operator IDs.
Schema:
<EVCOID> = <Country Code> <S> <Provider ID> <S> <CustomerID> <S> <Check Digit>
<Country Code> = 2 ALPHA ; two character country code according to ISO 3166-1 (Alpha-2-Code)
<Provider ID> = 3 (ALPHA / DIGIT) ; three alphanumeric characters, defined and listed by eMI3 group
<CustomerID> = 9 (ALPHA / DIGIT) ; nine alphanumeric characters
<Check Digit> = *1 (ALPHA / DIGIT) ; Optional
ALPHA = %x41-5A / %x61-7A ; according to IETF RFC 5234(7-Bit ASCII)
DIGIT = %x30-39 ; according to IETF RFC 5234 (7-Bit ASCII)
<S> = *1 (“-”) ; optional separator

Attributes	Datatype	Constraints	Description
id	string	cardinality: 1 max. length: 255	Contract ID which contains EVSP ID and Customer ID; case insensitive **Example**: DE0041T34CV780

3.2 Service Specification

1502: UC EV Identification, Authentication and Authorization

This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 1512: UC Start a roaming charging process with Clearinghouse

1510: UC Before charging

This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 1527: UC Search for EVSE
- 1528: UC Reservation of EVSE

- 2364: UC Routing to EVSE

1518: UC During charging

This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 1512: UC Start a roaming charging process with Clearinghouse

1519: UC After charging

This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 1524: UC Fleet manager monitors energy consumption of pool-cars
- 1561: UC Calculate CO2 Emission
- 1562: UC Report Electricity Consumption
- 1511: UC End a roaming charging process with Clearinghouse

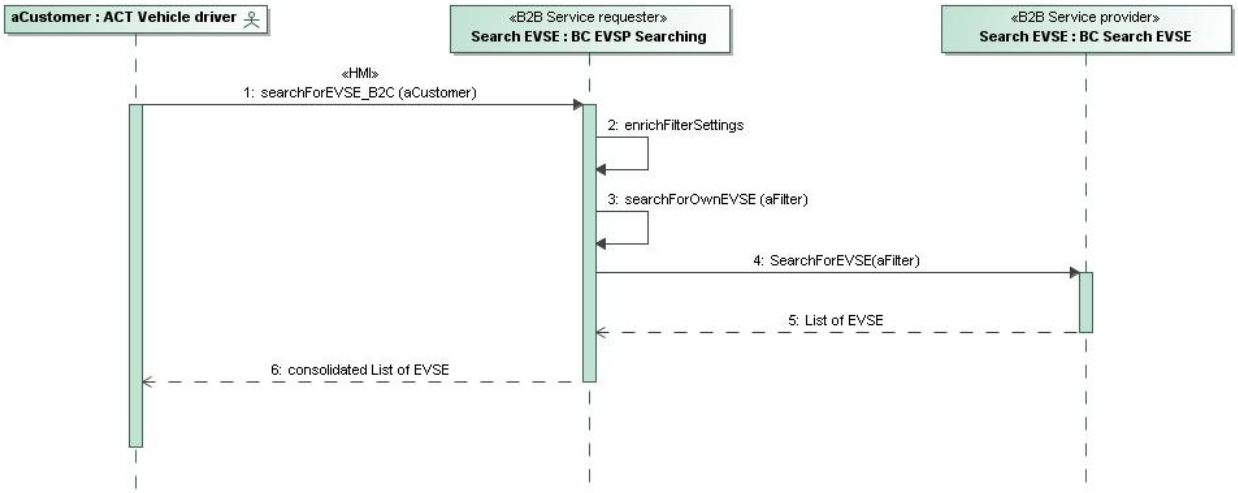
1527: UC Search for EVSE

Sequence Diagrams	2021: SQD Search EVSE 2047: SQD EVSE Base Data 2049: SQD EVSE Status 2066: SQD EVSE Push Status
Business Components	2018: BC Search EVSE 1958: BC Charging management system (CMS) 2121: BC EVSP Searching 2367: BC Reservation EVSE
Services	2012: SRV SearchForEVSE 2045: SRV ProvideEVSEMasterData 2046: SRV ProvideEVSEStatusData 2065: SRV PushEVSEStatus 2129: SRV InsertEVSE 2131: SRV DeleteEVSE 2132: SRV UpdateEVSE 2370: SRV Search EVSE Advertising data

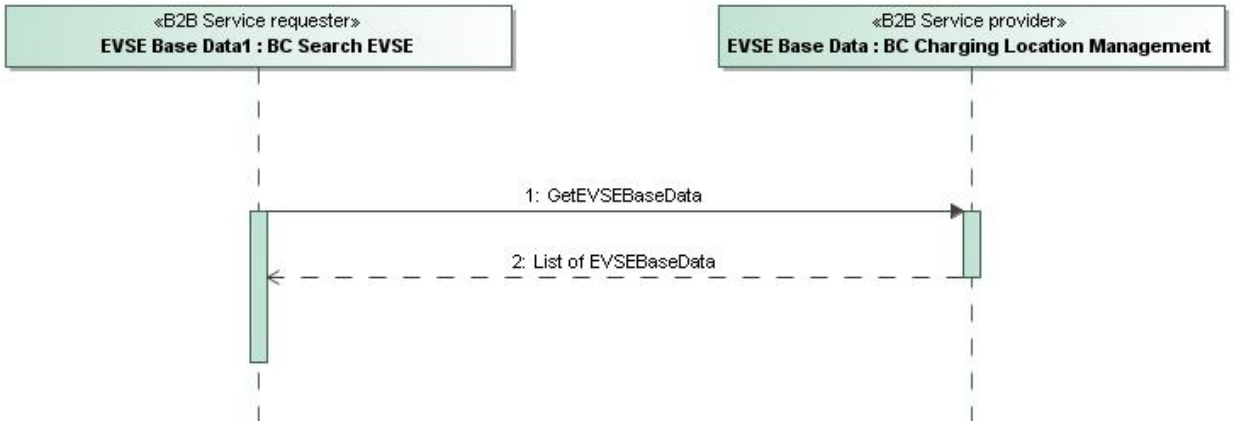
This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 1574: UC Third Party Information

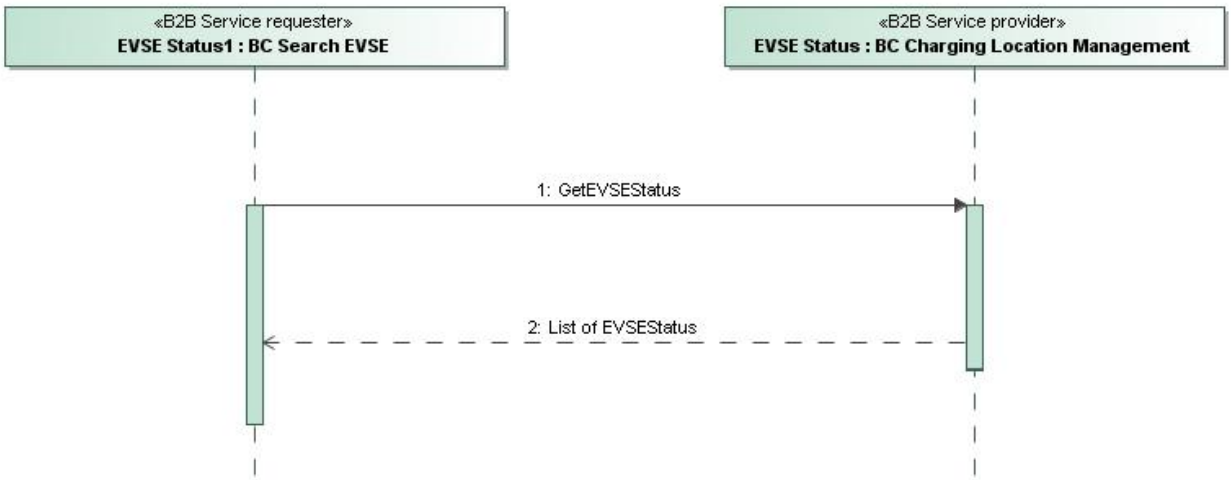
2021: SQD Search EVSE



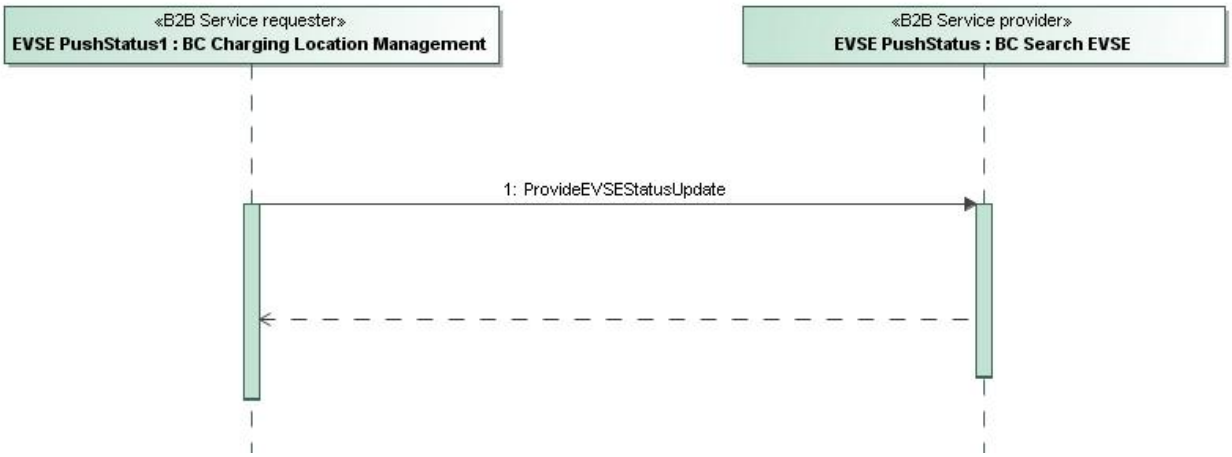
2047: SQD EVSE Base Data



2049: SQD EVSE Status



2066: SQD EVSE Push Status



2018: BC Search EVSE

The BC Search EVSE provides the functionality to search for EVSEs based on several search criteria.
A search request will be initiated by a contracted EVSP.
The BC Search EVSE internally stores a set of EVSE-s including base and status data, which can be updated in the following ways:

- a CMS operating the EVSE uploads base or status data changes via the provided interfaces of BC Search EVSE
- the BC Search EVSE requests data updates from CMS in a defined time interval

This service optionally returns advertising data related to the EVSE geographical (e.g. geo-coordinates, address) search data.
For this service it is necessary to integrate a third-party service which will provide the advertising data.
The EVSE-search service will then aggregate the result of the available EVSE data with the related advertising data and send this information to the requested EVSP-backend.
Alternatively it could also be possible that a stand-alone B2B service for advertising data will be offered by the marketplace and the aggregation will be done by the EVSP-backend.
It is not planned to realize advertising portion of the service / interface during GeM projectphase

Provided Services	2012: SRV SearchForEVSE 2065: SRV PushEVSEStatus 2129: SRV InsertEVSE 2131: SRV DeleteEVSE 2132: SRV UpdateEVSE
Required Services	2045: SRV ProvideEVSEMasterData 2046: SRV ProvideEVSEStatusData 2370: SRV Search EVSE Advertising data

2012: SRV SearchForEVSE

This service accepts a EVSEsearch object with a set of optional search criteria and returns a search result class with lists of all EVSE-s with different types (CP, BatterySwitchStation), which match the given search criteria. If a search criterion is not set, it will be ignored.
The return of an empty lists means, that none EVSE of that type is available.
.

SearchForEVSERequest

Attributes	Datatype	Constraints	Description
searchCriteria	1993: BO EVSESearch	Multiplicity: 1	The search criteria which will be used to refine the EVSE search results.

SearchForEVSEResponse

Attributes	Datatype	Constraints	Description
searchResult	2106: BO EVSECollection	Mulitiplicity:1	The search results contains lists of EVSEs with different types (CP, BatterySwitchStation) which match the given filter criteria.
Referenced by BO EVSECollection	2167: BO ChargingPool	Multiplicity: 0..*	List with found ChargingPool-s where at least one charging station meets the search criteria
Referenced by BO ChargingPool	2014: BO ChargingStation	Multiplicity: 1..*	List of charging stations of a pool, where at least one charging point meets the search criteria
Referenced by BO ChargingStation	1985: BO ChargingPoint	Multiplicity: 1..*	List of available charging points of a charging station
Referenced by BO ChargingPoint	1986: BO Plug	Muliplicity: 1..*	Available connection plugs at the CP

Attributes	Datatype	Constraints	Description
Referenced by BO EVSECollection	2015: BO BatterySwitchStation	Multiplicity: 0..*	List with found BatterySwitchStations-s

Service specific Business Objects (Not part of Domain Model)

1993: BO Search EVSE

This business object BO Search EVSE describes the search criteria of an EVSE search request.

Attributes	Datatype	Constraints	Description
critEVSEType	enum	• values: "CP", "BSS" • optional	Search criteria, if search for CP or battery switch station only
critEVSEOperatorId	BO EVSEOperatorId	• multiplicity 0..*	Search list with ID-s of EVSE operators to search for
critEVSEOperatorName	string	• max. length: 255 • optional	Search criteria for a single EVSE operator name (see BO EVSEOperator name)
critMaxAmpere	double	• optional	Search criteria for the max. allowed load. EVSE has to provide a maximum load equal to the filter value or more.
critChargingMode	ENUM ChargingMode	• optional	Search criteria for charging mode
critPlugType	ENUM PlugType	• optional	Search criteria for plug type
critAccessType	ENUM AccessType	• optional	Search criteria for the general access type of the CP
critPaymentMethod	ENUM PaymentMethod	• optional	Search criteria for payment methods at the EVSE
critBookable	boolean	• optional	True, if search only on bookable EVSEs.
critGeoCoordinates	BO GeoCoordinates	• optional	Search criteria for search based on geo position. Requires the setting of the attribute critRadius.
critRadius	integer	• optional	Search radius in km for the proximity search based on geo position. Proximity search is only possible if the attribute critGeoCoordinates is also set.

Attributes	Datatype	Constraints	Description
critCountry	string	• max. length: 3 • optional	Search criteria for country
critCity	string	• max. length:50 • optional	Search criteria for city
critZipCode	string	• max. length:10 • optional	Search criteria for zip code
critRegion	string	• max. length:50 • optional	Search criteria for region
critStreet	string	• max. length:100 • optional	Search criteria for street
critStatus	ENUM StatusType	• optional	Search criteria for current status of EVSE
critAuthenticationMethod	ENUM AuthenticationMethod	• optional	Search criteria for authentication method

2106: BO EVSECollection

The object EVSECollection acts as a wrapper object for the EVSEs and embeds a list for charging pools with available charging points and a list for available battery switch stations.

Attributes	Datatype	Constraints	Description
chargingPools	BO ChargingPool	• Multiplicity: 0..*	List of charging pools with available charging points
batterySwitchStations	BO BatterySwitchStation	• Multiplicity: 0..*	List of available battery switch stations

2065: SRV PushEVSEStatus

SRV PushEVSEStatus will be provided by a BC which deals with status information of EVSE-s (e.g. BC EVSESearch). The BC will request base data from the the CMS (2045: SRV ProvideEVSEMasterData) and may request status data regularly. But to enhance the timeliness of provided status data, the CMS may be obligated to push the information about each status change in a certain time interval to the BC. Therefore this service is embedded in the Service 2046: SRV ProvideEVSEStatusData.
Each subscribed CMS has to call this service after each status change in a configured time interval (e.g. once per minute).
If none of the connected EVSE-s changed the status since the last status change was propagated by the CMS, the CMS does not have to call this service.

The CMS has to call the services of all BC, with whom an enhanced EVSEStatus contract exists.
Note: The Market Place has to provide the possibility to initiate data transfer within the same contract not only from the service requester to the service provider, but also vice versa.

EVSEPushStatusRequest

Attributes	Datatype	Constraints	Description
evseOperatorId	2130: BO EVSEOperatorId	• Multiplicity: 1	Unique identifier of the CMS or EVSE operator
status	1989: BO EVSEStatus	• Multiplicity: 0..*	A list of all EVSE status data from the providing CMS, which changed since the last push of status data

2129: SRV InsertEVSE

This service enables a CMS component to insert new EVSE-s into the EVSE data storage of a central service component like BC EVSE Search.

InsertEVSERequest

evseOperatorId	2130: BO EVSEOperatorId	• Multiplicity: 1	Unique identifier of the CMS or EVSE operator
insert	2106: BO EVSECollection	• Multiplicity: 1	An EVSE collection object with lists of new EVSE-s to be inserted into the data storage of the central service component.

InsertEVSEResponse

Attributes	Datatype	Constraints	Description
insertErrors	2128: BO EVSEUpdateError	• Multiplicity: 0..*	A list with information about failed insert operations. An empty list means that all EVSE-s were inserted successfully.

Service specific Business Objects (Not part of Domain Model)

2128: BO EVSEUpdateError

This object describes a problem which occurred while trying to insert, update or delete a single EVSE within the data storage of a service component like BC Search EVSE.
Possible error scenarios are e.g.:

- a requester tried to delete an EVSE which does not exist before
- a requester tried to insert an EVSE which already exists before

Attributes	Datatype	Constraints	Description
evseId	BO EVSEId	• required	Globally unique identifier of a single EVSE
code	integer	• required	Code identifying the occurred problem (a list of possible error codes will be provided during implementation)

Attributes	Datatype	Constraints	Description
description	string	• optional • max. length: 255	Description of the error

2131: SRV DeleteEVSE

This service enables a CMS component to delete EVSE-s from the EVSE data storage of a central service component like BC EVSE Search.

deleteEVSERequest

Attributes	Datatype	Constraints	Description
evseOperatorId	2130: BO EVSEOperatorId	• Multiplicity: 1	Unique identifier of the CMS or EVSE operator
delete	2127: BO EVSEId	• Multiplicity: 1..*	A list of EVSE Id-s to be deleted from the data storage of the central service component.

deleteEVSEResponse

Attributes	Datatype	Constraints	Description
deleteErrors	2128: BO EVSEUpdateError	• Multiplicity: 0..*	A list with information about failed delete operations. An empty list means that all EVSE-s were deleted successfully.

Service specific Business Objects (Not part of Domain Model)

2132: SRV UpdateEVSE

This service enables a CMS component to update the base data of changed EVSE-s within the EVSE data storage of a central service component like BC EVSE Search.

updateEVSERequest

Attributes	Datatype	Constraints	Description
evseOperatorId	2130: BO EVSEOperatorId	• Multiplicity: 1	Unique identifier of the CMS or EVSE operator
update	2106: BO EVSECollection	• Multiplicity: 1	An EVSE collection object with lists of EVSE-s to be updated within the data storage of the central service component.

updateEVSEResponse

Attributes	Datatype	Constraints	Description
updateErrors	2128: BO EVSEUpdateError	• Multiplicity: 0..*	A list with information about failed update operations. An empty list means that all EVSE-s were updated successfully.

Service specific Business Objects (Not part of Domain Model)

1958: BC Charging management system (CMS)

The charging management system (CMS) is a IT-applikation to operate the Electric Vehicle supplier equipment (EVSE) in the GeM. It represent finding, maintaining and controlling a certain EVSE. It also helps transferring the final load information called the service detail record (SDR) from the EVSE to a European EVSP.

Provided Services	2045: SRV ProvideEVSEMasterData 2046: SRV ProvideEVSEStatusData 2054: SRV CLM ActualChargingStatus 2053: SRV Reservation to CMS
Required Services	2065: SRV PushEVSEStatus 2020: SRV Authorization 2032: SRV SDR Forwarding 2129: SRV InsertEVSE 2131: SRV DeleteEVSE 2132: SRV UpdateEVSE

2045: SRV ProvideEVSEMasterData

This service returns the base data of all EVSE-s which will be operated by EVSE Operators connected to the CMS. Information about the current status of EVSE-s will be provided by an additional service SRV ProvideEVSEStatusData.
The ProvideEVSEMasterData service may be used by a EVSESearch component to provide EVSE data from different CMS and should be called in a configured interval (e.g. once per week).

ProvideEVSEBaseDataRequest

Attributes	Datatype	Constraints	Description
filterTimestamp	dateTime	Multiplicity: 0..1	If filterTimestamp ist set, the CMS returns only the base data of those EVSE-s, where the base data have changed since that point in time. Otherwise the base data of all EVSE-s will be returned.

ProvideEVSEBaseDataResponse

Attributes	Datatype	Constraints	Description
searchResult	2106: BO EVSECollection	Mulitiplicity:1	The eveseCollection contains lists of EVSEs, which will be operated by the called CMS. None status information will be provided.
Referenced by BO EVSECollection	2167: BO ChargingPool	Multiplicity: 0..*	List with found ChargingPool-s operated by the CMS
Referenced by BO ChargingPool	2014: BO ChargingStation	Multiplicity: 1..*	List of charging stations of a pool
Referenced by BO ChargingStation	1985: BO ChargingPoint	Multiplicity: 1..*	List of charging points of a charging station

Attributes	Datatype	Constraints	Description
Referenced by BO ChargingPoint	1986: BO Plug	Muliplicity: 1..*	Connection plugs at the CP
Referenced by BO EVSECollection	2015: BO BatterySwitchStation	Multiplicity: 0..*	List with BatterySwitchStations-s

Service specific Business Objects (Not part of Domain Model)

2046: SRV ProvideEVSEStatusData

This service returns the current status data of all EVSE-s which will be operated by EVSE Operators connected to the CMS. Information about the base data of EVSE-s will be provided by an additional service SRV ProvideEVSEMasterData. Due to performance reasons the timestamp of the latest status update can be used to reduce the number of returned values.
The ProvideEVSEStatusData service may be used by a EVSESearch component to provide current status EVSE data from different CMS and should be called in a configured interval (e.g. once per 15 minutes) or on request.

ProvideEVSEStatusDataRequest

Attributes	Datatype	Constraints	Description
filterTimestamp	dateTime	• optional	If filterTimestamp ist set, the CMS returns only the status of those EVSE-s, where the status has changed since that point in time.
filterEVSEId	BO EVSEId	• Multiplicity: 0..*	A list of EVSEId-s for which current status information are requested. If the list is empty, the status information for all EVSE-s will be requested.

ProvideEVSEStatusDataResponse

Attributes	Datatype	Constraints	Description
status	1989: EVSEStatus	• Mulitiplicity: 0..*	A list of all EVSE status data, which will be operated by the called CMS and match the filter criteria. If a requested EVSE does not exist, the filter request will be ignored. An empty list may indicate, that none status has changed since the latest status request.

2054: SRV CLM ActualChargingStatus

This service interface serves for the description of the load data to be delivered to the requestor

ActualChargingStatusRequest

Attributes	Datatype	Constraints	Description
Timestamp	dateTime	• required	Timestamp, when the message was sent

Attributes	Datatype	Constraints	Description
ReqEVSEID	BO EVSEId	• required	defines the actual charging status of a certain charging point to be delivered

ActualChargingStatusResponse

Attributes	Datatype	Constraints	Description
Timestamp	dateTime	• required	Timestamp, when the message was sent
ActualChargingstatus	2040: BO PowerOutletStatus	• required	transmitted data field to the requestor; returns "0" if unknown

Service specific Business Objects (Not part of Domain Model)

2040: BO PowerOutletStatus

This object describes all actual current status of a power outlet point like Charging Point or Battery Switch Station.

Attributes	Datatype	Constraints	Description
EVCOID	2165: BO EVCOID	• required	Contract ID which contains EVSP ID and Customer ID
EVSEBasicStatus	1989: BO EVSEstatus	• required	Identifies the poweroutletpoint

2053: SRV Reservation to CMS

This SRV will send the service-request for reservation / booking of an EVSE to the CMS.
Each time a CMS wants to connect to the Reservation-Service, a system availability check is recommended in order to check whether or not the reservation module is able to receive and process update messages. The Reservation-Service verifies its availability with a positive response.
In the following the SRV for reservation of EVSE to CMS and canceling of reservation is described.
In the first step this SRV can be demonstrated with the SIEMENS CMS.

ReservationForEVSERequest

Attributes	Datatype	Constraints	Description
EVSE	2127: BO EVSEID	1..*	Unique identifier of the EVSEs which should be reserved
EVCO-ID	2165: BO EVCOID	1..*	One single EVCO-ID or a list of EVCO-ID which are allowed to use this reservation / EVSE for charging
End Time	2083: BO EVSEReservation	1	End time of reservation. At this time the CMS has to send a Cancel request to the EVSE if it is not already charging. UTC time is required.

Service specific Business Objects (Not part of Domain Model)

2083: BO Reservation EVSE

This objects contains information about a required reservation from an EVSP (end-user) to an EVSE-operator.
The information will be used for manage the reservation-process in will be stored in the Reservation-Service.

Attributes	Datatype	Constraints	Description
Reservation-ID	ID	0..*	Unique reservation ID as generated by EVSP. If this ID is filled from the EVSP, then the Reservation service will use this ID for all further processes. If not, the Reservation service will generat a unique ID instead.
Charing-Pool-ID	2167: BO ChargingPoolID	• required	Globally unique identifier of a charging pool, or EVSE if charging pool is not used by the EVSE-operator
EVCO-ID	2165: BO EVCOID	1..*	Single EVCO-ID or list of EVCO-ID of the end-user who will use this reservation for charging
TimeRange	dateTime	• required	Begin of reservation, based on ISO 8601. Start date End date
Reservationstatus	Status	1	Current reservation status
ReservationDate	DateTime	1	The date the reservation was booked
ReservationType	ReservationType	1	The type of reservation: - Point in time - Period

The information will be used to configure the reservation-process per EVSP.

Attributes	Datatype	Constraints	Description
Max Reservations	int	optional	Max allowed active reservation for a end-user (EV-driver)
Max Reservation time/days	int	optional	Max reservation time
Start booking	int	optional	Allowed start-time/days
Reservation time	int	optional	Max reservation time (duration)
Reservation Type	Reservation Type	1	The type of reservation ca be configured: • Point in time • Period

2121: BC EVSP Searching

The BC EVSE Search provides the functionality to search for EVSEs to EV drivers.
Through its backend it requests data from BC SearchEVSE connected via the market place.

Through its frontend the search results can be presented to the requesting vehicle driver as text-table or in a map.
The vehicle driver can access the search frontend via one of the following user interfaces:

- Smart phone application (e.g. iPhone app)
- Internet customer portal
- In car application

This business component can be implemented several times and not all of them need to offer all user interfaces. The look and feel of the individual implementation depends on designers choice and limitations of the device.

Provided Services	
Required Services	2012: SRV SearchForEVSE

2367: BC Reservation EVSE

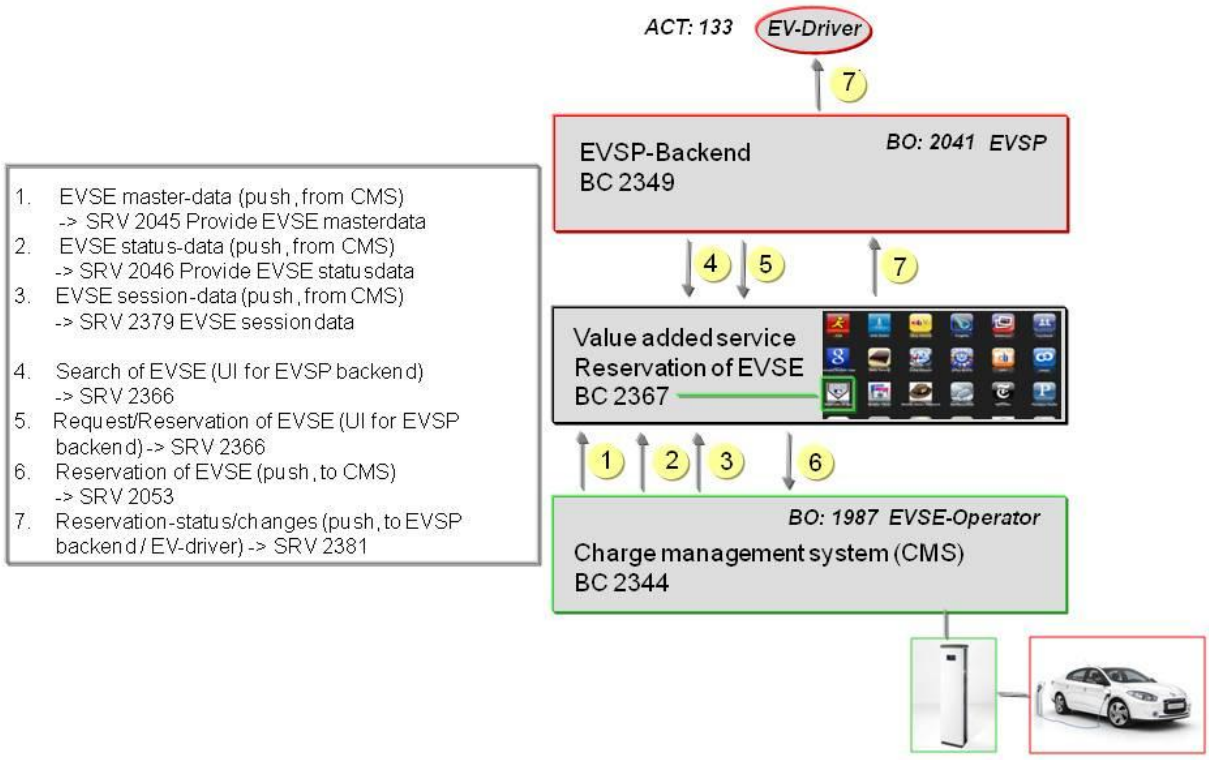
The BC Reservation EVSE provides the functionality to book EVSEs based on several search criteria.
A reservation request will be initiated by a contracted EVSP.
The BC reservation EVSE internally stores a set of EVSE-s including base and status data, which can be updated in the following ways:

- a CMS operating the EVSE uploads base or status data changes via the provided interfaces of BC Search EVSE
- the BC Search EVSE requests data updates from CLMs in a defined time interval

The service reservation of EVSE can send the service-request for booking of an EVSE.
The EVSP end-user can perform reservations via the web UI (User-Interface) for reservation. Those reservation can be sent to the relevant CMS at
a specified time in order to set the according EVSEs in a "Reserved" state.
The charging pools and the EVSE are maintained in Charging Management Systems (CMS).
For the reservation functionalities also the interface for EVSE masterdata and statusdata have to be implemented from the EVSE-operator in the
relevant CMS.
Each time a CMS wants to connect to the Reservation-Service, a system availability check is recommended in order to check whether or not the reservation module is able to receive and process update messages. The Reservation-Service verifies its availability with a positive response.
The connected CMS can use the interfaces for EVSE-masterdata for synchronizing its complete set of EVSE with the Reservation-Service.
New EVSE are inserted, modified EVSE are updated and excess EVSE are removed. The update should be done each time a EVSE master data information change.
In addition to these masterdata activities, the CMS can use the interface for EVSE status-data to update the current status of a single EVSE or even of a list of EVSEs.
These actual EVSE status are significant for EVSP end-users searching for EVSEs as well as for reservation purposes. So it is essential for the connected CMS to keep the Reservation-Service up to date about any status changes on any EVSE. This should be done immediately.
Each time an EVSP end-user starts charging, a session is created in the CMS. The session is kept up to date frequently until the driver stops charging and unplug his car from the EVSE. The CMS has to keep the Reservation-Service up to date about the session status as the used by the Reservation service for Reservation handling. Therefore an interface is provided for updating session information on Reservation-Service. As session information are essential for reservation handling, the CMS should update session information as soon and frequent as possible.
While the CMS feeds the Reservation-Service with EVSE master, status and session data, the Reservation-Service has to send the request for reservation to the relevant CMS and the CMS has to send the request for reservation to the EVSE.

In the following the SRV for request for reservation and canceling of reservation is described.
This SRV can be demonstrated with the SIEMENS CMS.

D3.6
BC 2367 (Reservation) relevant SRV



Provided Services	2129: SRV InsertEVSE 2132: SRV UpdateEVSE 2065: SRV PushEVSEStatus 2012: SRV SearchForEVSE 2379: SRV Reservation - Chargingsession data
Required Services	2045: SRV ProvideEVSEMasterData 2046: SRV ProvideEVSEStatusData 2366: SRV Reservation request from EVSP 2381: SRV Reservation status/changes 2053: SRV Reservation to CMS

2379: SRV Reservation - Chargingsession data

This SRV specifies a single session at a EVSE. This chargingsession is created when a end-user authenticates at a EVSE and plugs his EV.
It ends when the vehicle is unplugged.

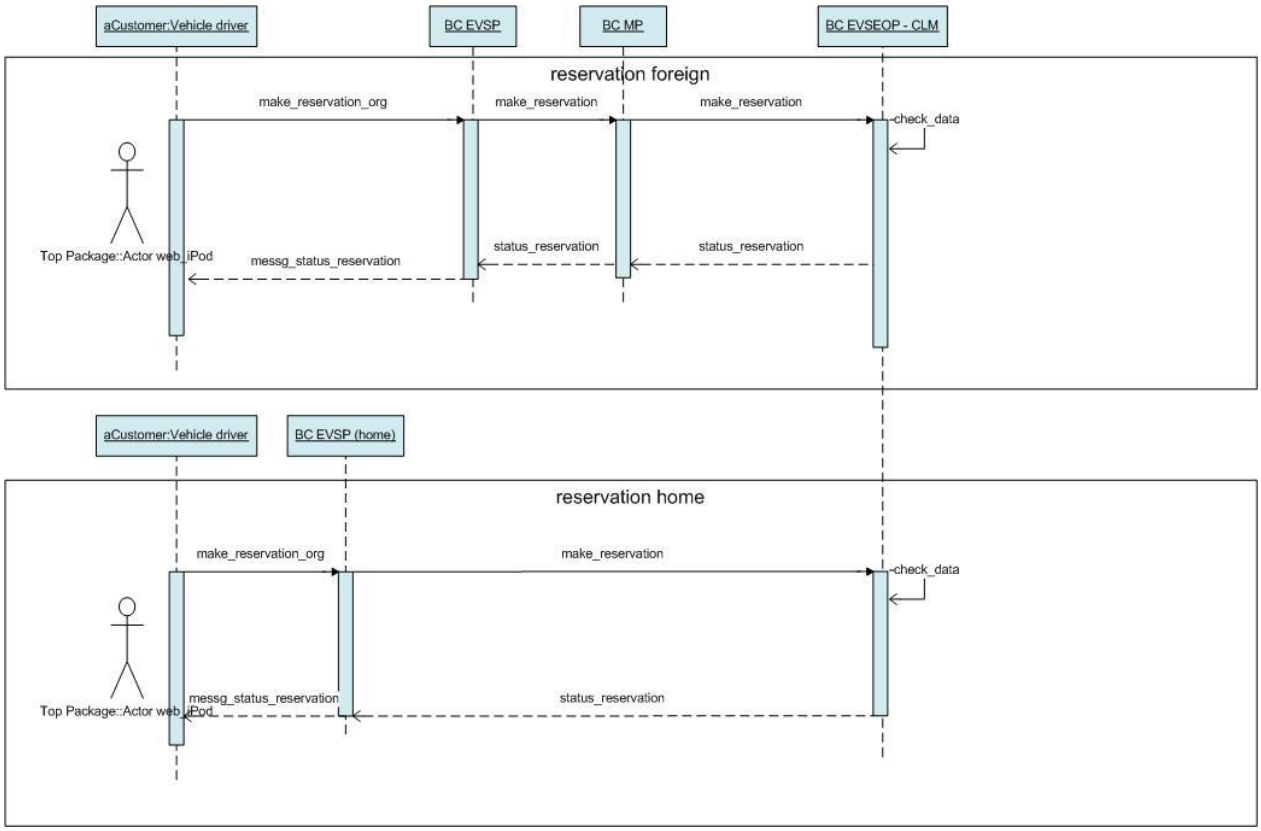
Session data provides actual information about such session including the actual charging-session user (EVCO-ID) and are necessary to support the reservation process.

Parameter	Datatype	Constraints	Description
MessageType	MessageType	1	The concrete session activity describing if the session is just started, ended or still in use
EVSE-ID	2127: BO EVSEID	1	The EVSE at which the EV is connected and charging
SessionId	string	1	will be created by EVSE operator backend system (just unique in one EVSE Operator system)
EVCO-ID	2165: BO EVCOID	1	EVCO-ID which is used for the charging session
MeterAbsolute	MeterAbsolute	1	Current status of the calibrated energy meter
MetercurrentSessio n	MeterCurentSession	1	Active energy consumption for the current sessoin (Kwh). The value is "0.000000" for type = START
FinishedNormally	FinishedNormally	1	Flag specifying whether or not the session is finished by normal unplug and not by an error. True: normal end False: session abort, this charge session should be ignored (default)
Timestamp	DateTime	1	Date and time when data were last modified, UTC time

1528: UC Reservation of EVSE

Sequence Diagrams	2094: SQD CLM Reservation
Services	2053: SRV Reservation to CMS 2366: SRV Reservation request from EVSP 2379: SRV Reservation - Chargingsession data 2381: SRV Reservation status/changes
Business Components	2367: BC Reservation EVSE

2094: SQD_CLM_Reservation



1529: UC Charging Location Management

Business Components	1958: BC Charging management system (CMS)
Services	2054: SRV CLM ActualChargingStatus

This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 1557: UC Parking Space Management

1958: BC Charging management system (CMS)

This BC was exported before.

2255: UC Open access

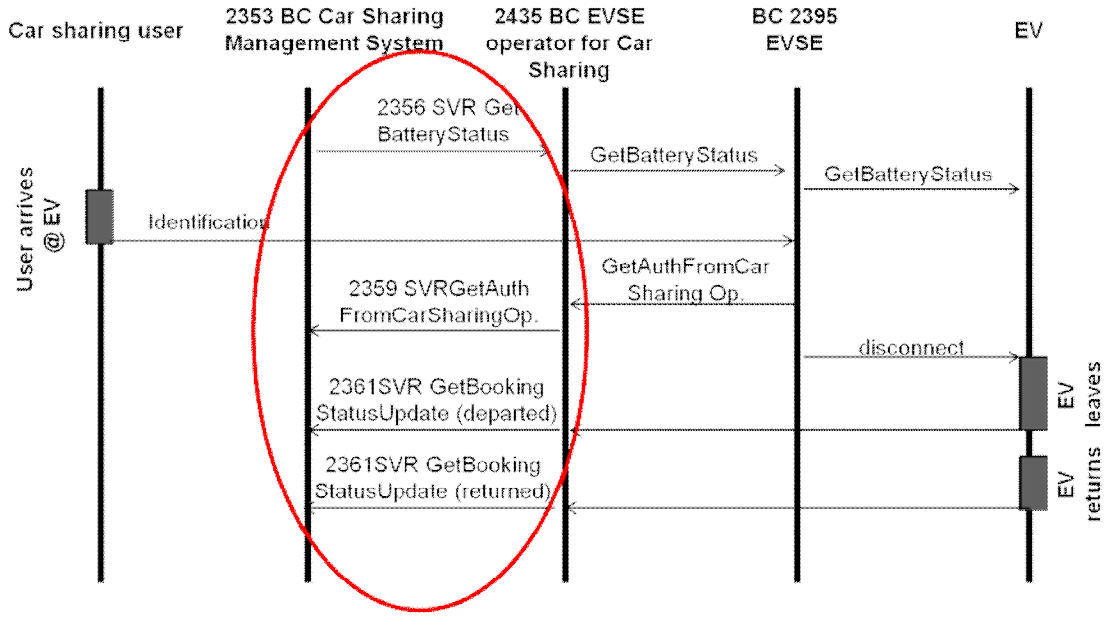
This Use Case is further specified by the following embedded Use Cases. The specification for these Use Cases is not listed in this chapter.

- 2362: UC Access without service contract

2287: UC Integration of car sharing

Sequence Diagrams	2429: SQD Integration of car sharing
Business Components	2353: BC Car Sharing Management System 2435: BC EVSE Operator for Car Sharing
Services	2356: SRV GetBatteryStatus 2359: SRV GetAuthFromCarSharingOp 2361: SRV CarSharingBookingStatusUpdate

2429: SQD Integration of car sharing



2353: BC Car Sharing Management System

The CSMS is the backend system of a Car Sharing Operator. It allows:
- integration with the EVSE operator back-end in order to retrieve information that are present in the EVSE operator backend as Recharge Cable Status, or State of Recharge, which are needed to the CSMS, to optimize the booking algorithm and maximize the frequency of access to the service.
- Managing a unique RFID Card to access the EV and to the EVSE to start/end recharge operations, which should improve and simplify the usabilty of the service.

Provided Services	2359: SRV GetAuthFromCarSharingOp

Provided Services	2359: SRV GetAuthFromCarSharingOp
Required Services	2356: SRV GetBatteryStatus 2361: SRV CarSharingBookingStatusUpdate

2359: SRV GetAuthFromCarSharingOp

This service is required by the EVSE operator to the Car Sharing operator in order to get authorization for the booking of the EV and unlock the cable from the EVSE socket

GetAuthFromCarSharingOpRequest

Attributes	Datatype	Constraints	Description
EVSE ID	2127: BO EVSEID	mandatory/MULTIPLICITY: 1	unique EVSE identification number.
EVCOID	2165: BO EVCOID	mandatory/MULTIPLICITY : 1	unique EV user contract identification number.
VIN	2360: BO VIN	mandatory/MULTIPLICITY: 1	unique EV identification number.

GetAuthFromCarSharingOpResponse

Attributes	Datatype	Constraints	Description
EVSE ID	2127: BO EVSEID	mandatory/MULTIPLICITY: 1	unique EVSE identification number
EVCOID	2165: BO EVCOID	mandatory/MULTIPLICITY : 1	unique EV user contract identification number.
VIN	2360: BO VIN	mandatory/MULTIPLICITY: 1	unique EV identification number
IsAuthorized	boolean	mandatory/MULTIPLICITY: 1	values: 1 for successfull auth; 0 for Auth refusal

Service specific Business Objects (Not part of Domain Model)

2360: BO VIN

This BO provides a unique identification for the EV

Attributes	Datatype	Constraints	Description
id	string	• Mandatory	The VIN identifies a vehicle

2435: BC EVSE Operator for Car Sharing

The BC describes the ICT capabilities of the EVSE operator to provide Car Sharing services.

Provided Services	2361: SRV CarSharingBookingStatusUpdate 2356: SRV GetBatteryStatus
Required Services	2359: SRV GetAuthFromCarSharingOp

2361: SRV CarSharingBookingStatusUpdate

This service is required by the EVSE operator to the Car Sharing operator in order to update the later wheter the EV has departed or arrived based respectively on the connection and disconnection of the cable from the EVSE

CarSharingBookingStatusRequest

Attributes	Datatype	Constraints	Description
EVSE *ID*	2127: BO EVSEID	mandatory/MULTIPLICITY: 1	unique EVSE identification number.
EVCOID	2165: BO EVCOID	mandatory/MULTIPLICITY: 1	unique EV user contract identification number.
VIN	2360: BO VIN	mandatory/MULTIPLICITY: 1	unique EV identification number
TimeStamp	DateTime	mandatory/MULTIPLICITY: 1	.
BookingStatus	enum	two possible values for the attribute: 1: vehicle departed/2: vehicle arrived	booking of the EV not EVSE

CarSharingBookingStatusResponse

Attributes	Datatype	Constraints	Description
CarSharingBookingStatusAcknowledge	boolean	Multiplicity:1 mandatory	if TRUE succesfull response i sent
CarSharingBookingStatusErrorCode	integer	Multiplicity:1 mandatory	if attribute CarSharingBookingStatusAcknowledge is FALSE an error code is given
CarSharingBookingStatusErrorDescription	text	Multiplicity:1 mandatory	desxcription of error code to be indicated in the previous attribute

Service specific Business Objects (Not part of Domain Model)

2356: SRV GetBatteryStatus

This service is required by the Car Sharing operator to the EVSE operator in order to get the EV battery status, which is a needed information for the CSMS booking algorithms

GetBatteryStatusRequest

Attributes	Datatype	Constraints	Description
EVSE ID	2127: BO EVSEID	mandatory/MULTIPLICITY: 1…*	unique EVSE identification number. battery status is requested for all EVSEID of EVSEs in the ownership of the car sharing operator

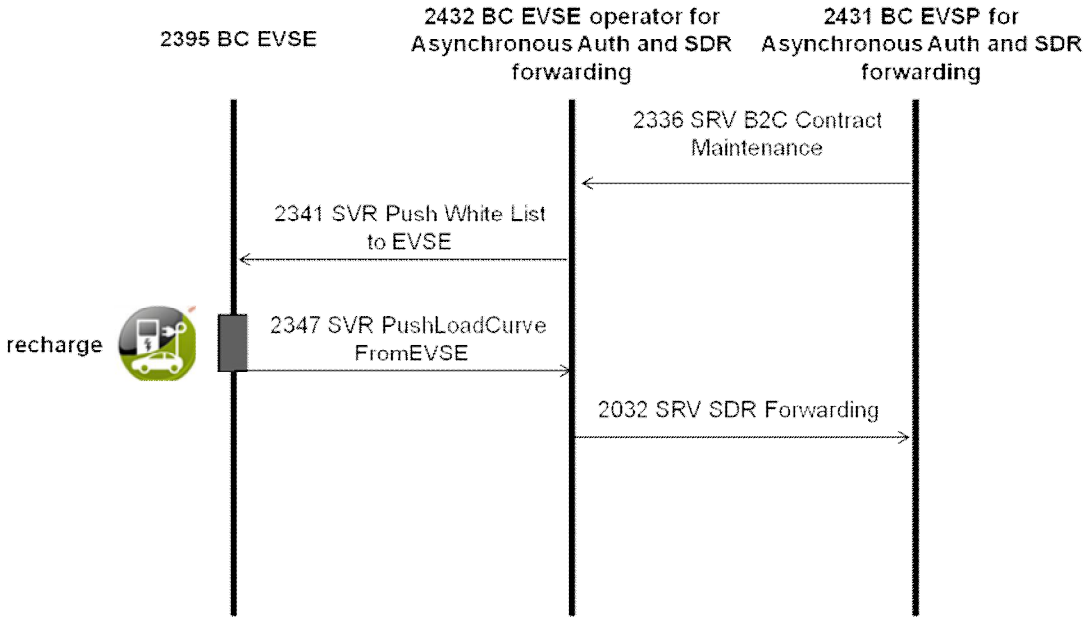
GetBatteryStatusResponse

Attributes	Datatype	Constraints	Description
EVSE ID	2127: BO EVSEID	mandatory/MULTIPLICITY: 1…*	unique EVSE identification number
BatteryStatusOfCharge	2146: BO EV charge data item	mandatory/MULTIPLICITY : 1…*	the battery status of charge of the indicated EVSEID in the request is a percentage of the available energy in the EV battery over its full capacity

2352: UC Recharging with no connectivity with the EVSE operator backend

Sequence Diagrams	2425: SQD Recharging with no connectivity
Business Components	2432: BC EVSE Operator for Asynchronous Auth and SDR forwarding 2431: BC EVSP for Asynchronous Auth and SDR forwarding
Services	2347: SRV PushLoadCurveFromEVSE 2341: SRV PushWhiteListUpdateToEVSE 2336: SRV B2C contract maintenance 2032: SRV SDR Forwarding

2425: SQD Recharging with no connectivity



2432: BC EVSE Operator for Asynchronous Auth and SDR forwarding

This component describes the EVSE operator ICT capabilities to require to EVSPs a list of allowed EV users that can recharge at its EVSEs, and the EVSE operator ICT capabilities to require the asynchronous SDR forwarding to intersested EVSPs

Provided Services	2336: SRV B2C contract maintenance 2347: SRV PushLoadCurveFromEVSE
Required Services	2032: SRV SDR Forwarding 2341: SRV PushWhiteListUpdateToEVSE

2336: SRV B2C contract maintenance

This interface contains four different methods which can be called to manage B2C contracts in a system which offers the service.
The four methods are create, read, update and delete.

CreateB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

CreateB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
responseCode	2174: ENUM ResponseCode	1	

ReadB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
EVCOID	2165: BO EVCOID	1..*	Contract ID which contains EVSP ID and Customer ID.

ReadB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

UpdateB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

UpdateB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
responseCode	2174: ENUM ResponseCode	1	

DeleteB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

DeleteB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
responseCode	2174: ENUM ResponseCode	1	

2347: SRV PushLoadCurveFromEVSE

This service is required by EVSE in order to send to the EVSE operator backend the charging session load curve

PushLoadCurveRequest

Attributes	Datatype	Constraints	Description
chargingSessionLoadCurve	2348: BO ChargeSessionLoadCurve	Multiplicity: 1	Charge session load curve, identified by chargeSessionId, EVSEId, EVCOId

PushLoadCurveResponse

Attributes	Datatype	Constraints	Description
ack	boolean	Multiplicity: 1	Load curve received acknowledge

Service specific Business Objects (Not part of Domain Model)

2348: BO ChargeSessionLoadCurve

This object contains detailed charging session's info, including load curve, EVSE identifier, EVCO identifier, charge session identifier

Attributes	Datatype	Constraints	Description
EVSEID	2127: BO EVSEID	Multiplicity: 1	EVSE unique identifier
EVCOID	2165: BO EVCOID	Multiplicity: 1	EVCO unique identifier

Attributes	Datatype	Constraints	Description
chargeSessionID	string	Multiplicity: 1	Charge session unique identifier
loadCurve	2133: BO Load Curve	Multiplicity: 1	Vector of time intervals and energy related to time intervals

2431: BC EVSP for Asynchronous Auth and SDR forwarding

The BC provides asynchronous authorization and SDR forwarding for an EV user.

Provided Services	2032: SRV SDR Forwarding
Required Services	2336: SRV B2C contract maintenance

2032: SRV SDR Forwarding

Interface to forward the service detail record (SDR) from one place to another.
This interface is provided by the clearing house and the EVSPs.
In that way, the EVSE operator can send the SDR to the clearing house and the clearing house can forward the SDR to the respective EVSP.

SDRForwarding

Attributes	Datatype	Constraints	Description
serviceDetailRecord	2033: BO SDR	Multiplicity: 1	
referenced by SDR class	2034: BO SDR Service Record	Multiplicity: 1..*	
referenced by SDR class	2035: BO SDR ChargeDataRecord	Multiplicity: 0..1	One SDR could contain one ChargeDataRecord
referenced by SDR ChargeDataRecord	2036: EnergyConsumptionRecord	Multiplicity: 1..*	One ChargeDataRecord can contain multiple EnergyConsumptionRecords

SDRForwardingResponse

Attributes	Datatype	Constraints	Description
responseCode	2174: ENUM ResponseCode	Multiplicity: 1	

Service specific Business Objects (Not part of Domain Model)

2174: ENUM ResponseCode

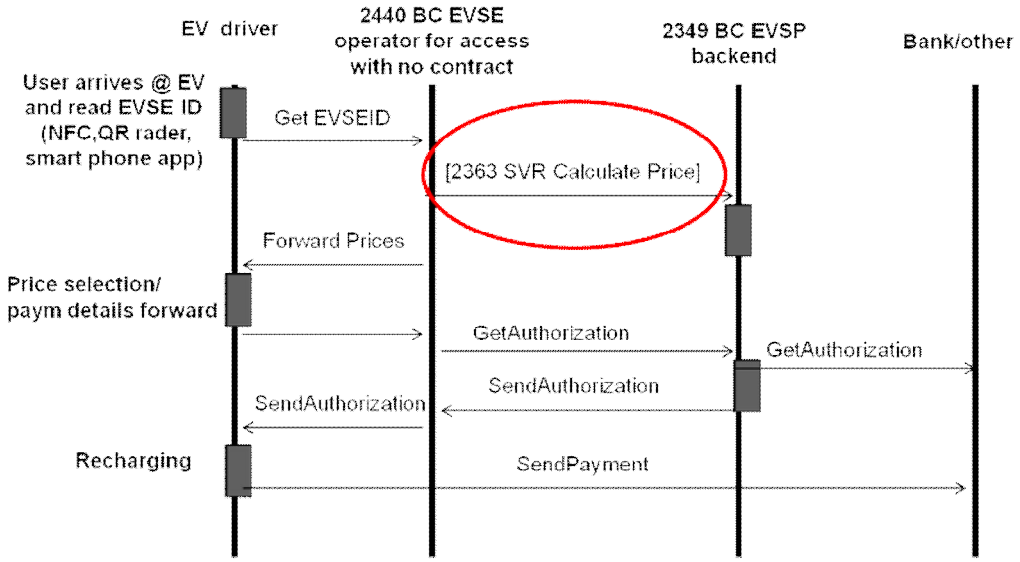
The enumeration ENUM ResponseCode contains the possible values for the response sent for the SDR Forwarding in the roaming domain. The values are based upon the definition of HTTP return codes but do not relate to the response on HTTP level but rather describe the response on the application level.

Values	Description
200	OK
400	Bad Request (Syntax Error)
500	Internal Server Error (Application Error)

2362: UC Access without service contract

Sequence Diagrams	2421: SQD Access without service contract
Business Components	2349: BC EVSP backend 2440: BC EVSE Operator for Acces with No Contract 2395: BC EVSE
Services	2363: SRV CalculatePrice

2421: SQD Access without service contract



2349: BC EVSP backend

This BC represents the ICT infrastructure of the EVSP

Provided Services	2363: SRV CalculatePrice 2381: SRV Reservation status/changes 2366: SRV Reservation request from EVSP
Required Services	

2363: SRV CalculatePrice

This service is required by the *EVSE operator* to the EVSP in order to get priced offers form energy provisioning at an EVSE where and EV driver arrives and has no contract with an EVSP

CalculatePriceRequest

Attributes	Datatype	Constraints	Description
EVSE ID	2127: BO EVSEID	mandatory/MULTIPLICITY: 1	unique EVSE identification number.

CalculatePriceResponse

Attributes	Datatype	Constraints	Description
EVSE ID	2127: BO EVSEID	mandatory/MULTIPLICITY: 1	unique EVSE identification number
EVCOID	2165: BO EVCOID	mandatory/MULTIPLICITY: 1	a temporary Contract ID is assigned to the customer in order to authorize charging and for billing
PriceList	2396: BO Price	mandatory/MULTIPLICITY : 1	provides a vector of offers+prices for the specified EVSEID

Service specific Business Objects (Not part of Domain Model)

2396: BO Price

This object describes the price options offered by EVSPs to an EV customer at an EVSE, who as no contract with EVSPs

Attributes	Datatype	Constraints	Description
Option	integer	• Multiplicity: 1..* • Mandatory	Identification of price option
Price	float	• Multiplicity:1...* • Mandatory	price of the charging services

2381: SRV Reservation status/changes

For some events during reservation handling the reservation service might also send notifications to the relevant EVSP.
This various notifications on reservation actions like reservation reminder, expiration or suspension will be send immediately when such action event is raised to the EVSP backend.
Reservation notifications are just for information purpose. There is no direct activity on such notification expected by the reservation service.
The EVSP can then inform the end-user (EV-driver) about the actual reservation status.
This SRV specifies a notification to an EVSP. It is raised when any reservation activity occurs on the reservation service which is relevant for the end-user (EV-driver):

Parameter	Datatype	Constraints	Description
Reservation	2083:BO EVSEReservation		Adress (City) of reservable EVSE
Reservation status	string	1	This enumeration specifies the reservation status (e.g. canceled, confirmed, expired, prebooked, suspended, used)

Service specific Business Objects (Not part of Domain Model)

2366: SRV Reservation request from EVSP

The service reservation-module will provide the following UIs which can be implemented to a EVSP customer web page or portal.
- EVSE reservation UI
- Manage EVSE reservation UI

The provided UIs layouts are optimized for Internet Explorer 8.
The following parameters can be configured from the EVSP customer web page for the reservation functionality:

Parameter	Datatype	Constraints	Description
EVSP-ID	2389: BO EVSPID	Mandatory	EVSP id from the registered EVSP portal
Location	2120: BO GeoCoordinates	Mandatory	Default location shown in map when launching the embedded UI
Max Result	int	0..1	Defines how many results should be displayed at maximum when executing a search for reservation of EVSE
Status	1989: BO EVSEStatus	0..1	Filter from EVSP to select the necessary EVSE (e.g. "Available", "Planned")
MapOffsetDistance	int	1	Defines the padding between map border and pin.
InfoboxOffsetPosition	int	1	Defines the offset in pixel for the info box/tooltip of the pins within the map
GroupingOffsetDistance	int	1	Defines distance in pixel between to pins on map from which those pins have to be aggregated to one pin
EVCO-ID	2165: BO EVCOID	1	The reservation request will be done for this EVCO-ID
TimeZone-ID	int	1	Defines time zone for center of map. This is necessary in order to retrieve the location status according to specified start and end time in correct time zone.
Reservation-Time	2083: BO EVSEReservation	0..1	This parameter enables the reservation functionality. The following values are possible: "Period_Reservation", "Point in Time Reservation"
MaxReservationDate	2083: BO EVSEReservation	0..1	This parameter defines the maximum for a reservation start or end date. Default are "6 month in advance"
MaxReservationDays	2083: BO EVSEReservation	0..1	This parameter defines the maximum offset between reservation stard and end date in case of Period reservation
ManageReservationURL	string	optional	This parameter should provide a full URL to the "Manage Reservation" page. The link to this URL is displayed on every "Congratulations" screen.

Parameter	Datatype	Constraints	Description
Search and Reserve URL	string	optional	This parameter should provide a full URL to the "Search and Reserve" page (Reservation UI)
Terms and Conditions URL	string	optional	It should provide a full URL to the terms and conditions of the EVSP
Max Reservations	2083: BO EVSEReservation	1	Defines how many reservations should be displayed in manage reservation UI
Reservation Start Date	2083: BO EVSEReservation	1	With this parameter only those reservations will be displayed on manage reservation UI where reservation start date is after specified date
Reservation End Date	2083: BO EVSEReservation	1	With this parameter only those reservations will be displayed on manage reservation UI where reservation end date is before specified date
Startbooking	2083: BO EVSEReservation	1	Defines the column which should be sorted on manage reservation UI (Startbooking, ChargingLocation, Status)

The following fields will be provided from the reservation UI for EVSP driver to send the reservation request:

Parameter	Datatype	Constraints	Description
Location	1988: BO Location	1	Adress (City) of reservable EVSE
Reservation start	2083: BO EVSEReservation	1	Reservation start time/date
Reservation end	2083: BO EVSEReservation	0..1	Reservation end time/date
EVSE plug type	1986: BO Plug	0..1	Required plug type for reservation (e.g. Type2, Schuko)
EVSE Charging mode	2114: ENUM ChargingMode	0..1	Required charging mode (e.g. AC, DC)
EVSE Max-amp	1985: BO ChargingPoint (EVSE)	0..1	Required charging load (e.g. 16 ampere, 32 ampere, 64 ampere)
EVSE-operator-ID	2130: BO EVSEOperatorID	0..1	Relevant EVSE-Operator

ReservationForEVSERequest

Attributes	Datatype	Constraints	Description
searchCriteria	2012: SRV SearchForEVSE	• required	Contains the search criterias necessary to book an EVSE.
AccessType	2119: ENUM AccessType	Multiplicity: 0..*	EVSE access type (e.g. public, company, private)
Authenticationmethod	2117: ENUM AuthenticationMethod	Multiplicity: 0..*	EVSE Authenticationmehtod (e.g. RFID, V2G_CI, SMS to EVSE operator, APP, Park&Charge, No authentication necessary)

Attributes	Datatype	Constraints	Description
Bookable	1985: BO ChargingPoint (EVSE)	Multiplicity: 0..1	True, if search only on bookable EVSEs
Chargingmode	2114: ENUM ChargingMode	Multiplicity: 0..*	EVSE chargingmode (AC, DC, Inductive)
GeoCoordinates (adress)	1988: BO Location	Multiplicity: 0..1	Search criteria for proximity search depending on adress (e.g. City, country) or geo coordinates (e.g. lattitude, longitude) of the EVSE
EVSE-operator-ID	2130: BO EVSEOperatorID	Multiplicity: 0..*	List of relevant EVSE-operator-ids to search for
EVSE-operator-name	1987: BO EVSEOperator	Multiplicity: 0..*	Search criteria for a single EVSE-operator name
EVSE-type	string	Multiplicity: 0..1	Search criteria, if search for EVSE or battery switch station
EVSE-Geo-search criteria	1988: BO Location	Multiplicity: 0..1	Search criteria for proximity search depending on a geo position (e.g. latitude, longitude)

ReservationForEVSEResponse

Attributes	Datatype	Constraints	Description
Charging pool	2167: BO ChargingPoolID	Multiplicity: 0..*	List with found Charging-pools where at least one EVSE meets the search criteria

Service specific Business Objects (Not part of Domain Model)

2041: BO EVSP

Describes the attributes of the business object EVSP within the EVSP Backend
(Not used for V1.0, might be used for SRV 2016 Current Charge Data via direct link connection EV to EVSP)

Attributes	Datatype	Constraints	Description
EVCOID	string	• max. length: 255 • required	Globally unique ID of the EVSP
web	string	• max. length: 255 • required	Home page address
VIN	string	• optional	one of the possibilities of authentication at EVSP
RFID	string	• optional	one of the possibilities of authentication at EVSP

Attributes	Datatype	Constraints	Description
UserID	string	• optional	one of the possibilities of authentication at EVSP, username + password

2440: BC EVSE Operator for Acces with No Contract

The BC describes the ICT capabilities of the EVSE operator dedicated to providing charging services to EV drivers with no service contract.

Provided Services	
Required Services	2363: SRV CalculatePrice

2395: BC EVSE

This component describes the ICT capabilities of an EVSE

Provided Services	2341: SRV PushWhiteListUpdateToEVSE
Required Services	2347: SRV PushLoadCurveFromEVSE

2341: SRV PushWhiteListUpdateToEVSE

This service is used by EVSE operator backend to update the list of allowed EV users on an EVSE

PushWhiteListUpdateRequest

Attributes	Datatype	Constraints	Description
whiteListUpdate	2342: BO WhiteList	multiplicity: 1	Unique identified list of EV users allowed to recharge

PushWhiteListUpdateResponse

Attributes	Datatype	Constraints	Description
ack	boolean		WhiteList accepted acknowledge

Service specific Business Objects (Not part of Domain Model)

2342: BO WhiteList

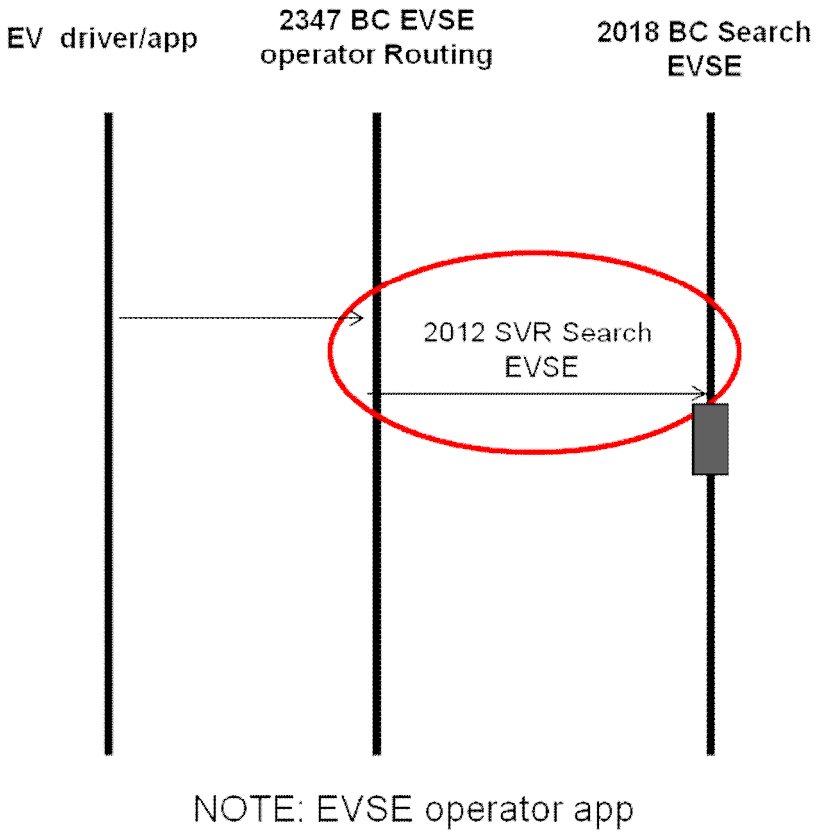
This BO is a unique identified list of EVCOID, allowed to recharge their vehicles on an EVSE

Attributes	Datatype	Constraints	Description
versionId	string	multiplicity: 1	Unique identifier of the white list's version
whiteList	2165: BO EVCOID	multiplicity: 1...*	White list of EV users locally allowed to recharge

2364: UC Routing to EVSE

Sequence Diagrams	2423: SQD Routing to EVSE
Business Components	2365: BC EVSP Routing 2437: BC EVSE operator Routing 2018: BC Search EVSE
Services	2012: SRV SearchForEVSE

2423: SQD Routing to EVSE



2365: BC EVSP Routing

The BC EVSP Routing provides routing services to the EV user to the nearest available EVSE.

In order to provide such service, the EVSP has to consume search functionalities from the EVSE operator.

Provided Services	
Required Services	2012: SRV SearchForEVSE

2437: BC EVSE operator Routing

The BC describes the ICT capabilities of the EVSE operator to provide services related to Routing to an EVSE.

Provided Services	
Required Services	2012: SRV SearchForEVSE

2018: BC Search EVSE

This BC was exported before.

4 Roaming

4.1 Domain Model

DM Roaming Domain

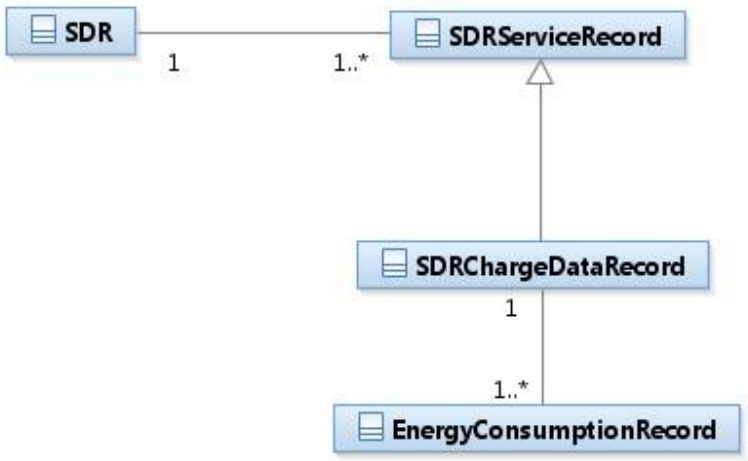


Figure 4.1: High level SDR domain model

Roaming of EV related services occur when a service is contracted between consumer A and provider B, but is delivered to consumer A by provider C, based on a contract between provider B and provider C. Consumer A does not need an additional contract with provider C. To support the roaming scenario, dedicated roaming services have to be in place. Roaming services can be handled in a bilateral agreement between an EVSE operator and an EVSP or with a clearinghouse to handle the contractual clearing between the EVSE operator and the EVSP.

This deliverable will show the needed services for authorization means either directly on the charge point or via a mobile application as well as the service for a standardized way of transferring data records about the service usage to the respective service partner. Additionally the contractual relationships have to be handled in a way that they are accessible in order to confirm them if necessary.

Authorization

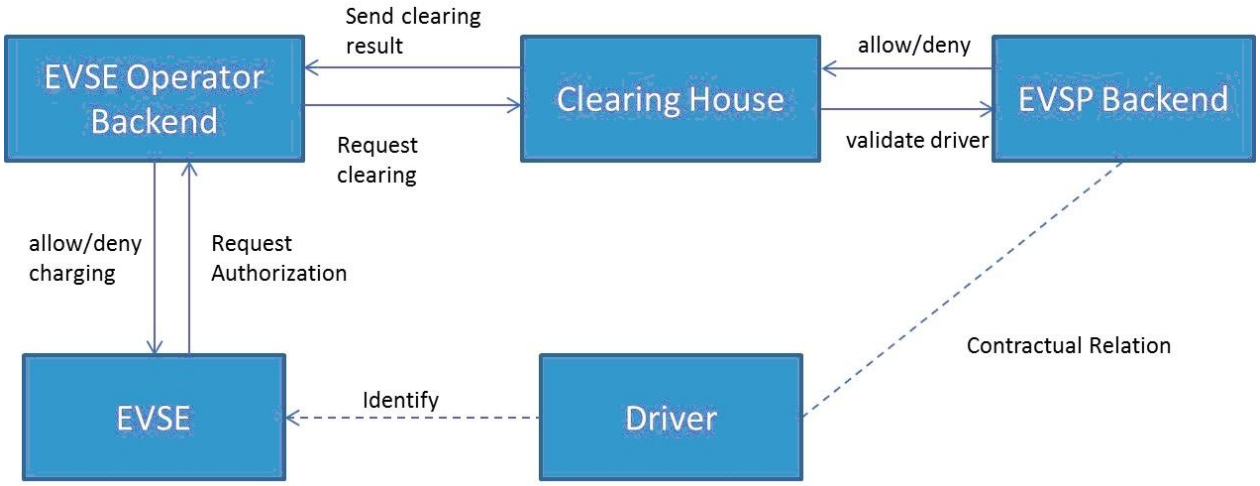


Figure 4.2: Authorization at EVSE

The service Authorization (SRV 2020) is used to handle the authorization of a user. In most cases the authentication is done by swiping a RFID card as illustrated in figure 4.2. In roaming use cases the RFID is unknown to the charge point and the EVSE operator backend. Instead of using the RFID card ID itself, a ContractID, which will be included in the RFID content, may be used to identify the user. This concept is already used by some companies. The same ContractID will be part of the ISO 15118 where the compatible car will trigger the authorization. The EVSE operator backend will forward the charge request directly to the EVSP in case of bilateral roaming agreements or to the clearinghouse in case of centralized roaming agreements. In both cases the Authorization interface will be used and has to be implemented at the EVSP and the clearinghouse side.

The information about the drivers' EVSP is part of the ContractID. Based on the RFID card ID the EVSP cannot be concluded. Here a translation service has to be in place which can reside within the clearinghouse. In this case, the clearinghouse receives the authorization request with the RFID card ID and translates it to the corresponding ContractID. Besides the ContractID, the EVSEID where the charging takes place has to be included in the AuhtorizationRequest message. Optional field such as vehicle identification number or battery identification number allow for advanced business models and security checks.

The clearinghouse then performs the contract validation for B2B contracts between the corresponding EVSE operator, which forwarded the request to the clearinghouse, and the EVSP of the EV-driver. The contractual relationship between business partners can be stored in the clearinghouse directly or by another entity. In the GeM setup, the B2B contracts are concluded and stored on the marketplace which provides the needed information to the clearinghouse. For that the service ContractValidation (SRV 1967) is provided by the marketplace and used by the clearinghouse. If a valid B2B contract is in place, the B2C contract is validated next. Therefore, the clearinghouse can check its own database or ask the corresponding EVSP. Each EVSP needs to have a system in place which implements the same Authorization service to receive those messages and react accordingly. Based on the B2B and B2C validation, the clearinghouse will send an AuthorizationResponse message with the responseValue true or

false for acceptance or rejection back to the EVSE operator backend where the charging request came from.

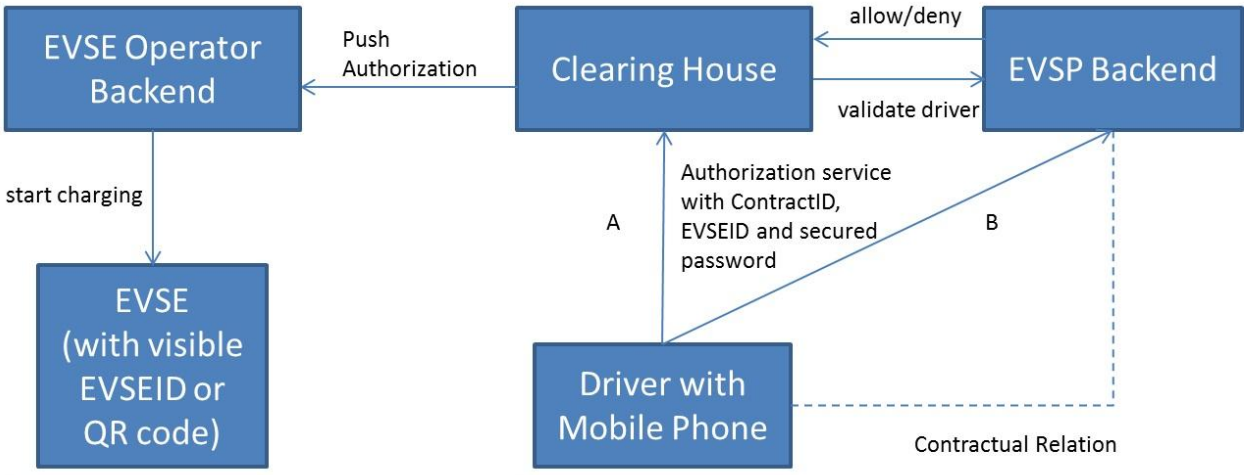


Figure 4.3: Authentication via Smartphone

Besides the identification with RFID card, Green eMotion supports the PushAuthorization service. That allows for another way of triggering the authorization without RFID.
The PushAuthorization service can be used by a smartphone native application or a web application as depicted in figure 4.3. The driver, who wants to charge at a charging point which has no RFID card reader, opens his app and enters his ContractID, password and the EVSEID at which he is going to charge. The logon information could either be sent to the service interface of the clearinghouse (A) or directly to the responsible EVSP's backend interface (B). After a positive assessment in case B, the EVSP would send a PushAuthorizationRequest to the clearinghouse which will forward this request to the corresponding EVSE operator so that he can start the charging process. In case A the smartphone application sends the logon information to the clearinghouse, which is depicted in figure 4.3. That triggers the regular authorization process. The clearinghouse would perform the B2B and B2C validation. After a positive validation, the clearinghouse would send the PushAuthorizationRequest to the EVSE operator. In both cases the EVSE operator backend will respond with a PushAuthorizationResponse message via the same data flow as the request did arrive. The response acts as a confirmation of the requested parameters to the requester.

Data Records

A charging process produces a data record in which for example the amount of consumed kilowatt hours as well as start and end of the charging process is noted. In case of roaming the responsible EVSP of the driver who charged, might want this information in order to charge his customer. So the EVSE operator has to send the data record to the EVSP. This starts with the end of the charging process which can be triggered by the vehicle, the driver, the charge point or the EVSE operator backend. In all cases the EVSE operator backend sends the corresponding service data record (SDR) to either the EVSP in case of bilateral agreements or the clearinghouse in case of centralized agreements. In case of the latter, the SDR is then validated and forwarded to the EVSP which is responsible for the customer who charged his car. The service interface SDRForwarding (SRV 2032) needs to be implemented by the clearinghouse and all EVSPs as they want to receive the roaming charges of all their customers.

Contract Information

As described in the Chapter 'Authorization', the clearinghouse can ask the EVSP about its customers and the respective contract information to perform the B2C validation. In order to allow a faster response by the clearinghouse to the EVSE operator, the EVSP can decide to store the relevant customer data in the clearinghouse centrally. As soon as a ContractID is stored in the clearinghouse, no additional request to the EVSP has to be performed by the clearinghouse. This procedure saves time compared to the additional request and allows for a quicker decision of the EVSE operator to grant or deny the charge request at the EVSE. In order to enhance the usability of maintaining the roaming ContractIDs by the clearinghouse, the service B2C contract maintenance (SRV 2336) is in place that can create, read, update and delete ContractIDs. Each EVSP can only see and modify its own ContractIDs. Either single entities or whole lists can be managed with one service call.

Roaming in a Multi-Marketplace environment

The current situation shows many marketplaces (MP) for electric mobility arising. Companies don't want to register at each and every marketplace there is, but still might want to offer their service everywhere in order to achieve a maximum coverage of potential customers. In this setup an EVSE operator might be registered to a MP1 and an EVSP, which has a roaming agreement with that EVSE operator, might be registered to another MP2. In order to keep the same service interfaces for roaming in the multi-marketplace scenario, we came up with a general converter which is service unspecific. More information can be found in the marketplace chapter. The roaming services such as Authorization and SDRForwarding are also handled by the converter. With this mechanism in place, each marketplace domain can keep their service specification as long as a conversion of two substituting services is possible. For example we have protocol 1 in MP1 domain and protocol 2 in MP2 domain. If every method of protocol 1 can be translated into a method of protocol 2, a converter can be used.
In order to allow roaming over different marketplaces the following steps have to be performed:

1. EVSE Operator offers Roaming Agreement on MP1
2. EVSE Operator buys Authorization and SDR service from CH1
3. MP1 sends offer to MP2 via converter
4. EVSP buys Roaming Agreement on MP2
5. MP2 informs EVSE Operator about purchase via converter
6. (a) EVSP stores EVCOIDs in CH2 -> buy B2C contract maintenance service on MP2) or b) EVSP offers Authorization service on MP2 and MP2 offers Authorization service on MP1 via converter and CH1 buys it
7. EVSP offers SDR service on MP2 and CH1 buys it

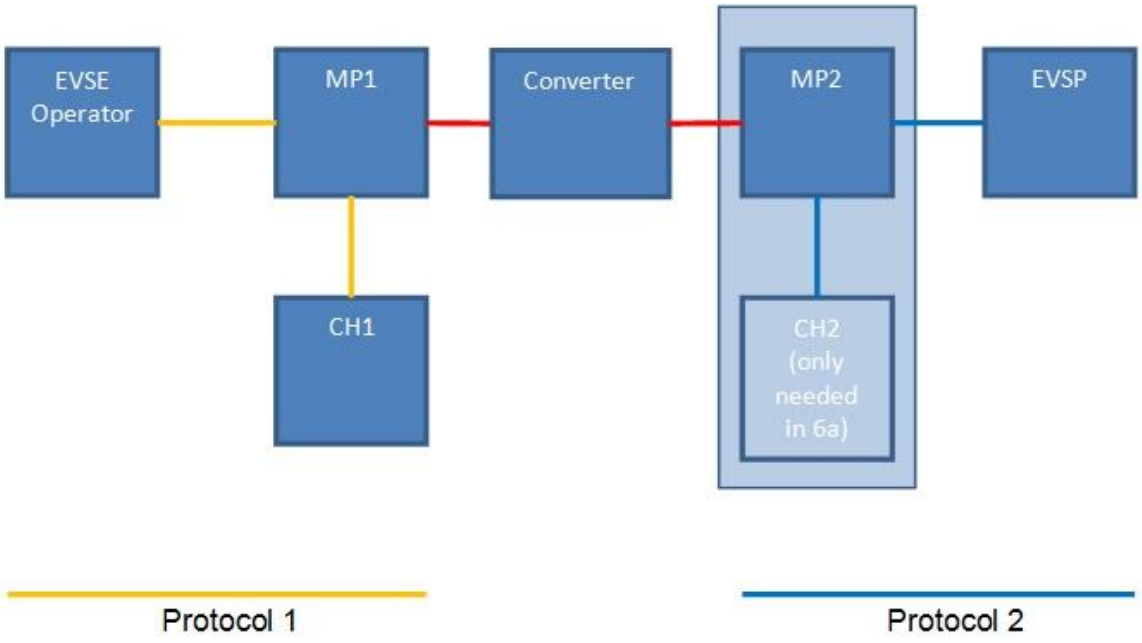


Figure 4.4: Roaming in multi-marketplace environment

Note: If EVSP does not offer an own Authorization service and stores the EVCOIDs or even the RFID IDs in the CH2, than the CH2 needs to offer an Authorization service on MP1 which can be bought by CH1.
Example: Customer of EVSP uses RFID card at EVSE. EVSE Operator sends auth request to CH1 with RFID ID. CH1 sends auth request to CH2 which performs the translation, B2B and B2C validation and returns the answer to CH1. CH1 responds to the EVSE Operator.

As depicted in Figure 4.4, the EVSE Operators and EVSPs don't have to create new interfaces. The only additional development is done by the marketplaces to introduce the converter. This is only done once regardless of how many different services need to be converted.

2033: BO SDR

Service Detail Record

Attributes	Datatype	Constraints	Description
sessionId	string	1	will be created by EVSE operator backend system (just unique in one EVSE Operator system)
EVCOID	2165: BO EVCOID	1	Contract ID which contains EVSP ID and Customer ID. The ID is split up into 3 sub elements (CountryCode, EVSPID, CustomerID)
services	2034: BO SDR Service Record	1..*	A list of services which are tracked in the SDR, so that the EVSP can bill his customer accordingly.

2035: BO SDR ChargeDataRecord

Shows the structure of the ChargeDataRecord which can be part of the SDR.
The SDR ChargeDataRecord inherits from the superclass BO SDR Service Record.

Attributes	Datatype	Constraints	Description
EVSEID	2127: BO EVSEID	1	ID of the charge point where the charging took place.
startOfCharging	dateTime	1	the date and time at which the charging point activates its electric contactor and starts the power supply to the car
endOfCharging	dateTime	1	the date and time at which the charging point deactivates its electric contactor and stops the power supply to the car
energyConsumptionList	2036: BO EnergyConsumptionRecord	1..*	A list of records which indicate how much energy was used during a specific period of time.

2036: BO EnergyConsumptionRecord

A record which indicates how much energy was used during a specific period of time.

Attributes	Datatype	Constraints	Description
start	dateTime	1	start of the time interval
end	dateTime	1	end of the time interval
consumedEnergy	float	1	the amount of consumed energy during the specified time interval in kWh

2034: BO SDR Service Record

This business object represents a generic superclass for services used in the service detail record.
Other business objects inherit from the SDR Service Record type:
In Green eMotion, only the ChargeDataRecord (2035) will be used. In the future other business objects might be added (e.g. ReservationDataRecord)

4.2 Functional Decisions

1991: FD Management of EVSP/EVSE Contracts

Affected Use Cases	1512: UC Start a roaming charging process with Clearinghouse

The Roaming agreements, contracts between EVSPs and EVSE Operators, will be managed in the Marketplace. The Clearing House will use a Marketplace Service to check if a contract exists for a specific Roaming transaction.
Roaming agreements will be treated like any other IT service related contract within the Marketplace. Therefore UCs, that are related to the management of such contracts do not need to be specified explicitly.

2037: FD System internal

Affected Use Cases	1497: UC Create EVSP/EVSE Contract within Clearing House 1500: UC Change EVSP/EVSE Contract within Clearing House 1513: UC Delete EVSP/EVSE Contract from Clearing House

This use cases need no service interface because the process is only working internally in one particular system. There is no need to provide that functionality to partners via interfaces.

4.3 Service Specification

1512: UC Start a roaming charging process with Clearinghouse

Business Components	1966: BC Business Services 2038: BC Clearing House
Services	2020: SRV Authorization 1967: SRV ContractValidation 2335: SRV Push Authorization

Related Functional Decisions	1991: FD Management of EVSP/EVSE Contracts

1966: BC Business Services

The "Business Services" component relates to the Business Services Sub Domain of the Marketplace described in the GeM Reference Architecture. The component will implement the following functionality and expose it to users of the marketplace through User Interfaces and - for some functionality that needs to be accessed by other systems, as the Clearing House - also through Service Interfaces:

- Service Store: Is used to browse and sell or buy Services. Service Requesters are using this functionality to browse through offered Services. During the purchase process a contractual relationship is established between the Business Partners. Furthermore the Service Store is used to bring in and remove services to/from the Marketplace.
- Partner Management: It offers functionality to manage “customer data” of all Marketplace participants. Customers in the Marketplace context are the Service Providers and Service Requesters.
- Contract Management: It offers functionality to administrate the relations between Marketplace participants. It maintains the information which service requester has contracted which services. This includes contracts for IT services, which are usually delivered through the marketplace, as well as conventional contracts that do not relate to an IT service (e.g. Roaming contracts).
  Not in scope: Payment and Billing, which is also described in the Reference Architecture, is currently not in scope for implementation.

Provided Services	1967: SRV ContractValidation
Required Services	

1967: SRV ContractValidation

This service is used to check if a contract of a defined category (e.g. roamagr) exists in the marketplace between a Service Provider and a Service Requester.

The Service Provider and Service Requester are usually identified with their Marketplace Business Partner ID. To support the Roaming scenario (where the Clearing House does not know the Marketplace Business Partner ID) it is also possible to use the EVSEOperatorId (part of the EVSEID) and the EVCOProviderId (part of the EVCOID) to identify the business partners.
While the first intended usage of the ContractValidation was to support the Roaming scenario only, the contract validation will be made available to all business partners in order to check for valid contracts of all types within the Marketplace. As a consequence the usage of alternative Ids will not be restricted to the scenario mentioned above and the idTypes must be explicitly passed to the service in order to put the alternative Ids in the right context.

ContractValidationRequest

Attributes	Datatype	Constraints	Description
provider	String		Matches the partnerId defined in 1461: BO BusinessPartner. This represents the Service Provider in the marketplace for the contract which has to be checked.
alternativeProviderId	2064: BO AdditionalId	Multiplicity: 0..1 Either "provider" or "alternativeProviderId" must be set	Can be used to identify the Service Provider if the Business Partner ID is unknown - e.g. in the Roaming scenario where only the EVSEOperatorId is known which is part of the EVSEID (suggested for standardization). (e.g. for a roaming agreement validation, EVSEOperatorId is the type to be used as idType for the BO AdditionalId)
requester	String		Matches the partnerId defined in 1461: BO BusinessPartner. This represents the Service Requester in the marketplace for the contract which has to be checked.
alternativeRequesterId	2064: BO AdditionalId	Multiplicity: 0..1 Either "requester" or "alternativeRequesterId" must be set	Can be used to identify the Service Provider if the Business Partner ID is unknown - e.g. in the Roaming scenario where only the EVCOProviderId is known which is part of the EVCOID (suggested for standardization). (e.g. for a roaming agreement validation, EVCOProviderId is the type to be used as idType for the BO AdditionalId)

Attributes	Datatype	Constraints	Description
service	String	alphanumeric identifier	Matches the serviceTypeId defined in 2087: BO ServiceType Identifies the contract type that will be checked for the provider and requester. (e.g. for a roaming agreement validation, "roamagr" is the literal to be used)

ContractValidationResponse

Attributes	Datatype	Constraints	Description
result	boolean		**True** if a valid contract of the defined category exists between requester and provider. **False** if no valid contract of the defined category exists between requester and provider.

2038: BC Clearing House

The clearing house deals with incoming authorization request and SDRs. It is used to enable roaming.
The authorization requests can come from the EVSE operator or from the EVSP. In the first case the EVSE operator sends an AuthorizationRequest defined in SRV Authorization (2020) to the clearing house if it recognizes an unknown EVCOID. In the second case the EVSP sends a PushAuthorizationRequest defined in SRV Push Authorization (2335) to the clearing house if he receives a charge request of his customer.
In both cases the SDR is send after a charging process as defined in SRV SDR Forwarding (2032). If a charging process has been started without the use of the clearing house, a SDR can be send anyway.
The clearing house offers an additional service to the EVSPs via the service interface SRV B2C Contract maintenance (2336). It is an optional interface for the EVSPs in the roaming domain. By loading a subset of their customer information into the clearing house, the authorization requests for customers of that EVSP can be handled without the additional request to the EVSP. In total the decision if a customer is allowed to roam is conducted faster.
The EVSP can store the EVCOIDs of their customers in the clearing house together with an indication if roaming is allowed or not. In the end this service allows for a black and white list of EVCOIDs.

Provided Services	2020: SRV Authorization 2335: SRV Push Authorization 2032: SRV SDR Forwarding 2336: SRV B2C contract maintenance
Required Services	2020: SRV Authorization 2335: SRV Push Authorization 1967: SRV ContractValidation 2032: SRV SDR Forwarding

2020: SRV Authorization

The clearing house interface which deals with requests from EVSE operators to determine if an EV driver is allowed to charge.
The same interface is implemented by the EVSPs so that the clearing house can ask the EVSP if a customer is allowed to charge at the specified charge point.

AuthorizationRequest

Attributes	Datatype	Constraints	Description
sessionId	string	1	will be created by EVSE operator backend system (just unique in one EVSE Operator system)
EVCOID	2165: BO EVC OID	1	Contract ID which contains EVSP ID and Customer ID. The ID is split up into 3 sub elements (CountryCode, EVSPID, CustomerID)
EVSEID	2127: BO EVSEID	1	Charge Point ID
VIN	string	0..1 see ISO 3779 and others	Vehicle identification number
BIN	string	0..1	Battery identification number

AuthorizationResponse

Attributes	Datatype	Constraints	Description
sessionId	string	1	will be created by EVSE operator backend system (just unique in one EVSE Operator system)
responseValue	boolean	1	if true is sent back to the EVSE operator system, the customer is allowed to charge, if false is returned, the customer is not allowed to charge

2335: SRV Push Authorization

The clearing house interface which deals with requests from EVSPs to trigger the charging process at an EVSE.
The same interface is implemented by the EVSE Operator so that the clearing house forward the push authorization message to the corresponding EVSE Operator at which the customer of the EVSP wants to charge.

PushAuthorizationRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	will be created by sending system (just unique in the sending system)
EVCOID	2165: BO EVCOID	1	Contract ID which contains EVSP ID and Customer ID. The ID is split up into 3 sub elements (CountryCode, EVSPID, CustomerID)
EVSEID	2127: BO EVSEID	1	Charge Point ID
endOfCharge	datetime	0..1	can be set to stop the charging at a specific point in time, timestamp in UTC
duration	int	0..1	can be set to restrict the charge duration, in minutes

Attributes	Datatype	Constraints	Description
kwh	int	0..1	can be set to restrict the amount of kilowatt hours which are provided by the charge point, in watt hours
speedOfCharge	int	0..1	can be set to restrict the speed of charge, in watt
typeOfCharge	2374: BO ENUM TypeOfCharge	0..1	can be set to restrict the type of charge

PushAuthorizationResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	the transactionId of the request to associate the request to the response.
sessionId	string	1	will be created by EVSE operator backend system (just unique in one EVSE Operator system). null if the EVSE operator denies the charging for any reason
endOfCharge	boolean	0..1	value of true is a confirmation of the EVSE Operator that this restriction will be applied, value false or if the field is missing in the response means the restriction cannot be assured
duration	boolean	0..1	value of true is a confirmation of the EVSE Operator that this restriction will be applied, value false or if the field is missing in the response, the restriction cannot be assured
kwh	boolean	0..1	value of true is a confirmation of the EVSE Operator that this restriction will be applied, value false or if the field is missing in the response, the restriction cannot be assured
speedOfCharge	boolean	0..1	value of true is a confirmation of the EVSE Operator that this restriction will be applied, value false or if the field is missing in the response, the restriction cannot be assured
typeOfCharge	boolean	0..1	value of true is a confirmation of the EVSE Operator that this restriction will be applied, value of false or if the field is missing in the response, the restriction cannot be assured

2032: SRV SDR Forwarding

Interface to forward the service detail record from one place to another.
This interface is provided by the clearing house and the EVSPs.
In that way, the EVSE operator can send the SDR to the clearing house and the clearing house can forward the SDR to the respective EVSP.

SDRForwarding

Attributes	**Datatype**	**Constraints**	**Description**
serviceDetailRecord	2033: BO SDR	Multiplicity: 1	
referenced by SDR class	2034: BO SDR Service Record	Multiplicity: 1..*	
referenced by SDR class	2035: BO SDR ChargeDataRecord	Multiplicity: 0..1	One SDR could contain one ChargeDataRecord
referenced by SDR ChargeDataRecord	2036: EnergyConsumptionRecord	Multiplicity: 1..*	One ChargeDataRecord can contain multiple EnergyConsumptionRecords

SDRForwardingResponse

Attributes	Datatype	Constraints	Description
responseCode	2174: ENUM ResponseCode	Multiplicity: 1	

Service specific Business Objects (Not part of Domain Model)

2174: ENUM ResponseCode

The enumeration ENUM ResponseCode contains the possible values for the response sent for the SDR Forwarding in the roaming domain. The values are based upon the definition of HTTP return codes but do not relate to the response on HTTP level but rather describe the response on the application level.

Values	Description
200	OK
400	Bad Request (Syntax Error)
500	Internal Server Error (Application Error)

2336: SRV B2C contract maintenance

This interface contains four different methods which can be called to manage B2C contracts in a system which offers the service.
The four methods are create, read, update and delete.

CreateB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

CreateB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
responseCode	2174: ENUM ResponseCode	1	

ReadB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
EVCOID	2165: BO EVCOID	1..*	Contract ID which contains EVSP ID and Customer ID.

ReadB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

UpdateB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

UpdateB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
responseCode	2174: ENUM ResponseCode	1	

DeleteB2CEntityRequest

Attributes	Datatype	Constraints	Description
transactionId	string	1	Will be created by EVSP system (just unique in one EVSP system).
B2CEntity	2384: BO B2CEntity	1..*	Object which defines for a EVCOID if it is allowed to charge or not.

DeleteB2CEntityResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	The transactionId of the request to associate the request to the response.
responseCode	2174: ENUM ResponseCode	1	

1511: UC End a roaming charging process with Clearinghouse

Business Components	2038: BC Clearing House
Services	2032: SRV SDR Forwarding

2038: BC Clearing House

This BC was exported before.

1514: UC Create Customer Contract by Service Provider in Clearing House

Business Components	2038: BC Clearing House
Services	2336: SRV B2C contract maintenance

2038: BC Clearing House

This BC was exported before.

1515: UC Change Customer Contract by Service Provider in Clearing House

Business Components	2038: BC Clearing House
Services	2336: SRV B2C contract maintenance

2038: BC Clearing House

This BC was exported before.

1516: UC Delete Customer Contract by Service Provider from Clearing House

Business Components	2038: BC Clearing House
Services	2336: SRV B2C contract maintenance

2038: BC Clearing House

This BC was exported before.

1497: UC Create EVSP/EVSE Contract within Clearing House

Related Functional Decisions	2037: FD System internal

1500: UC Change EVSP/EVSE Contract within Clearing House

Related Functional Decisions	2037: FD System internal

1513: UC Delete EVSP/EVSE Contract from Clearing House

Related Functional Decisions	2037: FD System internal

2249: UC Remote Authentication / Push Authorization

Business Components	2038: BC Clearing House
Services	2335: SRV Push Authorization

2038: BC Clearing House

This BC was exported before.

5 Energy

5.1 Domain Model

Energy domain model

The Energy UCs described in D3.4 are implemented by the SRV and BC depicted in this chapter.
All Energy B2B services shall enable DSOs and EVSE Operators/EVSP to deal with basic load management requests, upon the implementation of specific BCs on their systems. Indeed, BCs and SRVs being described in this document enable the proper information flow between EVSE Operators and DSOs at a M2M communication layer.
Such ICT service infrastructure is the backbone of the E2E process execution of load management, which prerequisite is a proper infrastructure running on the DSO side.
The purpose of the E2E process execution of load management is to control a charging process through EVs dynamic load to respond to grid-supporting opportunities, including the increase of local RENs hosting, as further described in D4.2 (see http://www.greenemotion-project.eu/dissemination/deliverables-infrastructure-solutions.php).
Release 1 specifications of load management, described in D3.5, have already provided a comprehensive set of BCs and SRVs that will be further validated by Release 2 E2E demonstrations for load management. In particular, two main BCs have been identified in the exchange of service interfaces in the Energy Domain:

- **BC EVSE energy back-end**

A software module within EVSE Operator's system able to acknowledge load management request as B2B service and provide back tracking of process. In fact, this component is used to provide energy related services to the DSO. This component is also able to control power modulation at the physical layer of EVSEs according to DSO request.

- **BC DSO Load Management**

A component within DSO's systems with the capability to generate and publish a Load Area (LA), according to a specific LA definition and LV grid description. Also, this component is able to receive inputs from RENs forecast and work-plan management IT systems, in order to perform secondary level load management forecast and export load management requests as a B2B service. Indeed, this component is used to request EVSE energy back-end business component for load / congestion management services and other energy related services.

Through these components DSOs and EVSE Operators will be able to implement the energy use cases cited in D3.4 and provide or request the related services, such as getting information on DSO's load areas, receiving and accepting/rejecting DSO's load management requests and tracking back load management process to inform the DSO of how much load has been managed to support the original request. The following specified SRVs respond to such use case:

· 2126: SRV Get Load Areas Update from DSO
· 1909: SRV EVSE History Use
· 2013: SRV Load Curve
· 2017: SRV DSO predefines peaks
· 2019: SRV Load Management Target
· 2023: SRV Congestion Management Target
· 2024: SRV Load Management Tracking
· 2025: SRV Congestion Management Tracking

A detailed description of these SRVs follows in the document.

121: BO DSO sends signal to aggregator with the preference of CP to be switched

The Business Object contains the following elements.

Name	Description
Marketplace will route specific DSO congestion signal to the Aggregator	Marketplace sends signal to the aggregator with request for curtailment or V2G from particular CP
Marketplace will send general request to the Aggregator for curtailment of load in certain location	Marketplace has received a general (lower priority) congestion signal which suggests to curtail load in a particular location. This signal is routed to the Aggregator (EVSP)

2393: BO AuthorizationToken

The BO AuthorizationToken is used to ensure that the calling application has been validated and has the rights to call the web service.

Attributes	Datatype	Constraints	Description
tokenId	String	Multiplicity: 1 Mandatory Max Lenght: 64 char	Token Id of the consumer for the WS: it is a string that uniquely identify the application. This token ensures that the calling application has been validated and has the rights to call the web service

119: BO Congestion signal from LV substation

The BO contains the following elements.

Name	Description
LV substation is congested	DSO monitors LV substations and it evaluates which one is under congestion
Signal is sent to the marketplace	DSO sends a congestion signal to the marketplace with the specifics of the location

2394: BO EVSEHistoryUse

The BO is used to provide data on EVSE historical use.

Attributes	Datatype	Constraints	Description
evseID	2127: BO EVSEID	Multiplicity: 1 Mandatory	The EVSE ID is the Primary Key of EVSE Item.
numberOfFailedAuthorization	integer	Multiplicity: 1 Mandatory	The number of failed authorization in the time slot specified in the request.

Attributes	Datatype	Constraints	Description
usagePercentage	float	Multiplicity: 1 Mandatory	Shows the usage percentage of the specified EVSE in the specified time slot.
consumedEnergy	float	Multiplicity: 1 Mandatory	It reports the amount of energy consumed for the specified EVSE in the specified time slot in [KWh]
numberOfRecharges	integer	Multiplicity: 1 Mandatory	It counts only recharges with energy consumption in the specified time slot.
averageEnergyPerRecharge	float	Multiplicity: 1 Mandatory	in [Watt hour]
averagePowerOfMeterReading	double	Multiplicity: 1 Mandatory	Average power of all meter readings
averagePowerOfRecharges	double	Multiplicity: 1 Mandatory	Average power of all recharges
nominalPowerOfEvse	double	Multiplicity: 1 Mandatory	Nominal power linked to evseID

2146: BO EV charge data item

This BO identifies the data of EV charge data response.

Attributes	Datatype	Constraints	Description
rechargeID	string	Multiplicity: 1	Unique identifier of the recharge session of the EVSE operator
VIN	string	Multiplicity: 1	ISO 3779 offers a worldwide unambiguous identification number of a vehicle that allows the DSO and other stakeholders to identify the vehicle. If this standard will be adopted by the DSO it is possible to identify the EVs recharge patterns for statistical purposes or energy availability for future V2G applications. The use of a common VIN standard by all EVSE Operator back-ends has to take into consideration, however (out of demo scope), privacy issues.
startBatteryCharge	double	Multiplicity: 1 value between 0 and 100	percentage of the battery charge at the beginning of the recharge session (if not available from the EV, deafult 0)
startBatteryKwh	double	Multiplicity: 1 value equal or greater than 0	Initial stored energy at the beginning of the recharge session (if not available from the EV, deafult 0)
lastBatteryCharge	double	Multiplicity: 1 value between 0 and 100	last known percentage of the battery charge. If the recharge session is over this is the final percentage
lastBatteryKwh	double	Multiplicity: 1 value equal or greater than 0	last known stored energy. If the recharge session is over this is the final stored energy into the battery
rechargeState	char	Multiplicity: 1 "A" for active, "F" for finished	

Attributes	Datatype	Constraints	Description
rechargeStartTime	date	Multiplicity: 1 ISO 8601 format YYYY-MM-DDTHH:MM:SSZ	
rechargeEndTime	date	Multiplicity: 1 ISO 8601 format YYYY-MM-DDTHH:MM:SSZ	

125: BO Ancillary services signal

The BO contains the following elements.

Name	Description
Frequency control	local frequency control signal by the DSO or the EVSE operator to the Marketplace or EVSP
Voltage control	voltage control signal by the DSO or the EVSE operator to the Marketplace or EVSP
Reactive power	demand for the reactive power signalled by the DSO or Energy trader

124: BO Imbalance information

The BO contains the following elements.

Name	Description
Imbalance reduction request by the TSO	EVSP (aggregator) would through the VPP scenario serve as a non obligatory spinning reserve. TSO would based on reaction to its request excercise this reserve (in amount as needed).
Imbalance market (signal from the Energy trader)	EVSP aggregator is following the pre set market treshold prices on the imbalance market and reacts on them, or reacts on signals from the energy trader.

47: BO Interrupt charging

The BO contains the following elements:

Name	Description
Congestion signal from DSO	DSO identifies congestion and sends the signal to the marketplace
EVSP approving the curtailment	EVSP has to confirm or override (charge anyways)
EVSE reducing the throughput or interrupting charging	If the EVSP confirms, the EVSE can interrupt charging

2133: BO Load Curve

The Load Curve BO indicates the average daily load curve of an EVSE or EVSEs within a Load Area, described by a list of average power consumed each quarter of hour in the 24 hours. It's a vector of the sum of the power [kW] of meter readings from the EVSE(s) within a Load Area. The daily load curves can be averaged over different time periods as weekdays, saturdays and holidays.

Attributes	Datatype	Constraints	Description
loadCurve	2391: BO Value Quarter	Multiplicity: 1 Mandatory	List of power each quarter of hour for max 48 hours (two days).
typicalWeekDaysLoadCurve	2392: BO Value Day	Multiplicity: 1 Mandatory	Daily load curves averaged on the week days
typicalSaturdayLoadCurve	2392: BO Value Day	Multiplicity: 1 Mandatory	Daily load curve averaged on saturdays
typicalHolidayLoadCurve	2392: BO Value Day	Multiplicity: 1 Mandatory	Daily load curve averaged on holidays

2138: BO Load Area

The BO LA indicates LV/MV network sub-area, which is dynamically defined by the DSO under certain criteria, for instance, maximum hosting capacity.
The format of a LA can be a list of PODs, within a specific network area (e.g. under LV substations in a defined geographic area) which cumulated capacity does not exceed the maximum hosting capacity.
The aim of defining LAs is to provide DSO and EVSE operators with a common term to univocally identify the location where a congestion or load management need occur.

Attributes	Datatype	Constraints	Description
loadAreaID	string	Multiplicity: 1 Max lenght: 20 char Mandatory	Identification of the load area defined by the DSO
loadAreaValidityStartDate	date	Multiplicity: 1 Mandatory ISO 8601 format YYYY-MM-DDTHH:MM:SSZ	Start date of the validity of the Load Area update
loadAreaValidityEndDate	date	Multiplicity: 1 Optional ISO 8601 format YYYY-MM-DDTHH:MM:SSZ	End date of the validity of the Load Area update. This value is specified only for old updates and not current ones
PODlist	2390: BO POD	Multiplicity: 1..* Mandatory	list of PODs (Points of Delivery) within a specific load area in the DSO's network .The DSO defines this value under specific criteria set by the DSO itself.

122: BO Flexibility of EV batteries

The BO contains the following elements.

Name	Description
Battery ellapsed charge time	When is the expected end of charging for particular battery and cummulated for EVSE location?
Battery current status of charge	The percentage of each battery status for each EVSP location
Drive ranges	How many kilometers do the drivers want to drive
Current cumulated battery load	How many batteries are connected to each EVSP location at every moment
Cumulated ramp up/down potential	How much can each EVSE location ramp up/down at a given moment

2139: BO Load Management Detail Record

The BO LMDRis generated for each load/congestion management event occurred within a recharge transaction and accomplished by the EVSE Op. or EVSP. The LMDR is sent to the DSO in order for him to be rewarded.

Attributes	Datatype	Constraints	Description
LMDRID	string	Multiplicity: 1 Mandatory Max length: 20 char	Unique identification of the load management detail record.
loadAreaID	String	Multiplicity: 1 Mandatory Max length: 20 char	Identication of the load area.
startOfLMR	dateTime	Multiplicity: 1 Mandatory YYYY:MM:DD HH:MM:SS	Time stamp indicating the beginning of the time interval for the load curve
eventDuration	float	Multiplicity: * Mandatory	Duration of LM request as requested by the DSO
realPower	float	Multiplicity: * Mandatory	Vector of values of effective power throughput of EVSEs at the instants t0, t0+15 min, t0 + 30 min, etc, (per each quarter of an hour within the eventDuration)
deltaPower	float	Multiplicity: * Mandatory	Vector of values equal to the difference between DSO target power throughput and EVSE operator effective power throughput at the instants t0, t0+15 min, t0 + 30 min, etc, (per each quarter of an hour within the eventDuration)
priority	enum	Multiplicity: 1 Mandatory value range is from 1 (max priority) to 5 (lowest priority)	indicates the type of urgency/priority of the DSO request to which the EVSE Operator responded

2390: BO POD

The BO POD identifies the connection point to the grid.

Attribute	Data Type	Constraints	Description
podID	String	Multiplicity: 1 Max lenght: 20 char Mandatory	Identification of the Point Of Delivery

126: BO Phase balancing

The BO contains the following elements:

Name	Description
LV substation phase imbalance	Reduction of phase imbalance is requested by DSO
Parking lot phase imbalance	Phase imbalance reduction requested by the EVSE operator or the parking lot operator

118: BO Prolongued charging alert

The BO contains the following elements:

Name	Description
Prolongued charging alert created	When EVSE interupts charging it routes an alert to the Marketplace (EVSP)
Alert is sent to the end user (driver)	EVSP picks the alert and delivers it to the end user (sms, email, etc.) based on the contract option

123: BO Load overview

The BO contains the following elements:

Name	Description
EVSE load overview	how much electricity is each EVSE operator consuming at its charging points: • currently • history • expected future consumption (if available) • overview of CP locations operated by one EVSE operator
EVSP load overview	how much electricity is each EVSP consuming: •currently, history, expected (predictive analysis), •overlap with the EVSE location (whoch EV is charging where)

120: BO Marketplace checks the availability of EVSP and EVSE for curtailment or V2G

The BO contains the following elements:

Name	Description
Availability of EVSE	Marketplace checks the current charging of the EV and the EVSE enabling remote control
Approval of EVSP	Marketplace checks the EVSPs for availability of curtailment or V2G (based on end-user contracts)

2392: BO Value Day

The value day BO indicates the average daily load curve of an EVSE or EVSEs within a Load Area.

Attributes	Datatype	Constraints	Description
valueList	2391: BO Value Quarter	Multiplicity:1..* Mandatory Max Length: 96	List of 4 (one for each quarter of hour) x24 (hour in a day) values of ValueQuerter Type
power	double	Multiplicity:* Mandatory	Value of the power in watt
timestamp	dateTime	Multiplicity:* Mandatory	Timestamp of the power value

2391: BO Value Quarter

The Load Curve BO indicates the average daily load curve of an EVSE or EVSEs within a Load Area, described by a list of average power consumed each quarter of hour in the 24 hours. It's a vector of the sum of the power [kW] of meter readings from the EVSE(s) within a Load Area. The daily load curves can be averaged over different time periods as weekdays, saturdays and holidays.

Attributes	Datatype	Constraints	Description
power	double	Multiplicity:* Mandatory	Value of the power in watt
timestamp	dateTime	Multiplicity:* Mandatory	Timestamp of the power value

5.2 Service Specification

1572: UC Reduce Charge Power by DSO

Business Components	2122: BC EVSE energy back-end 2123: BC DSO Load Management
Services	2017: SRV DSO predefines peaks 2126: SRV Get Load Areas Update from DSO

2122: BC EVSE energy back-end

The EVSE energy back-end BC is run by the Charge Management System and it is used to provide energy related services to the DSO.

Provided Services	2019: SRV Load Management Target 1909: SRV EVSE History Use 2013: SRV Load Curve 2017: SRV DSO predefines peaks 2016: SRV Current EV Charge Data 2126: SRV Get Load Areas Update from DSO 2023: SRV Congestion Management Target
Required Services	2024: SRV Load Management Tracking 2025: SRV Congestion Management Tracking

2019: SRV Load Management Target

Load Management Target service is used by a DSO to "push" a request of load management within a specified Load Area.

LMTARGET request

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1 Mandatory	Used to authorize the call to the WS method
DSOrequestID	string	Multiplicity: 1 Mandatory Max length: 20 char	Unique identification of the load management request form the DSO.
loadAreaID	string	Multiplicity: 1 Mandatory Max length: 20 char	Load area identification
startOfLMR	dateTime	Multiplicity: 1 Mandatory	Timestamp indicating the beginning of the time interval for the load curve. Format UTC/ISO 8601
eventDuration	float	Multiplicity: 1 Mandatory	It is the length of the x-axis of the load curve.
energyType	enum	Multiplicity: 1 Mandatory Max length: 1	Unique number which refers to the energy type delivered by DSO
targetLoadCurve	2133: BO Load Curve	Multiplicity: 1 Mandatory	Maximum forecast depth is 48 hours from the current time. These are the values of target power to be achieved from the EVSE operator backend
priority	integer	Multiplicity: 1 Mandatory value range is from 1 (max priority) to 5 (lowest priority).	1 is the max priority – 5 is the lowest priority

LMTARGET response

Data Element	Data Type	Constraints	Description
DSOrequestID	String	Multiplicity:1 Mandatory Max length. 20 char	Unique identification of the load management request from the DSO
maxTargetTime	dateTime	Multiplicity: 1 Mandatory	This field states till which time the EVSE backend will comply to the DSO request, after this time the EMM backend will not anymore consider the DSO request

1909: SRV EVSE History Use

The SRV is used to retrieve the historical use of an EVSE in a specific time interval.

EVSEHistoryUseRequest

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1 Optional	Used to authorize the call to the WS method
evseID	2127: BO EVSEID	Multiplicity: 1..* Mandatory	t is the identification of a specific EVSE..
startDay	Date	Multiplicity: 1 Optional Format ISO 8601: "YYYY-MM-DD"	Identifies the start of the timeslot in which the data aggregation has to be narrowed inIf it's not provided all data retrieved according to the aggregation criteria will be provided since the first item recorded in the system.
endDay	Date	Multiplicity: 1 Optional Format ISO 8601: "YYYY-MM-DD"	Identifies the end of the timeslot in which the data aggregation has to be narrowed in. If it's not provided all data retrieved according to the aggregation criteria will be provided up to the last item recorded in the system.

EVSEHistoryUseResponse

Attributes	Datatype	Constraints	Description
EVSEhistoryUseList	2394: BO EVSEHistoryUse	Multiplicity: 1...* Mandatory	List of the histroy use.

2013: SRV Load Curve

The SRV is used to retrieve a list of historical use of EVSEs aggregated either per EVSE ID or load area. The information to be provided is specified for a definite time frame and consists of daily typical load curves during weekdays, Saturdays or holydays.

LoadCurveRequest

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1 Mandatory	Used to authorize the call to the WS method
loadAreaID	string	Multiplicity: 1 Mandatory max length: 20 char	loadAreaID as defined in 2138: BO LoadArea. The Load Area is one of the possible aggregation criteria
startDay	Date	Multiplicity: 1 Optional Format ISO 8601: "YYYY-MM-DDThh:mm:ss"	Identifies the beginning of the timeslot in which the data aggregation has to be narrowed in. If it's not provided all data retrieved according to the aggregation criteria will be provided since the first item recorded in the system.

Attributes	Datatype	Constraints	Description
endDay	Date	Multiplicity: 1 Optional Format ISO 8601: "YYYY-MM-DDThh:mm:ss"	Identifies the end of the timeslot in which the data aggregation has to be narrowed in. If it's not provided all data retrieved according to the aggregation criteria will be provided since the first item recorded in the system.

LoadCurveResponse

Attributes	Datatype	Constraints	Description
loadCurve	2133: BO Load Curve	Multiplicity: 1 Mandatory	Load Curve

2017: SRV DSO predefines peaks

The SRV is used by the ES DSO to “push” the list of predefines peaks to manage hosting capacity of LV grid as a preliminary condition for electric mobility infrastructure to be installed and later on operated in a certain load area.

DSOpredefinesPeaks request

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1 Mandatory	Used to authorize the call to the WS method
loadAreaID	string	Multiplicity: 1 Mandatory max Length: 20 char	loadAreaID
peakCurveWinterTime	2133: BO Load Curve	Multiplicity: 1 Mandatory	Target load curve defined by the DSO for winter time for load management request
peakCurveSpringTime	2133: BO Load Curve	Multiplicity: 1 Mandatory	Target load curve defined by the DSO for spring time for load management request
peakCurveSummerTime	2133: BO Load Curve	Multiplicity: 1 Mandatory	Target load curve defined by the DSO for summer time for load management request
peakCurveFallTime	2133: BO Load Curve	Multiplicity: 1 Mandatory	Target load curve defined by the DSO for fall time for load management request

DSOpredefinesPEAKS response

Attributes	Datatype	Constraints	Description
fulfilledDSOpredefinesPeaksAcknowledge	boolean	TRUE/FALSE	If TRUE the request has been fulfilled
falseDSOpredefinesPeaksAcknowledge	string	Multiplicity: 1..*	If acknowledge is FALSE, this field contains the error code. List of predefined error codes.
falseDSOpredefinesPeaksDescription	string	Multiplicity: 1..*	Textual error description. If acknowledge is FALSE, this field contains the error description.

2016: SRV Current EV Charge Data

Retrieves either current or historical information of EVs State of Charge within a specific load area. This is a fundamental service to be run in order to:
- Sustain V2G scenarios because it delivers the first glance picture of power availability on LV grid.
- Perform statistical analysis and create value in the customer use patterns.
- Decide priorities among recharge process according to the customer requests and state of charge of their EV.

EVChargeData Request

Attributes	Datatype	Constraints	Description
loadAreaID	string	Multiplicity: 1	loadAreaID as defined in 2138: BO LoadArea Identifies the Load Area for which the requestor wants visibility of current EV charge data, if load area aggregation criteria is chosen. The load area is one of the possible families of aggregation criteria.
EVdataRangeStartTime	dateTime	Multiplicity: 1 ISO 8601 format YYYY-MM-DDTHH:MM:SSZ	Defines the start of the search range. The range defines the recharges that are/were active in this range. The start time cannot be more than 5 days in the past. If start time is not valued, the service returns only the active recharges
EVdataRangeEndTime	dateTime	• Optional • ISO 8601 format YYYY-MM-DDTHH:MM:SSZ	Defines the end of the search range. If start time has no value, the end time must have no value.

EVChargeData Response

Attributes	Datatype	Constraints	Description
EVchargedataResponse	2146: EV charge data item	Multiplicity: *	Contains data to be forwarded by the system that gathers EV charge data.

2126: SRV Get Load Areas Update from DSO

The SRV is used by DSO to update Load Area.
LAUpdate request

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1	Used to authorize the call to the WS method
loadAreaUpdated	2138: BO Load Area	Multiplicity: 1..*	List of updated load areas

LAUpdate response

Attributes	Datatype	Constraints	Description
ack	integer	Multiplicity: 1	value 1 for succesfull receipt of the message other error values

2023: SRV Congestion Management Target

This SRV is used by a DSO to “push” an urgent congestion issue within a LoadArea and pushes the mandatory level of maximum power.
CMtarget request

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1 mandatory	Used to authorize the call to the WS method
DSORequestID	string	Multiplicity: 1 mandatory Max length. 20 char	Unique identification of the CM request from the DSO.
loadAreaID	string	Multiplicity: 1 Mandatory Max length. 20 char	Load area identification
startOfLMR	dateTime	Multiplicity: 1 Mandatory	Timestamp indicating the beginning of the time interval for the load curve. Format UTC/ISO 8601
eventDuration	float	Multiplicity: 1 Mandatory	It is the length of te x-axis of the load curve.
targetLoadCurve	2133: BO Load Curve	Multiplicity: 1 Mandatory	Maximum forecast depth is 48 hours from the current time. These are the values of target power to be achieved from the EVSE operator backend

Attributes	Datatype	Constraints	Description
priority	integer	Multiplicity: 1 Mandatory max length: 1	1 is the max priority

CMTarget Response

Attributes	Datatype	Constraints	Description
DSOrequestID	string	Multiplicity: 1 mandatory Max length. 20 char	Unique identification of the load management request from the DSO
fulfilledCMtargetAcknowledge	boolean	TRUE/FALSE	if TRUE the EVSE energy back-end has accomplished the congestion management request
errorCode	string	Multiplicity: *	List of predefined error codes. If acknowledge is FALSE, this field contains the error code
errorDescription	string	Multiplicity: *	Textual error description. If acknowledge is FALSE, this field contains the error description

2123: BC DSO Load Management

The DSO Load Management business component is run by the DSO and it is used to request load / congestion management services and other energy related services to the EVSE energy back-end business component.

Provided Services	2024: SRV Load Management Tracking 2025: SRV Congestion Management Tracking
Required Services	2017: SRV DSO predefines peaks 2019: SRV Load Management Target 2126: SRV Get Load Areas Update from DSO 2023: SRV Congestion Management Target

2024: SRV Load Management Tracking

Load Management Tracking service. Derived from Network Congestion / Flexible Load Management use cases.

At the end of each load management transaction, the EVSE operator forwards to the DSO the LMDR in order to be remunerated based on the load shedding details provided in the LMDR.

Verifies that the LM target has been accomplished by either the EVSE Op. or the EVSP in order to fulfill DSO needs and get remunerated.

The SRV is used by an EVSE operator to forward to the DSO the detail record about the load shedding releated to a previous load management request.

LMTRACKING request

Attributes	Datatype	Constraints	Description

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity: 1 Mandatory	Used to authorize the call to the WS method
DSOrequestID	string	Multiplicity: 1 Mandatory max length: 20 char	Unique identification of the load management request from the DSO
declinedAUTH	integer	Multiplicity: 1 Mandatory	Number of declined authorization because of the DSO load mgmt request
sheddingEventsDetails	2139: BO Load Management Detail Record	Multiplicity: 1 Mandatory	Shedding events details

LMTRACKING response

Attributes	Datatype	Constraints	Description
LMDReceiptAcknowledge	boolean	TRUE/FALSE	if TRUE the LMDR had been correctly received by the DSO Load Management BC
errorCode	string	Multiplicity: *	list of predefined error codes . If acknowledge is FALSE, this field contains the error code
errorDescription	string	Multiplicity: *	Textual error description . If acknowledge is FALSE, this field contains the error description

2025: SRV Congestion Management Tracking

Congestion Management Tracking service. Derived from Network Congestion / Flexible Load Management use cases.
At the end of each congestion managent transaction, the EVSE operator forwards to the DSO the LMDR in order to be remunerated based on the load sheddings details provided in the LMDR.
Verifies that the CM target has been accomplished by either the EVSE Op. or the EVSP in order to fulfill DSO needs and get remunerated

CMTrackingRequest

Attributes	Datatype	Constraints	Description
token	2393: BO AuthorizationToken	Multiplicity:1 Mandatory	Used to authorize the call to the WS method
DSOrequestID	string	Multiplicity: 1 Mandatory Max length: 20 char	Unique identification of the load management request from the DSO
declinedAUTH	integer	Multiplicity: 1 mandatory	Number of declined authorization because of the DSO mgmt request
sheddingEventsDetails	2139: BO Load Management Detail Record	Multiplicity: 1 Mandatory	Shedding events details

CMTrackingResponse

Attributes	Datatype	Constraints	Description
LMDReceiptAcknowledge	boolean	TRUE/FALSE	if TRUE the LMDR had been correctly received by the DSO Load Management BC
errorCode	string	Multiplicity: *	list of predefined error codes . If acknowledge is FALSE, this field contains the error code
errorDescription	string	Multiplicity: *	Textual error description . If acknowledge is FALSE, this field contains the error description

1596: UC Peak load threshold on a substation

Services	2126: SRV Get Load Areas Update from DSO

1597: UC Peak shaving

Services	2126: SRV Get Load Areas Update from DSO

1598: UC Aggregated EV charge overview by the DSO

Business Components	2122: BC EVSE energy back-end 2125: BC DSO Mobility Management
Services	2016: SRV Current EV Charge Data

2122: BC EVSE energy back-end

This BC was exported before.

2125: BC DSO Mobility Management

The DSO Mobility Management business component is run by the DSO and it is used to request mobility services to the EVSE energy back-end business component.

Provided Services	
Required Services	1909: SRV EVSE History Use 2013: SRV Load Curve 2016: SRV Current EV Charge Data

1599: UC History of EVSE use

Business Components	2122: BC EVSE energy back-end 2125: BC DSO Mobility Management
Services	1909: SRV EVSE History Use 2013: SRV Load Curve 2016: SRV Current EV Charge Data 2126: SRV Get Load Areas Update from DSO

2122: BC EVSE energy back-end

This BC was exported before.

2125: BC DSO Mobility Management

This BC was exported before.

1601: UC provide balancing capacity

Services	2126: SRV Get Load Areas Update from DSO

1602: UC flexible load for congestion management

Business Components	2122: BC EVSE energy back-end 2123: BC DSO Load Management
Services	2019: SRV Load Management Target 2023: SRV Congestion Management Target 2024: SRV Load Management Tracking 2025: SRV Congestion Management Tracking 2126: SRV Get Load Areas Update from DSO

2122: BC EVSE energy back-end

This BC was exported before.

2123: BC DSO Load Management

This BC was exported before.

1604: UC Vehicle to grid signal

Services	2126: SRV Get Load Areas Update from DSO

1605: UC Reserve and activate ancillary services

Services	2126: SRV Get Load Areas Update from DSO

6 Marketplace

6.1 Domain Model

DM Marketplace Domain

Per definition, the Green eMotion Marketplace is a virtual B2B marketplace for e-mobility related services. Like any other marketplace, the Green eMotion Marketplace allows the market participants to offer, request and trade goods. In contrast to a real marketplace where people need to meet in person, the GeM Marketplace is a virtual marketplace, accessible through the internet and hosted in a cloud environment.
The market participants are parties which operate a business in the area of e-mobility - this includes but is not limited to operators of charging equipment (EVSE Operators), EV service providers who offer services to EV drivers (EVSPs), energy service providers (utilities, DSOs, TSOs, energy retailers), car manufacturers (OEMs) and IT service providers. While the end customers do not have direct access to the GeM Marketplace, they use IT as well as non IT services from the market participants who offer and access IT services on the GeM Marketplace in order to collaborate and ultimately enable end user services.

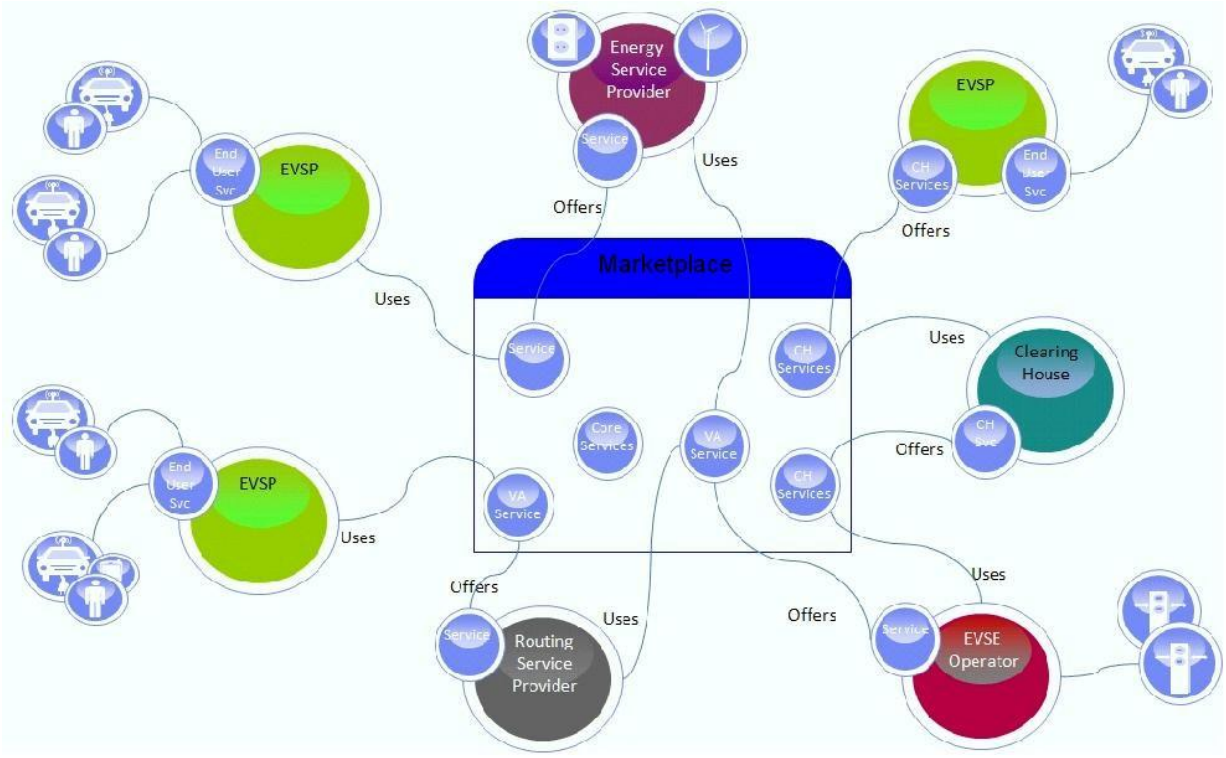


Figure 6.1: Electric mobility stakeholders interconnected via the Marketplace

The Figure illustrates how different market participants can be connected to the Green eMotion Marketplace through backend services and how they provide services through different devices to the end

users. Its sole purpose is to show the variability of potential usage scenarios, while the actual scenarios applicable for each domain have been described in the previous chapters.

The goods being traded are e-mobility related IT services. In other words, no physical goods like charging equipment or electric vehicles are being traded, not even energy. Instead IT services which support the processes required to perform the e-mobility related business operations within and across companies are the subject of offers and requests.
The business partners as well as services, offering, contracts and many more have been modelled as business objects in a single domain model which serves as a common repository for all use cases, business components and services within the marketplace domain.
The GeM marketplace can logically be split into two major areas:

- the technical part, which performs the actual execution of the IT services across different parties
- the commercial part, which supports the processes to register parties and offer/contract services

In GeM the technical part is called dynamic service gateway (DSG). From a high level perspective the DSG can be looked at as a central service hub which interconnects the backend services from all marketplace participants. Instead of just routing the requests from one party to another, it identifies the requesting party and - depending on contractual relationships between the parties - it authorizes and forwards the request or denies it. All forwarded request are tracked in order to provide a transaction history to the parties involved.
DSG is described through the Business Component 1965 Service Brokerage and the Service Interface 1968 Dynamic Service Gateway.
The commercial part provides a user interface to the market place participants and offers services in the following functional areas:

- Registration and maintenance of business partner/user accounts
- Authentication (login/logout)
- Creation and maintenance of service categories (standard services)
- Creation and publication of service offers
- Creation/Termination of contracts between business partners
- Search functionalities for categories, services and contracts
- Transaction details for service usage

The functional areas are modelled in several use cases which are all be implemented through the Business Component 1966 Business Services. A whole set of sketches illustrates how a potential user interface supporting the individual use cases could look like.
The domain model of the marketplace is illustrated in the following picture.

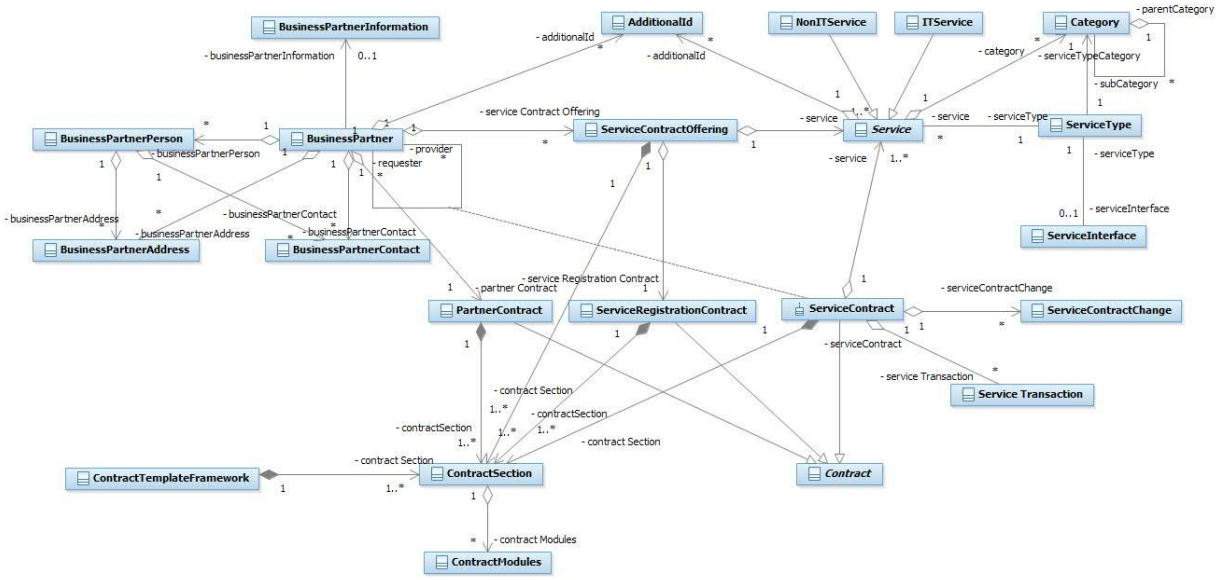


Figure 6.2: Marketplace Domain model

The picture shows all business objects of the marketplace domain model as well as the relations between them. The numbers at the end of each relationship arrow denote the multiplicity

- 1 means exactly one
- 0..1 means none or one
- * means many (including none)
- 1..* means at least one)

As an example one can see, that a BusinessPartner can have as many ServiceOfferings as he likes, but he does not need to have one.
In order to focus on the structure of the domain model, the attributes of the business objects have not been included in figure 6.2. Instead they will be explained in the following sections.

2076: BO Contract

The business object "BO Contract" is a data container representing all types of contracts as an abstract object.

Attributes	Datatype	Constraints	Description
contractId	string	• 7 alphanumeric characters • Required	Identifier of the contract
contractDate	dateTime	• Required	Timestamp of the last contract change.

Attributes	Datatype	Constraints	Description
validFrom	dateTime	• Optional	Timestamp of the start date of the contract. validFrom can be left empty when creating a ServiceContract or a ServiceContractOffering. The validFrom date will be set on approval (to the current date). If the duration is set also the validThrough date will be set on approval (to current date + duration). After the validThrough date is set, this will be the relevant attribute to determine if the contract is valid. The duration will be kept for information only. It is not possible to set both the duration and the validThrough date.
validThrough	dateTime	• Optional	Timestamp of the end date of the contract. Either the duration or the validThrough date must be set (for ServiceContract and ServiceContractOffering)
duration	duration	• Optional	The duration of the contract. Either the duration or the validThrough date must be set (for ServiceContract and ServiceContractOffering).

1460: BO ServiceContractOffering

The business object "BO ServiceContractOffering" is a data container representing a service offer of a Business Partner (Service Provider) on the marketplace.

Attributes	Datatype	Constraints	Description
offeringId	string	• 7 alphanumeric characters • Required	Identifier of the offering
offeringDate	dateTime	• Required	Timestamp of the last offering change.
validFrom	dateTime	• Required	Timestamp of the start date of the offering.
validThrough	dateTime	• Optional	Timestamp of the end date of the offering.
offeringStatus	string	• Enumeration • Required	Status of the offering. Possible status in Release 1: • Submitted • Published
contractSection	2079 BO ContractSection	• Multiplicity: *	Text sections of the contract offering

Attributes	Datatype	Constraints	Description
service	544 BO Service	• Multiplicity: *	Multiplicity changed from 1 to 1..* in order to support FTR 2293 Service Bundles
serviceRegistrationContract	1464 BO ServiceRegistrationContract	• Multiplicity: 1	Contract between the service provider and the marketplace.

1996: BO BusinessPartnerAddress

The business object "BO BusinessPartnerAddress" is a data container for an address used by the BusinessPartner or by the BusinessPartnerPerson.

Attributes	Datatype	Constraints	Description
addressType	string	• Enumeration: "Postal Address", "Street Address" • Required	Type of the specified address, i.e. postal address (address with pobox) or street address.
street	string	• Max. length: 100 characters • Optional, but required on addressType "Street Address"	First address line.
street2	string	• Max. length: 100 characters • Optional	Second address line.
poBox	string	• Max. length: 50 characters • Optional, but required on addressType "PostalAddress"	Post office box.
city	string	• Max. length: 50 characters • Required	Town or city of the address.
zip	string	• Max. length: 10 characters • Required	Postal code of the address
region	string	• Max. length: 50 characters • Optional	State or province of the address

Attributes	Datatype	Constraints	Description
countryCode	string	• Max. length: 2 characters • Required	Country code defined by ISO 3166-1

1461: BO BusinessPartner

The business object "BO BusinessPartner" is a data container for a legal entity (company) registered at the marketplace acting as service provider and/or service requester.

Attributes	Datatype	Constraints	Description
partnerId	string	• 19 numeric characters • Required	Identifier of the Business Partner
additionalId	2064 BO AdditionalId	• Multiplicity : *	Additional Ids of the Business Partner, e.g. EVCOID, EVSEID
dn	string	• 150 alphanumeric characters • Required	Distinguished name of the Business Partner
companyName	string	• Max. length: 50 characters • Required	Official name of the Business Partner
companyAcronym	string	• Max. length: 5 characters • Optional	Acronym of the company's name
legalRegistration	string	• Max. length: 100 characters • Optional	ID of the local legal registration
status	string	• Enumeration: "R", "A", "I" • Required	Status of the Business Partner: • R: Registered, unconfirmed • A: Active, confirmed • I: Inactive, confirmed
businessPartnerAddress	1996 BO BusinessPartnerAddress	• Multiplicity : 1..*	Addresses of the Business Partner
businessPartnerContact	1997 BO BusinessPartnerContact	• Multiplicity: 1..*	Contacts of the Business Partner

Attributes	Datatype	Constraints	Description
businessPartnerInformation	2003 BO BusinessPartnerInformation	• Multiplicity: 1	Additional information on the Business Partner
businessPartnerPerson	2004 BO BusinessPartnerPerson	• Multiplicity: 1..*	Users of the Business Partner
provider	1461 BO BusinessPartner	• Multiplicity : *	Business Partners, acting as providers against the given Business Partner, connected by the 547 BO Service Contract.
requester	1461 BO BusinessPartner	• Multiplicity : *	Business Partners, acting as requesters against the given Business Partner, connected by the 547 BO Service Contract.
partnerContract	1462 BO PartnerContract	• Multiplicity: 1	Registration contract of the marketplace.
serviceContractOffering	1460 BO ServiceOffering	• Multiplicity: *	Set of the service offerings of the Business Partner.

1997: BO BusinessPartnerContact

The business object "BO BusinessPartnerContact" is a data container for the contacts of a BusinessPartner or a BusinessPartnerPerson.

Attributes	Datatype	Constraints	Description
id	string	• Required	Unique identifier of the contact
contactType	string	• Enumeration: "Email", "Phone (Office)", "Phone (Mobile)", "Fax" • Required	Type of the specified contact.
contact	string	• Max. length: 100 characters • Required	Phone number or email address

1462: BO PartnerContract

The business object "BO PartnerContract" is a data container representing contract type which has to be accepted during the registration of a Business Partner.

Attributes	Datatype	Constraints	Description
contractStatus	string	• Enumeration • Required	Status of the contract

Attributes	Datatype	Constraints	Description
contractSection	2079 BO ContractSection	• Multiplicity: *	Text sections of the contract

1464: BO ServiceRegistrationContract

The business object "BO ServiceRegistrationContract" is a data container representing contract type which has to be accepted during the registration of a service by the Business Partner (Service Provider).

Attributes	Datatype	Constraints	Description
contractStatus	string	• Enumeration • Required	Status of the contract
contractSection	2079 BO ContractSection	• Multiplicity: *	Text sections of the contract

2007: BO NonITService

The business object "BO NonITService" is a data container representing a single non-IT related service offered at the marketplace. It is used to handle contracts that does not cover IT services, e.g. Roaming Contracts.
The object itself provides no additional attributes upon the attributes provided by the parent BO Service Object.

2004: BO BusinessPartnerPerson

The business object "BO BusinessPartnerPerson" is a data container for an employee (user) of a company registered at the marketplace acting as service provider and/or service requester.

Attributes	Datatype	Constraints	Description
personId	string	• 7 alphanumeric characters • Required	Identifier of the BusinessPartnerPerson (User)
dn	string	• Max. length: 150 characters • Required	Distinguished name of the user
firstName	string	• Max. length: 50 characters • Required	Firstname of the user
middleName	string	• Max. length: 50 characters • Optional	Middlename of the user

Attributes	Datatype	Constraints	Description
lastName	string	• Max. length: 50 characters • Required	Lastname of the user
nameSuffix	string	• Max. length: 10 characters • Optional	Name suffix of the user, e.g. Jr., Sr., ...
role	string	• Enumeration: "Administrator" • Required	In release 1 there is only one user per company with the administrator role
department	string	• Max. length: 50 characters • Optional	Department of the user
preferredLang	string	• 2 characters	Corresponding language id to ISO 639-1
businessPartnerAddress	1996 BO BusinessPartnerAddress	• Multiplicity : *	Addresses of the user
businessPartnerContact	1997 BO BusinessPartnerContact	• Multiplicity: *	Contacts of the user
preferredContact	1997 BO BusinessPartnerContact	• Multiplicity: 1	Preferred contact of the user

2064: BO AdditionalId

The business object "BO AdditionalId" is a data container for additional IDs represented as key/value pairs.
On a conceptual level the concept allows to cover a large number of different idTypes, only restricted by the length of the idType field. While at the beginning of GeM phase 2 only two types are known to be required and even during the lifespan of GeM the number of different idTypes will be rather small (most likely less than 10), it is nevertheless highly recommended to maintain the flexibility of the concept in the actual implementation.

Attributes	Datatype	Constraints	Description
idType	string	• Max. length: 20 characters • Required	Key of the ID Currently known and defined key values: EVCOProviderId (contained in 2165: BO EVCOID) EVSEOperatorId (equivalent to 2130: BO EVSEOperatorID) (those values are just the first two of an extendable list of idTypes)

Attributes	Datatype	Constraints	Description
idValue	string	• Max. length: 20 characters • Required	Value of the ID.

2003: BO BusinessPartnerInformation

The business object "BO BusinessPartnerInformation" is a data container for additional information on a Business Partner (company).

Attributes	Datatype	Constraints	Description
companyDescription	string	• Max. length: 1000 characters • Optional	Description of the company
companyUrl	string	• Max. length: 150 characters • Optional	URL of the website of the company
companyType	string	• Enumeration • Optional	Possible values are, e.g.: • Energy Retailer • OEM • Non-profit organization • ...
companyEmployees	string	• Enumeration • Optional	Possible values are, e.g.: • 1 -10 Employees • 10 - 100 Employees • 100 -1000 Employees • more than 1000 Employees
companyRevenue	string	• Enumeration • Optional	Possible values are, e.g.: • less than 1 M € • 1 M - 10 M € • 10 M - 100 M € • more than 1000 M €

2069: BO Category

The business object "BO Category" is a data container representing categories in a hierarchical structure, e.g. service category, region.

Attributes	Datatype	Constraints	Description
categoryId	string	• Max. length: 5 characters • Required	Identifier of the category entry

Attributes	Datatype	Constraints	Description
categoryValue	string	• Max. length: 100 characters • Required	Value of the category entry
parentCategory	2069 BO Category	• Multiplicity : 0..1	
subCategory	2069 BO Category	• Multiplicity : *	

2006: BO ITService

The business object "BO ITService" is a data container representing a single IT related service offered at the marketplace.

Attributes	Datatype	Constraints	Description
input	string	• Max. length: 1000 characters • Required	Textual description of the input parameters.
output	string	• Max. length: 1000 characters • Required	Textual description of the output parameters.

544: BO Service

The business object "BO Service" is a data container representing a single service offered at the marketplace.

Attributes	Datatype	Constraints	Description
serviceId	string	• 7 alphanumeric characters • Required	Identifier of the service
additionalId	2064 BO AdditionalId	• Multiplicity : *	Additional Ids of the service
subject	string	• Max. length: 100 characters • Required	Subject of the service
description	string	• Max. length: 1000 characters • Required	Textual description of the service

Attributes	Datatype	Constraints	Description
category	2069 BO Category	• Multiplicity : *	Different categories for ordering services, e.g. • Service Category: Charging, Roaming • Region • Target Customer
serviceType	2087 BO ServiceType	• Multiplicity: 1	Service type of the service

554: BO ServiceInterface

The business object "BO ServiceInterface" is a data container representing the service interface of an IT service.

Attributes	Datatype	Constraints	Description
serviceIfcId	string	• 7 alphanumeric characters • Required	Identifier of the service
wsdlFileUrl	string	• Max. length: 500 characters • Required	URL directing to the WSDL file of the service interface.
service	2087 BO ServiceType	• Multiplicity : 1	Service type implementing the service interface.

546: BO ServiceTransaction

The business object "BO ServiceTranaction" is a data container representing log data of the call of a service.

Attributes	Datatype	Constraints	Description
transactionId	string	• Max Length: 50 characters • Required	Identifier of the transaction
transactionDate	dateTime	• Required	Timestamp of the completed transaction
transactionStatus	string	• Required	Status of the completed transaction
transactionText	string	• Optional	Additional textual information of the transaction.
serviceContract	547 BO ServiceContract	• Multiplicity: 1	Reference to the corresponding contract.

547: BO ServiceContract

The business object "BO ServiceContract" is a data container representing contract type which is created between two Business Partners (Service Provider and Service Requester).

Attributes	Datatype	Constraints	Description
contractStatus	string	• Enumeration • Required	Status of the contract
contractSection	2079 BO ContractSection	• Multiplicity: *	Text sections of the contract
service	544 BO Service	• Multiplicity: 1	In further releases the multiplicity will change to 1..*
serviceTransaction	546 BO ServiceTransaction	• Multiplicity: *	Set of transactions belonging to the contract.
serviceContractChange	2116 BO ServiceContractChange	• Multiplicity: *	Set of change objects belonging to the contact.

2087: BO ServiceType

The business object "BO ServiceType" is a data container representing a type for services offered at the marketplace. In case of IT Services a service type is characterized by its service interface. For Non-IT Services the service interface remains null.

Attributes	Datatype	Constraints	Description
serviceTypeId	string	• 7 alphanumeric characters • Required	Identifier of the service type.
description	string	• Max. length: 1000 characters • Required	Textual description of the service type.
standardType	boolean	• Required	Indicator, if the service type represents a standard or not, i.e. a standard type with a service interface represents a standard interface.
category	2069 BO Category	• Multiplicity : 1	The main category for ordering services, e.g. Service Category: Charging, Roaming
serviceInterface	554 BO ServiceInterface	• Multiplicity: 0..1	Service interface that characterizes the service type in case of an IT service.
service	544 BO Service	• Multiplicity : *	Services belonging to the service type.

Multi Marketplace Domain Model

The major aim of the multi market place approach is to allow service provider offering their services to more partners and to allow service requester selecting between more services. Basically this is not mandatory but a decision influenced by the service provider and the involved market places as well as the compatibility of the services between the market places. The latter aspect should be reduced by a harmonization of interfaces, which is currently supported by the industry movement of eMI3 group (http://emi3group.com/).
As pointed out in the deliverable 3.4 there are multiple ways to realize the multi market place environment. For practical reason within Green eMotion several assumptions were made:

Concerning the Market Place:

1. The communication is done via the market places, which allows aggregating communication resp. to reduce the number of connections to be established.

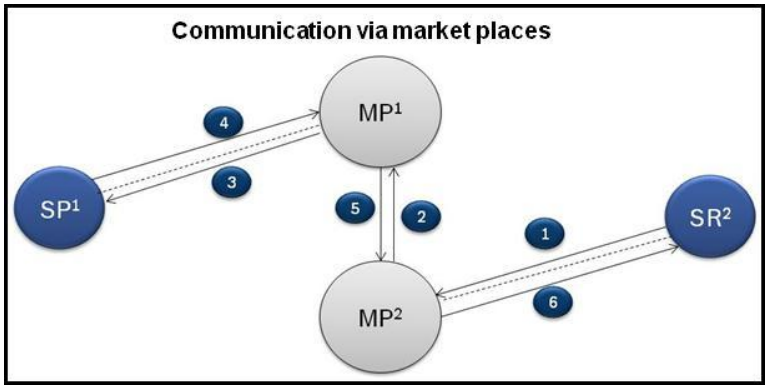


Figure 6.3: Communication via market places

2. Contracts can be aggregated by the market place but bilateral contracts are possible. This allows facilitating the contracting and reduces the number of contracts to be established.

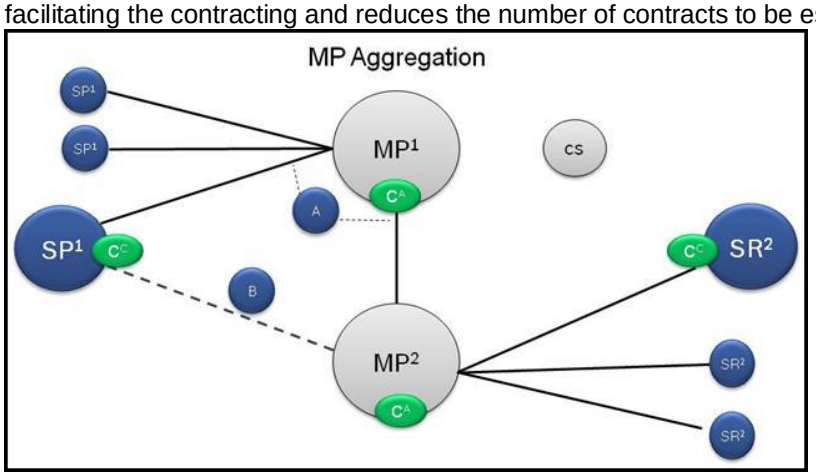


Figure 6.4: Contract aggregation via the market place

3. As pointed out in D3.4 in the long run the services on the different market place may converge into one overall market place. So the intention is to demonstrate also the integration of core services from two or

more market places. This also implies that a single core services on one market place can be used by partners from different market places.

4. Assuming that in the beginning market places compete on the number of partners to gain size and relevance, it is obvious that by default market places are not keen to share their customer base with other market places. However additional services may be used to circumvent this limitation.

5. In order to identify each service and assign it to a corresponding market place or hosting environment, each service offered by a business partner must have a unique ID. Each service with a unique ID can be registered (offered) only on one market place.

6. The service provider can decide if he would like to publish the service on other connected market places or not. As an optional feature one market place may offer to his service provider to select the market places where the service should be offered. One pre condition is that the service is compatible with similar services offered on the other market places or that the service is new to the other market place.

7. In all cases bilateral contracts must be possible with all partners on the market including partners from other market places.

E.g. EDF (connected to MP1) offers a special service to ENEL (connected to MP2).Based on the basic steps to establish a contract for a service there are mainly two service interfaces. First to publish process for services: a) create offer (SRV: PushServiceOffer), b) create contract ((SRV: PushServiceContract). Both service can be also used for update and delete purposes. In order to circumvent the challenge of implementing different service interfaces, a converter approach is proposed. The converter connects the marketplaces and translates the corresponding messages between the market places. This implies that some functionality may not be offered or in a reduced manner on other market places due to incompatibility of the service interfaces. However, with regard to the basic service interfaces for create offer and create contract it is assumed that for those a conversion is not necessary. Nonetheless these requests are routed via the converter to assure a corresponding converter is in place.

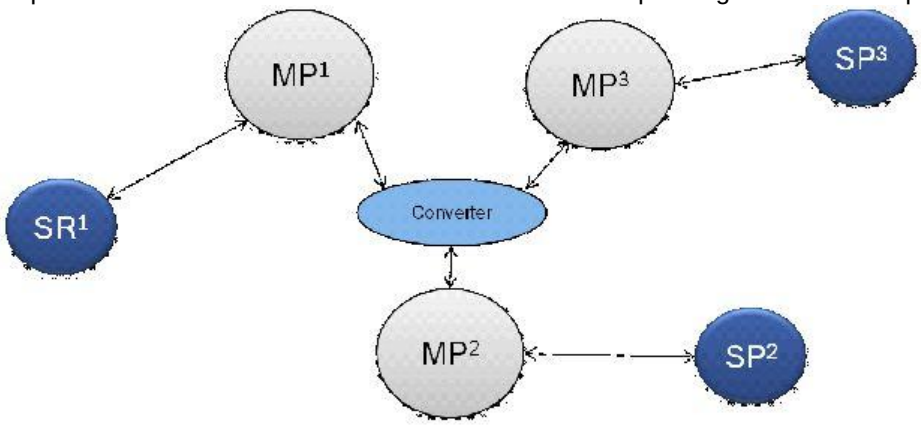


Figure 6.5: Converter between market places

6.2 Functional Decisions

2143: FD Search for Services

Affected Use Cases	1239: UC Search and Select Services

Due to the fact, that there is a very limited number of services available in Release 1, a search functionality will not be implemented. Instead a list of available services is presented to the user.

2147: FD Search of Business Partner

Affected Use Cases	1373: UC Search and Select Business Partner

In the first release the search functionality of a business partner will be used internally. There will be no user interfaces.

2159: FD Transaction Details Not Required

Affected Use Cases	1471: UC View Service Transaction Details

Due to the fact that there will only be a minimal set of data recorded with service transactions in release 1, there will be no detailed view of single service transactions. Instead all service transactions will be listed in a table that shows all necessary information for each service transaction. For future releases this might change when additional data is recorded with service transactions.

2166: FD Use of existing administrative tools

Affected Use Cases	1255: UC Start/Stop Service 1250: UC Register Service

Use cases that exclusively describe administrative tasks and do not involve actors of the business partner hierarchy (see Actors defined in D3.3 chapter 8.2) will not be specified if the functionality that is described can be provided by existing administrative tools.

2416: FD Communication between Marketplaces

Affected Use Cases	1239: UC Search and Select Services

Communication is done initially via the marketplace, at a later point in time communication may take place directly. Anyhow, bilateral communication between partners is possible in both scenarios.

2417: FD Contracting between Marketplaces

Affected Use Cases	1239: UC Search and Select Services

Due to the current market development stage, contracts are aggregated by the market place to reduce the complexity. However bilateral contracts are possible.

2418: FD Core Services Integration of two markets

Affected Use Cases	1239: UC Search and Select Services

Core services between market places can be integrated

6.3 Service Specification

1239: UC Search and Select Services

Related Functional Decisions	2143: FD Search for Services 2416: FD Communication between Marketplaces 2417: FD Contracting between Marketplaces 2418: FD Core Services Integration of two markets

1241: UC Call of Service

Business Components	1965: BC Service Brokerage 2407: BC Service Converter
Services	1968: SRV DynamicServiceGateway

1965: BC Service Brokerage

The "Service Brokerage" component relates to the Service Brokerage Sub Domain of the Marketplace described in the GeM Reference Architecture. The component will implement the following functionality and expose it to users of the marketplace through Service Interfaces:

- Service Requester Access Gateway: It is the initial entry point for Service Requesters to access Services offered on the Marketplace.
- Service Proxy Environment: It manages the interactions with Services executed outside the Marketplace.
- Service Selection Engine: This is a service allowing Service usage based on certain criteria. Service interfaces may contain certain properties. Based on these properties, services can be selected. In this case the Service Requester requests a certain service and a certain criteria just from the Marketplace and not from a special Service Provider. The Selection Engine selects an appropriate Service based on the given criteria and the subscriptions of the Service Requester. Release 1 scope: For Release 1 the Service Selection Engine will only be able to select a service based on the Service ID and the Provider ID that has to be specified when calling a service.
- Not in scope: Service Aggregation Engine is currently not in scope.

Provided Services	1968: SRV DynamicServiceGateway
Required Services	

1968: SRV DynamicServiceGateway

The "DynamicServiceGateway" service will allow marketplace users to call services that they contracted on the Marketplace. The service will take care of authentication of the Service Requester, will check if a valid contract with the Service Provider exists for the service that is called and will then route the service call to the Service Provider. Additionally the service will create a Service Transaction for the service call.
This Service Interface differs from other Service Interfaces that are provided by or offered on the marketplace, because it will not directly be used in the implementation of client applications. Those applications will be created using the Service Interface (WSDL) provided by the Service Provider. To use the Marketplace DynamicServiceGateway only the URI of the service call will be modified to the address of this marketplace service. The URI will also contain the required parameters for selection of the correct Service by the Marketplace.
Authentication will be required using the HTTPS protocol with a valid certificate that is associated with the Marketplace account of the Service Requester. The partnerId of the Service Requester, as well as all additional Ids (idType and value) defined for this Business Partner, will be added to the service call to the Service Provider as HTTP "X-On-Behalf-Of" header attribute, so that the Service Provider can identify the source of the request. The Service Provider can use this for example to provide user specific data via a service offered on the marketplace. However the Service Provider does not need to implement an

authorization mechanism itself based on this attribute, since this is already taken care of by the marketplace.
Notes:

- For the demonstration (at least for the first release) only SOAP 1.1/1.2 over HTTPS will be supported as protocol for service calls. This is to enable maximum compatibility for the demonstration with justifiable efforts. For a commercial marketplace multiple other protocols should be supported to allow as many business partners as possible to integrate new and existing systems.
- Although this service will be involved in most UCs, it will not be explicitly included in the specification. In general all services outside the marketplace domain can be invoked using the "DynamicServiceGateway" service.

DynamicServiceGatewayRequest

Attributes	Datatype	Constraints	Description
provider	String		Matches the partnerId defined in 1461: Bo BusinessPartner. This identifies the Service Provider for the service that will be called. The attribute will be passed to the DynamicServiceGateway Service as parameter in the URI and will not be part of the payload of the message.
alternativeProviderId	2064: BO AdditionalId	Multiplicity: 0..1 Either "provider" or "alternativeProviderId" must be set	Can be used to identify the Service Provider if the Business Partner ID is unknown. idType and value must be provided in order to match an AdditionalId. (e.g. to be used in the Roaming scenario where only the EVCOProviderId is known which is part of the EVCOID (suggested for standardization))
service	String		ID of service to call Either the ID of an individual service offered on the marketplace or the ID of a standard interface. The attribute will be passed to the DynamicServiceGateway Service as parameter in the URI and will not be part of the payload of the message.
payload	xsd:anyType		Payload of service that will be forwarded to the Service Provider. The content depends on the Service Interface of the Service that is called.

DynamicServiceGatewayResponse

Attributes	Datatype	Constraints	Description
result	xsd:anyType		Result of the service call to the Service Provider

Exceptions

Name	Description

Name	Description
ProviderNotAvailable	The Service Provider identified by the provider attribute of the Request is not available in the Marketplace.
ServiceNotAvailable	The Service identified by the service attribute of the Request is not available in the Marketplace.
NoValidContractExists	No valid contract exists between the Business Partner calling this Service and the Business Partner identified by the provider attribute of the Request for the Service identified by the service attribute of the Request.

2407: BC Service Converter

The "Service Converter" component relates to a system that facilitates the communication between market places (Service brokering components). In order to overcome the current situation, of different market places using different interfaces, the component aims to convert and/or to translate the messages. Due to incompabilities of the interfacese there may be limitation in the functionality and in the worst case the functionality cannot be converted. In the future a harmonization of the interfaces is expected especially supported by the eMI3 group (www.emi3group.com).
The component will implement the following functionality and expose it to marketplaces wishing to communicate with each other through the service interfaces:
Service PushServiceInformation: Is used to publish a service offer on another market place. The service offer can be contracted by the connected user of this market place. In other word user of a market place can buy a service from another market place. The service can be also used to manage the service details or to modify, cancel the offer.
Service PushContractInformation is used to inform the originating market place (where the service was created first) about a user of another market place having agreed to the offer and established a contractual realtionship. The service can be also used to modify and cancel the contract.
Before market places can communicate they must accept the terms and conditions of the other market place.

Provided Services	2412: SRV PublishContract 2411: SRV PublishService
Required Services	

2412: SRV PublishContract

The Publish Contract interface deals with requests from Service Brokering Components (market places) to create, read, update delete the Contract information via the service converter to another service brokering component. Both sides must implement the service. For this service no conversion of the message is intended.
PublishContractRequest

Attributes	Datatype	Constraints	Description
ContractId	string	1	
BusinessPartnerID	1461 BO Business Partner	1	
BusinessPartnerID	1461 BO Business Partner	1	
ServiceID	ServiceID	1	
BusinessContract	BO BusinessContract	1	
sessionID	string	1	
transactionId	string	1	

PublishContractResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	
sessionId	string	1	
statusCode	string	1	
statusDescription	string	1	

2411: SRV PublishService

The Publish Service interface deals with requests from Service Brokering Components (market places) to create, read, update delete service offers on ohter market places via the service converter to another service brokering component. Both sides must implement the service. For this service no conversion of the message is intended.

PublishContractRequest

Attributes	Datatype	Constraints	Description
serviceId	string	1	
service	BO Service	1	
serviceContract	BO ServiceContract	1	
serviceInterface	BO ServiceInterface	1	
offeringBusinessPartner	BO BusinesPartner	1	
RetunContract	Boolean	1	

PublishContractResponse

Attributes	Datatype	Constraints	Description
transactionId	string	1	
sessionId	string	1	
statusCode	string	1	
statusDescription	string	1	

1250: UC Register Service

Business Components	2407: BC Service Converter

Related Functional Decisions	2166: FD Use of existing administrative tools

2407: BC Service Converter

This BC was exported before.

1255: UC Start/Stop Service

Business Components	2407: BC Service Converter

Related Functional Decisions	2166: FD Use of existing administrative tools

2407: BC Service Converter

This BC was exported before.

1373: UC Search and Select Business Partner

Related Functional Decisions	2147: FD Search of Business Partner

1471: UC View Service Transaction Details

Related Functional Decisions	2159: FD Transaction Details Not Required

1491: UC Create Service Transaction

Business Components	1965: BC Service Brokerage 2407: BC Service Converter
Services	1968: SRV DynamicServiceGateway

1965: BC Service Brokerage

This BC was exported before.

2407: BC Service Converter

This BC was exported before.

6.4 User Interface Specification

Unlike all other parts of this document, the sketches being provided within this User Interface Specification are not binding requirements for any potential implementation.
Instead they just aim to provide an idea how a user interface for certain Use Cases could look like.

1239: UC Search and Select Services

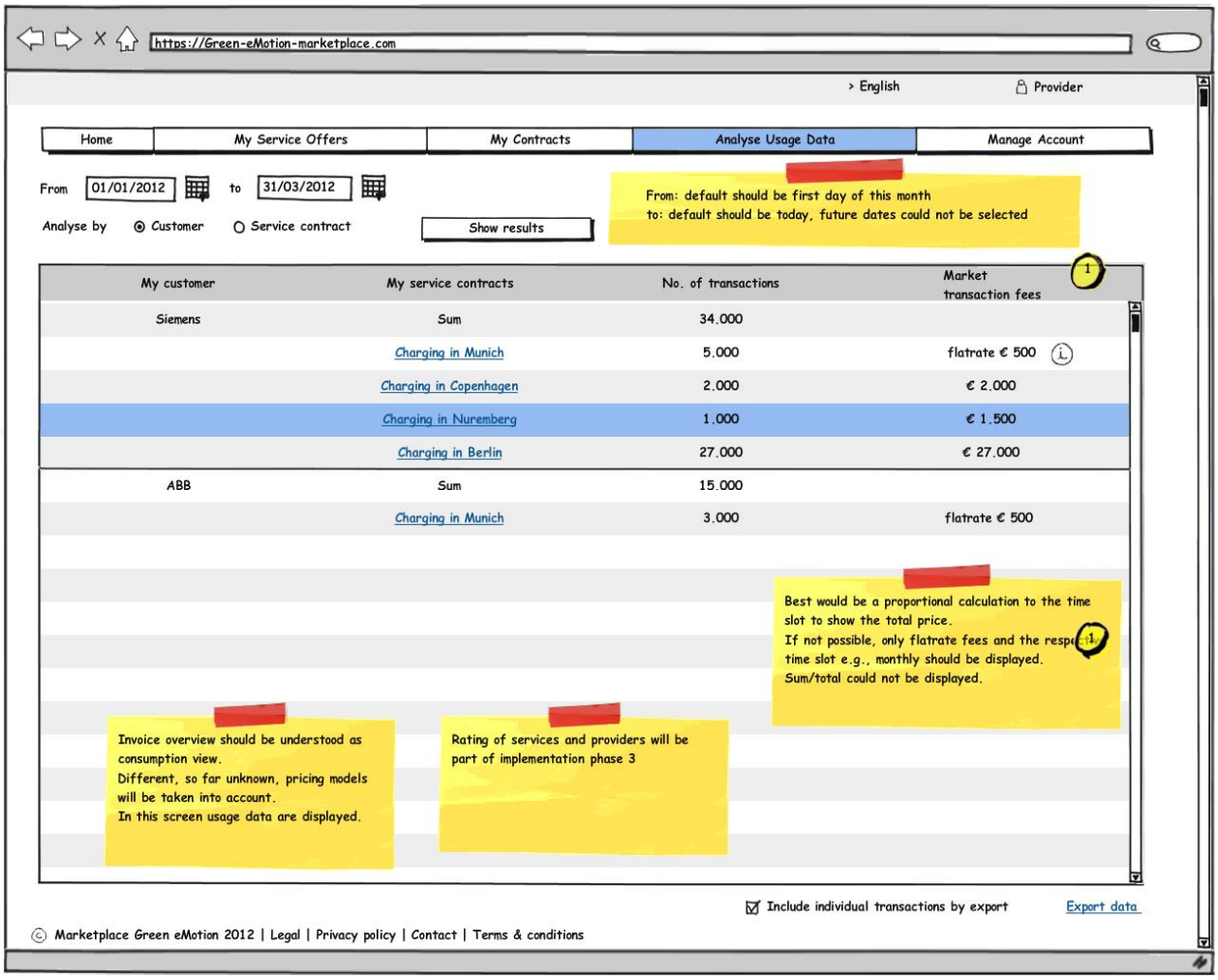
Related Functional Decisions	2143: FD Search for Services 2416: FD Communication between Marketplaces 2417: FD Contracting between Marketplaces 2418: FD Core Services Integration of two markets

1242: UC Search and Select Service Transactions

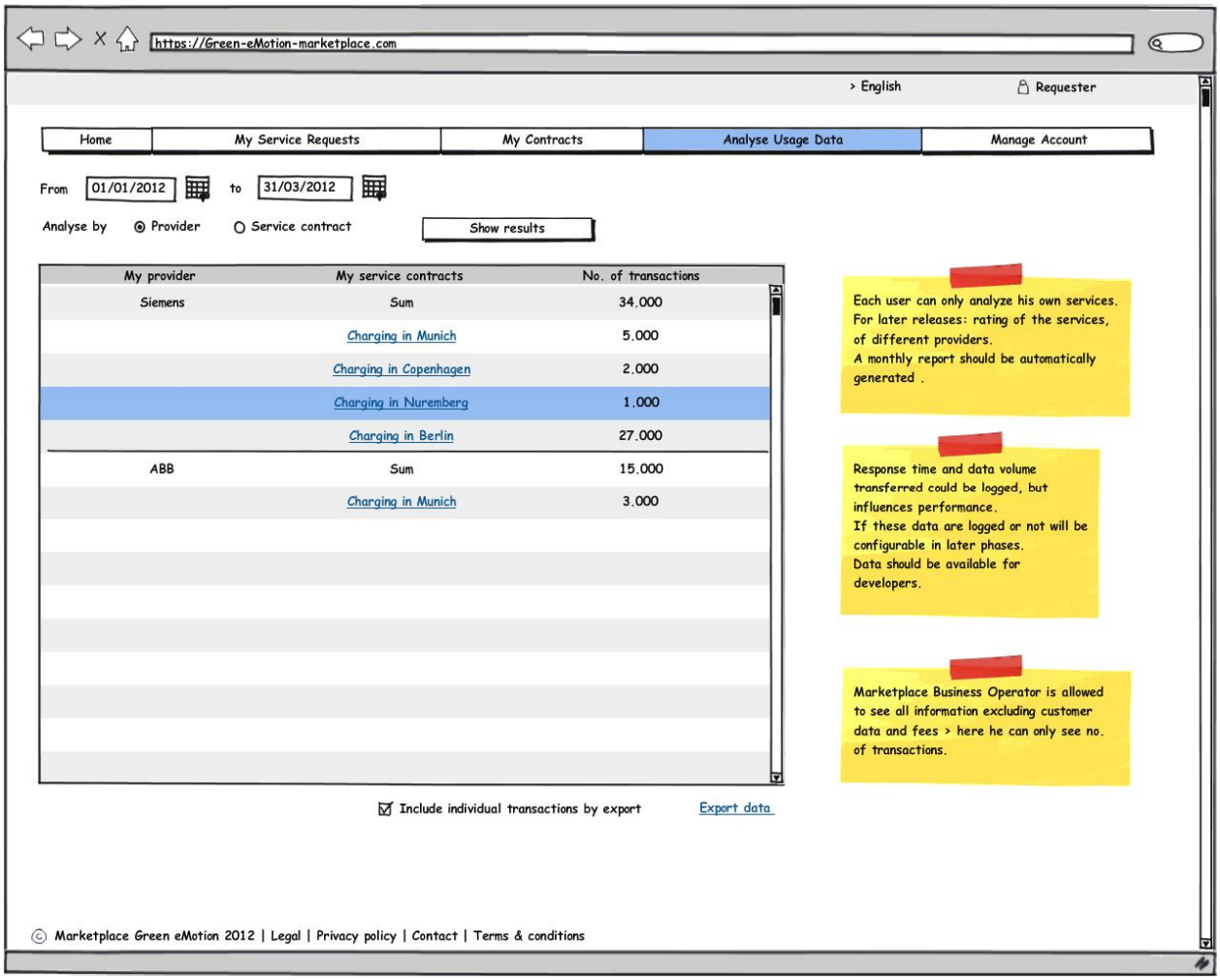
1966: BC Business Services

This BC was exported before.

2101: SKT Search and Select Service Transactions (Provider)



2158: SKT Search and Select Service Transactions (Requester)

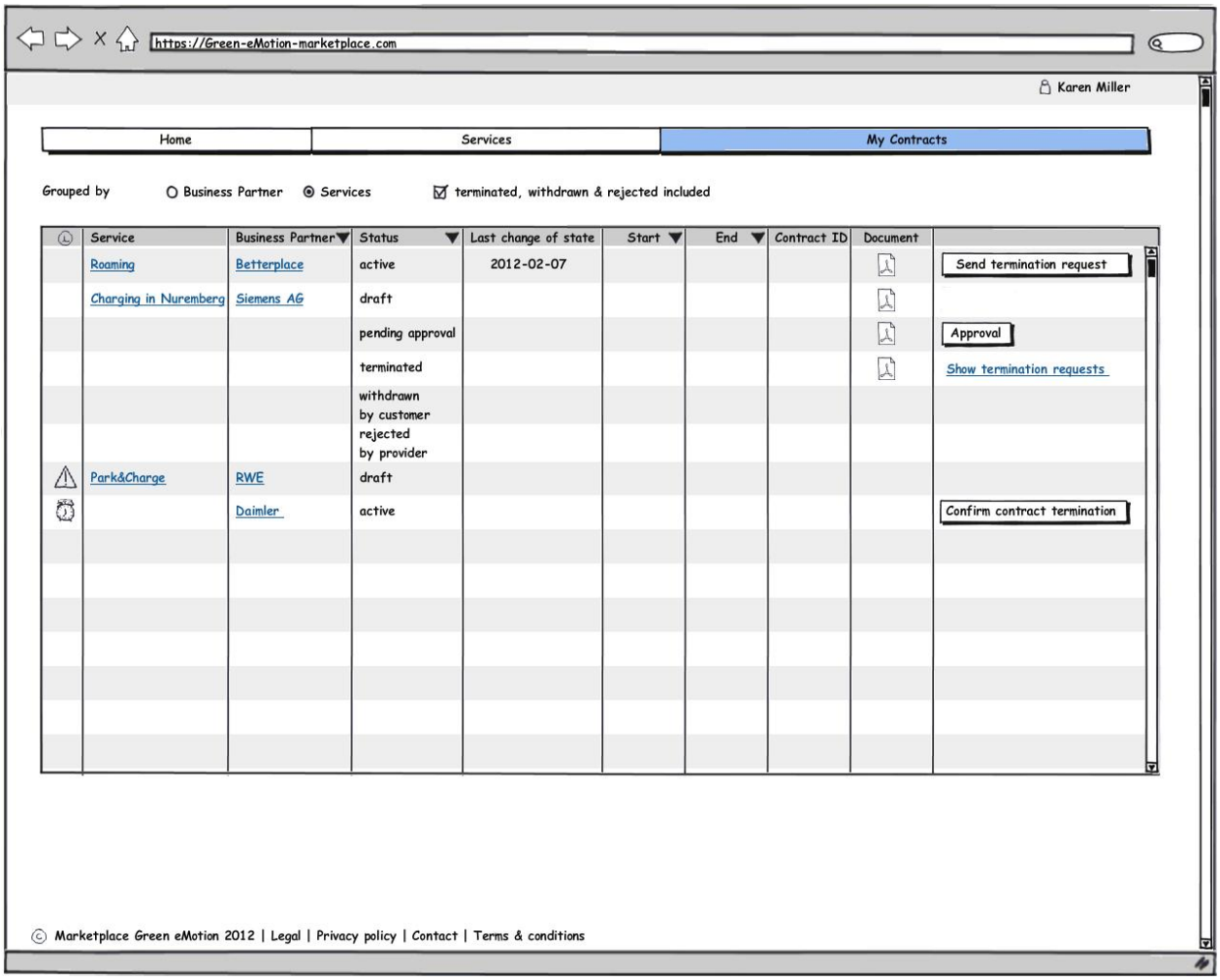


1245: UC Search and Select Service Contracts

1966: BC Business Services

This BC was exported before.

2098: SKT Search and Select Service Contracts



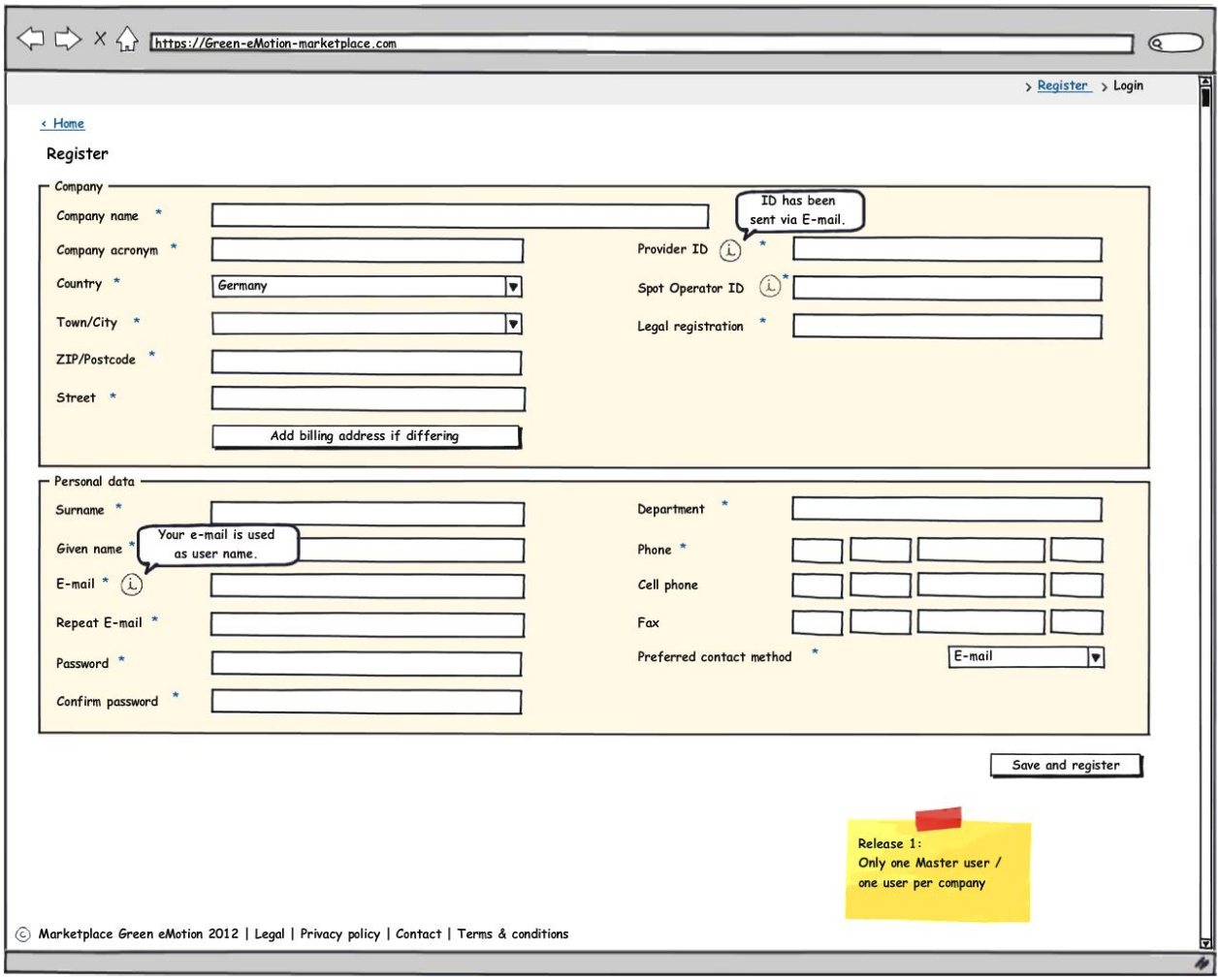
Restrictions:
The status "Withdrawn by Requester" will not be available in Release 1.

1374: UC Create Business Partner Account

1966: BC Business Services

This BC was exported before.

1982: SKT Register



Restrictions

Terms and conditions will be displayed on a separate page after this registration form for approval by the user.

Certificate will not be created and provided by the Marketplace but the public key will be uploaded during the registration process.

While the sketch of the screen creates the impression only two alternative Ids are required for a business partner (Provider ID and Spot Operator ID), those are in fact just the only two known alternative Ids for the time being. The implementation should be prepared to cover more than only two IDs in the long run. In order to be aligned with the specification the labels for the two known IDs should be EVCOProviderId and EVSEOperatorId and the information field for those should cover that they consist of the country code and the actual Provider/Spot Operator Id. For all other occurences of those IDs in any screen, the same logic applies.

2407: BC Service Converter

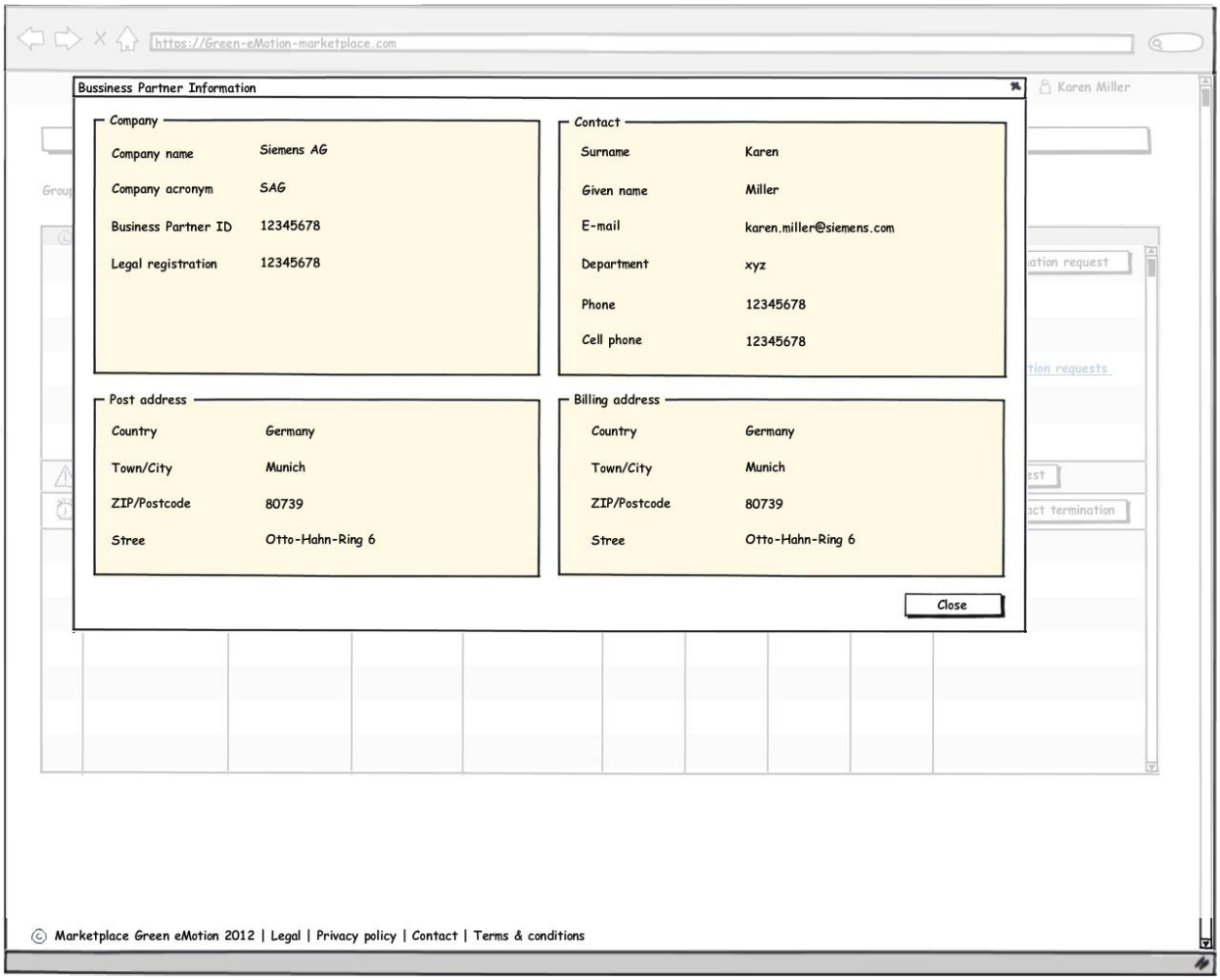
This BC was exported before.

1377: UC View Business Partner Account Details

1966: BC Business Services

This BC was exported before.

2096: SKT View Business Partner Details

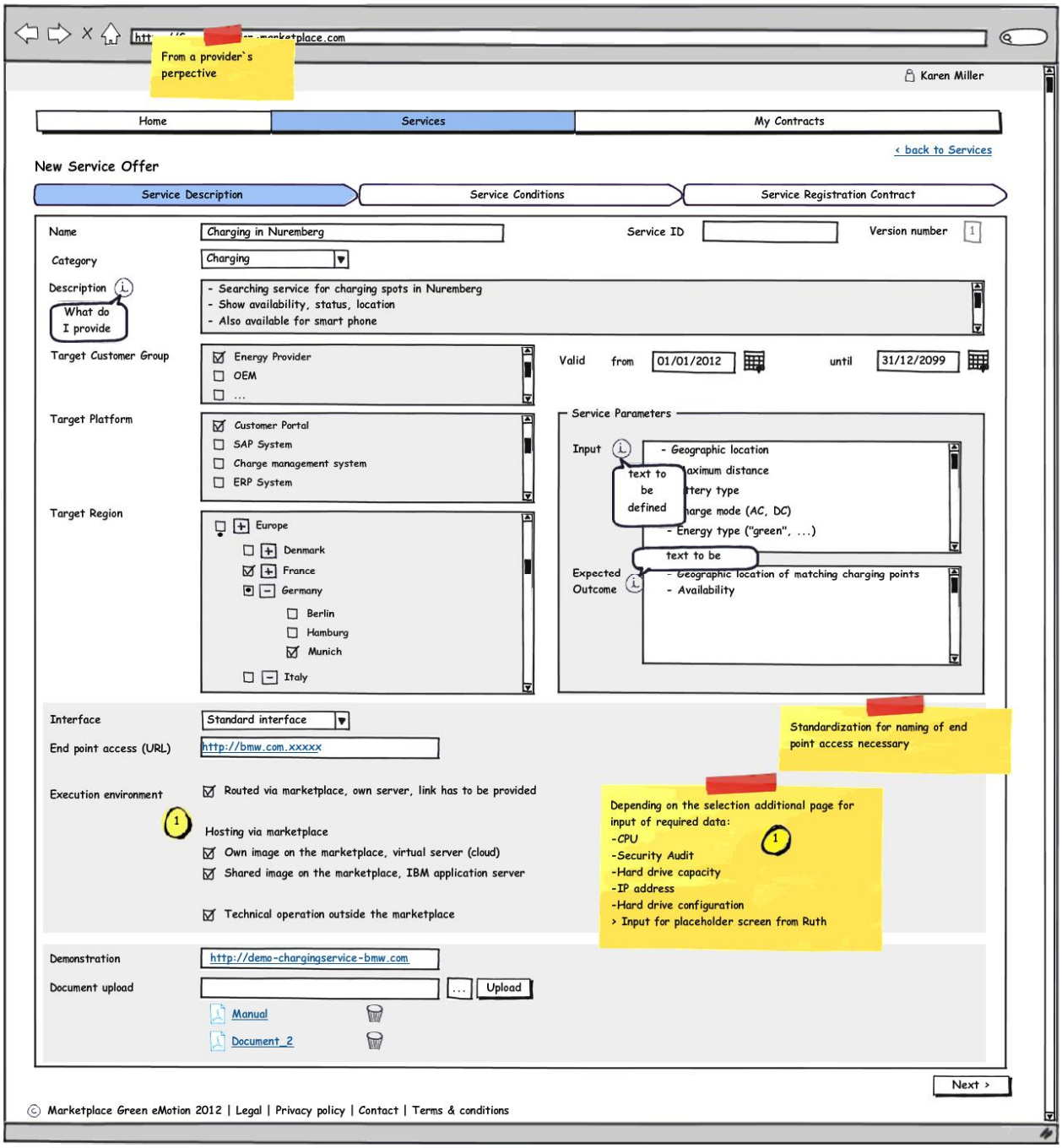


1470: UC Create Service Contract Offering

1966: BC Business Services

This BC was exported before.

1979: SKT New Service Offer



Restrictions
Execution Environment will be handled as radio group (no checkboxes).
Company information is inherently given by the logged in user.
The price is not part of this screen but is handled as part of the Service Conditions.

Dropdowns for "Target business customer", "Target platform", "Target region" will allow multiple selections.

2407: BC Service Converter

This BC was exported before.

1475: UC Search and Select Standard Interface

1966: BC Business Services

This BC was exported before.

2097: SKT Search and Select Service Type

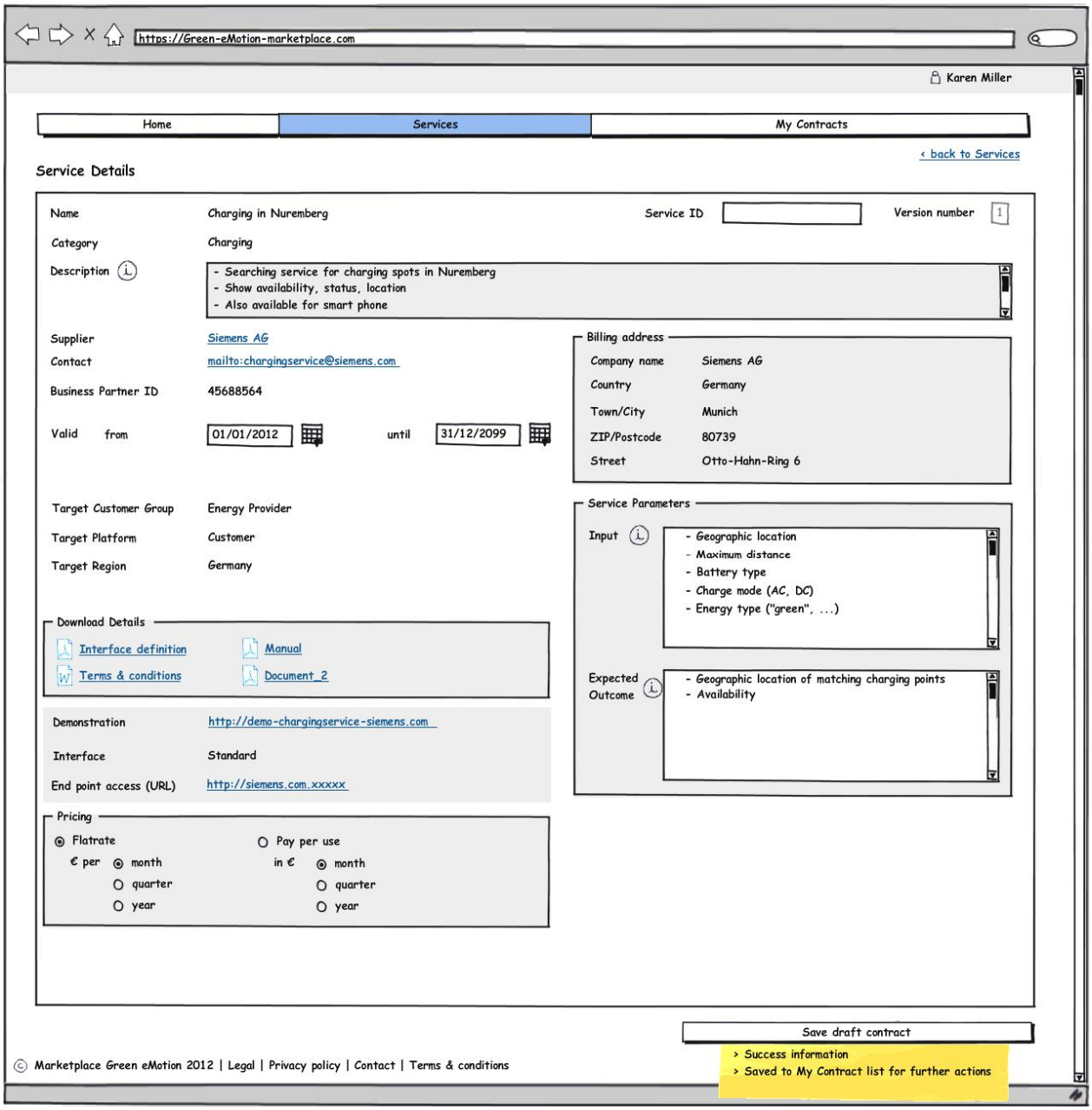
In the first release, standard interfaces represent service types and vice versa.
The user interface will be identical to "Search Service Offer".

1476: UC View Standard Interface Details

1966: BC Business Services

This BC was exported before.

1981: SKT Service Details



Restrictions

Company link is sufficient to get provider details. No additional company description will be displayed.

Terms and Conditions are placed on a separate page only and include the pricing conditions and billing address (not editable, taken from registration information).

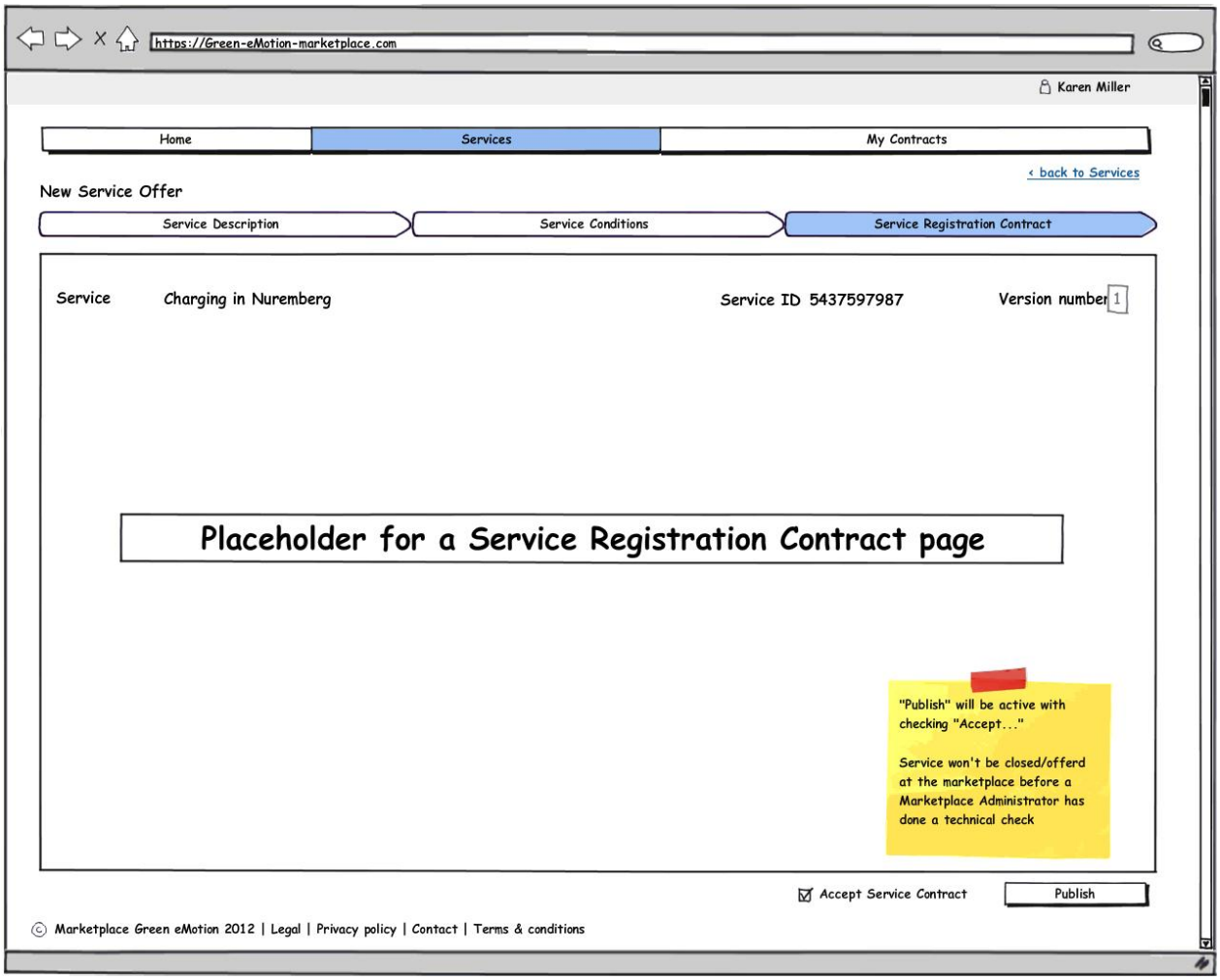
Only after determining the contract options the contract can be exported to be shown to legal (after saving it to contracts) or subscribed.(no one click buy as in B2C marketplaces).

1477: UC Create Service Registration Contract

1966: BC Business Services

This BC was exported before.

1977: SKT Registration Contract



2407: BC Service Converter

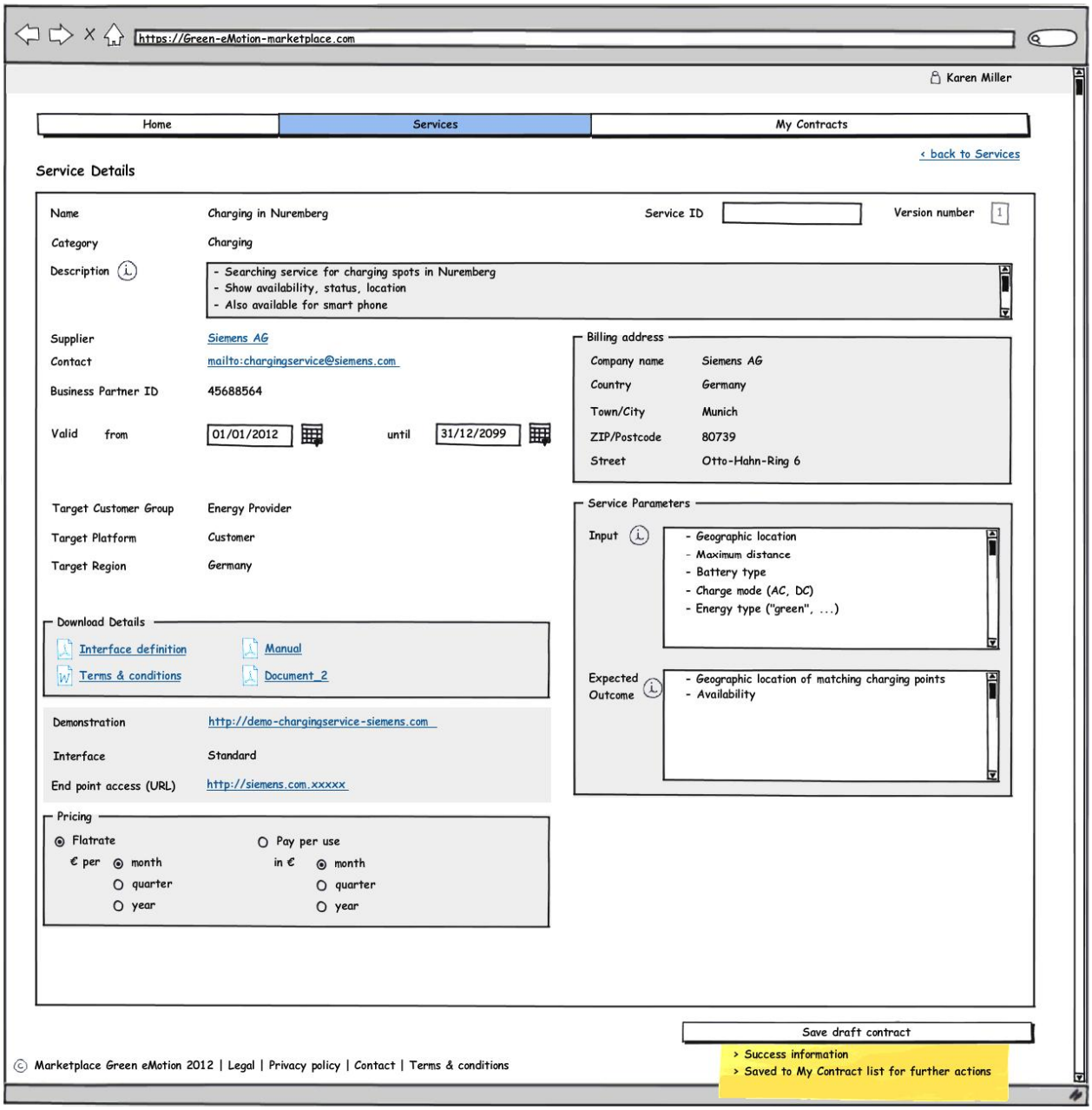
This BC was exported before.

1478: UC Download Service Content

1966: BC Business Services

This BC was exported before.

1981: SKT Service Details



Restrictions

Company link is sufficient to get provider details. No additional company description will be displayed.

Terms and Conditions are placed on a separate page only and include the pricing conditions and billing address (not editable, taken from registration information).

Only after determining the contract options the contract can be exported to be shown to legal (after saving it to contracts) or subscribed.(no one click buy as in B2C marketplaces).

1479: UC View Service Contract Details

1966: BC Business Services

This BC was exported before.

2099: SKT View Service Contract Details

The user interface shows the same details as the service details page.
Additionally, it provides the following information:

- Information on the service requester in the same way as on the service provider.
- Duration of the contract

Via a link or a button the service contract can be viewed or downloaded.
Via a link or a button a popup user interface for service contract termination requests will be displayed.

2407: BC Service Converter

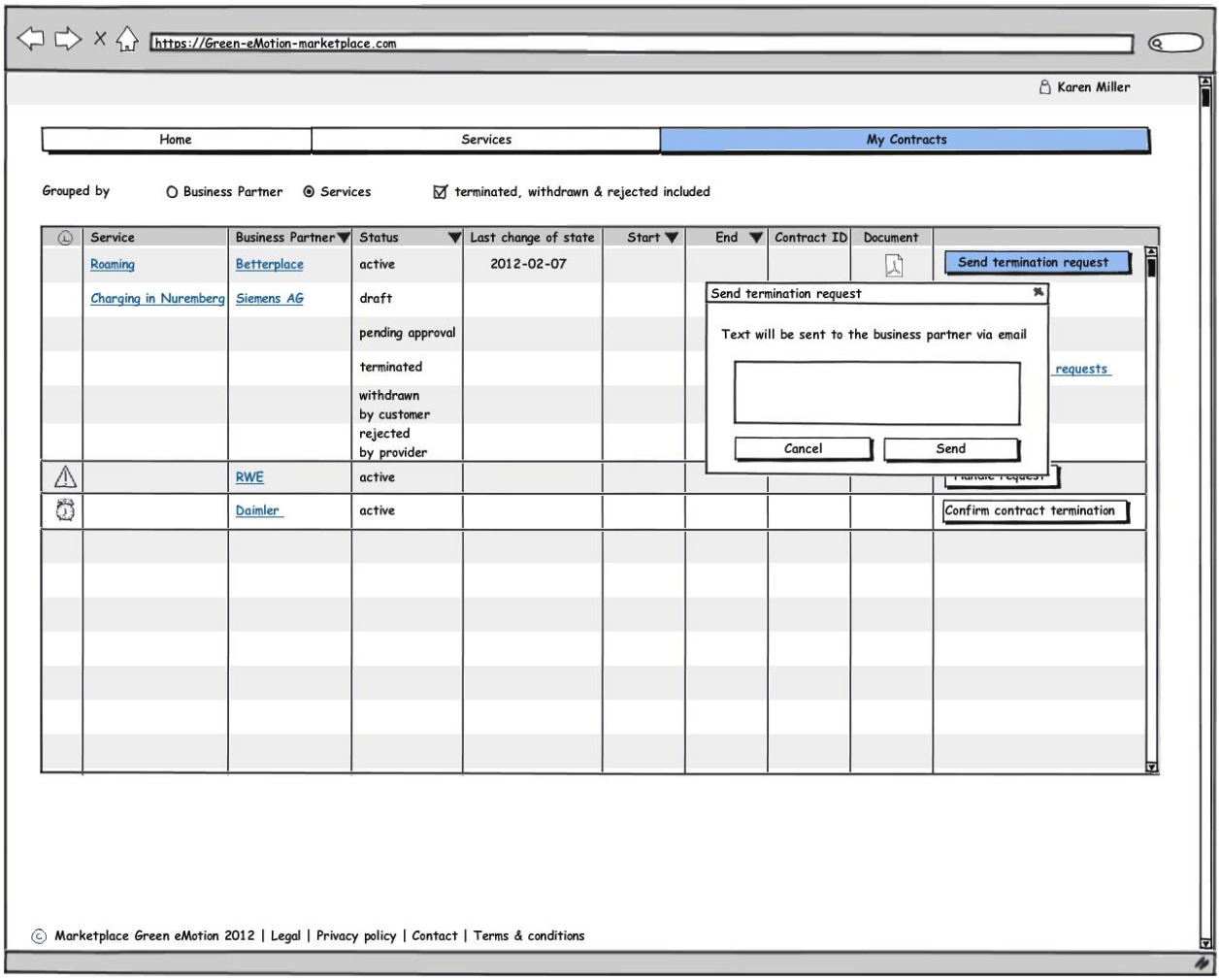
This BC was exported before.

1480: UC Create Service Contract Termination Request

1966: BC Business Services

This BC was exported before.

2100: SKT Send Contract Termination Request



2407: BC Service Converter

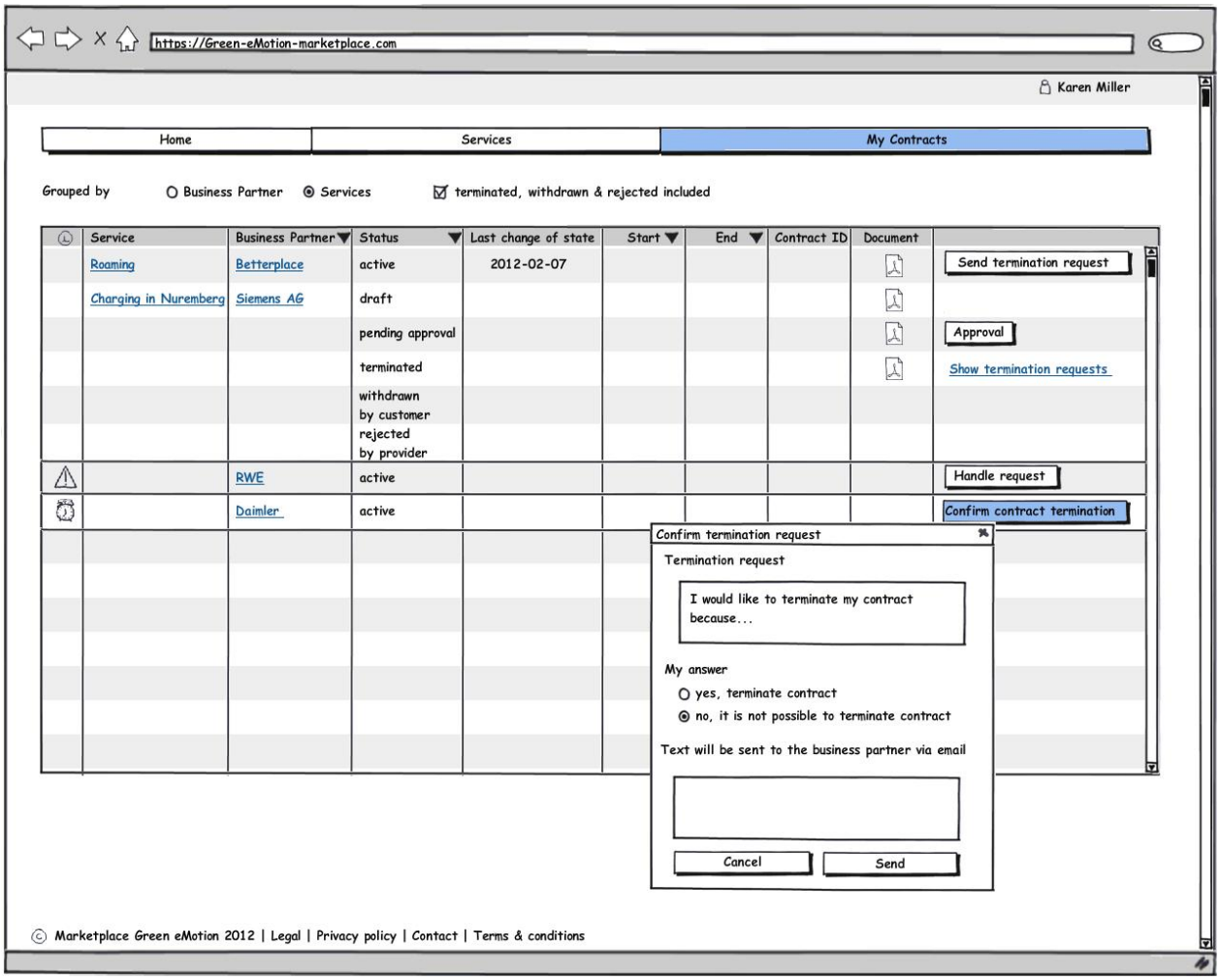
This BC was exported before.

1481: UC Confirm Service Contract Termination

1966: BC Business Services

This BC was exported before.

2152: SKT Confirm Contract Termination



2407: BC Service Converter

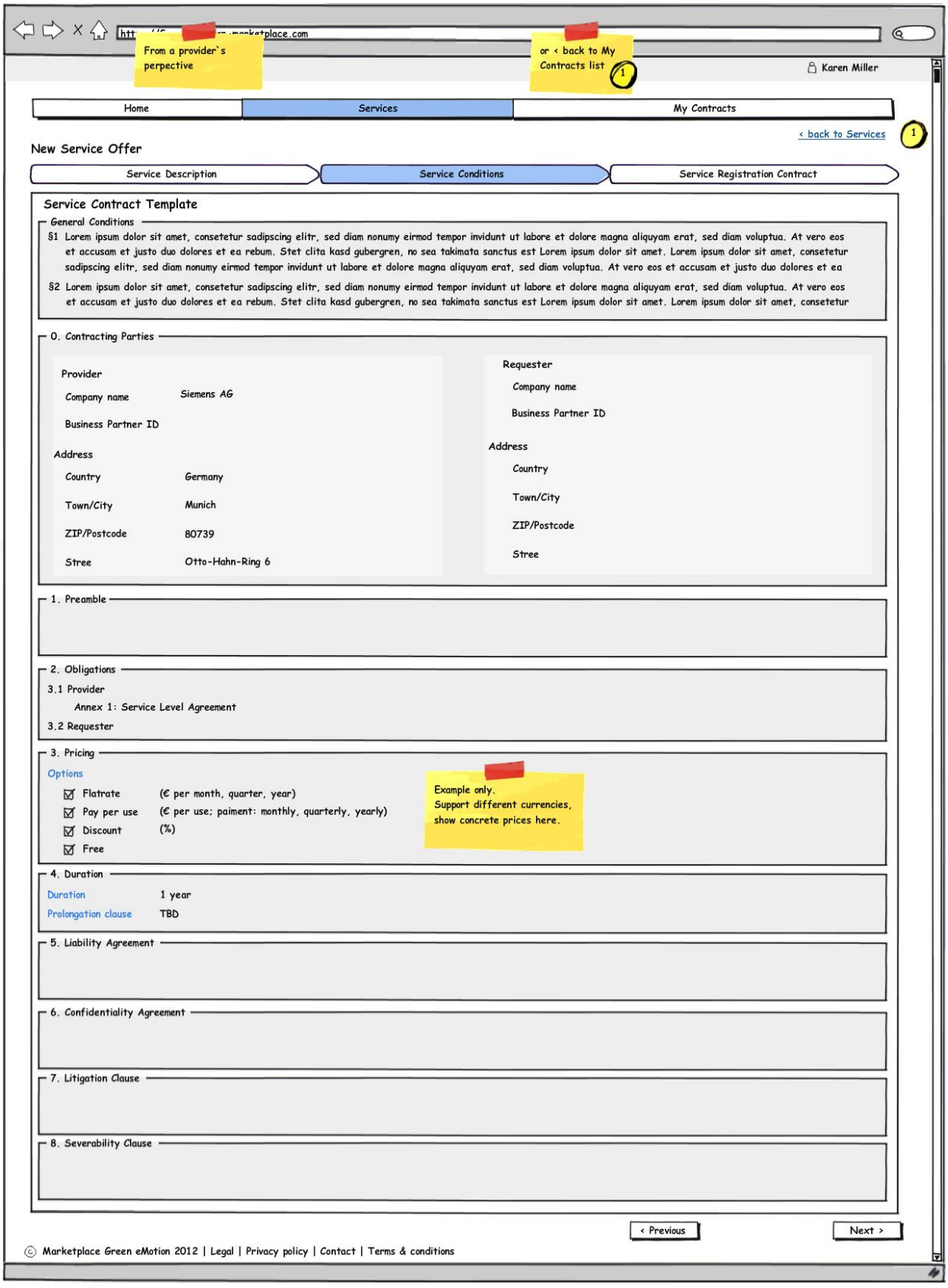
This BC was exported before.

1482: UC View Service Contract Template

1966: BC Business Services

This BC was exported before.

1978: SKT Service Conditions

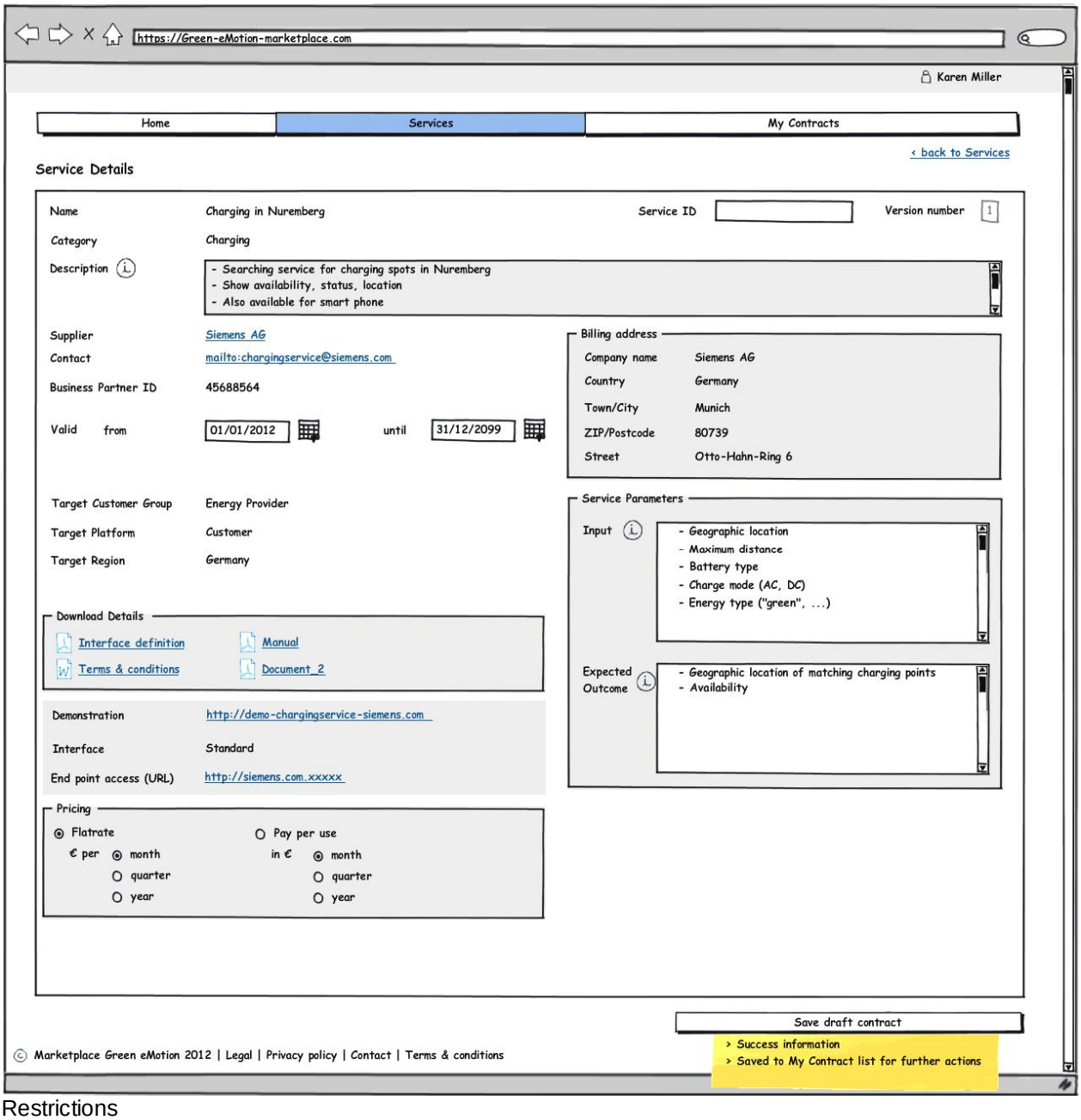

From a provider`s perpective
or < back to My Contracts list
1
Karen Miller
Home
Services
My Contracts
< back to Services
1
New Service Offer
Service Description
Service Conditions
Service Registration Contract
Service Contract Template
General Conditions
§1 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
§2 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur
0. Contracting Parties
Provider
Company name Siemens AG
Business Partner ID
Address
Country Germany
Town/City Munich
ZIP/Postcode 80739
Stree Otto-Hahn-Ring 6
Requester
Company name
Business Partner ID
Address
Country
Town/City
ZIP/Postcode
Stree
1. Preamble
2. Obligations
3.1 Provider
Annex 1: Service Level Agreement
3.2 Requester
3. Pricing
Options
Flatrate (€ per month, quarter, year)
Pay per use (€ per use; paiment: monthly, quarterly, yearly)
Discount (%)
Free
Example only.
Support different currencies, show concrete prices here.
4. Duration
Duration 1 year
Prolongation clause TBD
5. Liability Agreement
6. Confidentiality Agreement
7. Litigation Clause
8. Severability Clause
< Previous
Next >
© Marketplace Green eMotion 2012 | Legal | Privacy policy | Contact | Terms & conditions

1484: UC View Service Details

1966: BC Business Services

This BC was exported before.

1981: SKT Service Details



Restrictions

Company link is sufficient to get provider details. No additional company description will be displayed.

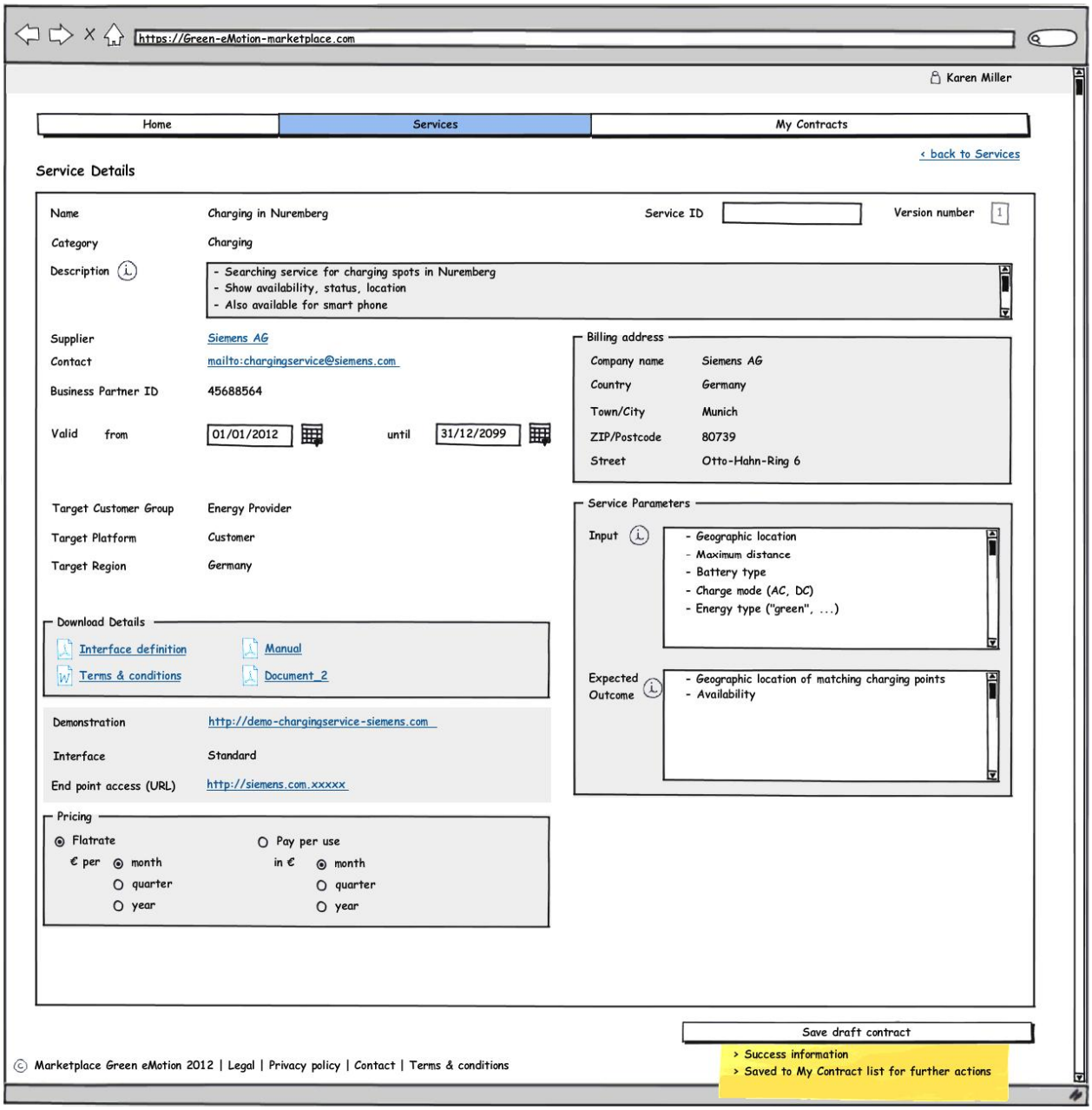
Terms and Conditions are placed on a separate page only and include the pricing conditions and billing address (not editable, taken from registration information).
Only after determining the contract options the contract can be exported to be shown to legal (after saving it to contracts) or subscribed.(no one click buy as in B2C marketplaces).

1485: UC Create Service Contract

1966: BC Business Services

This BC was exported before.

1981: SKT Service Details



Restrictions

Company link is sufficient to get provider details. No additional company description will be displayed.

Terms and Conditions are placed on a separate page only and include the pricing conditions and billing address (not editable, taken from registration information).

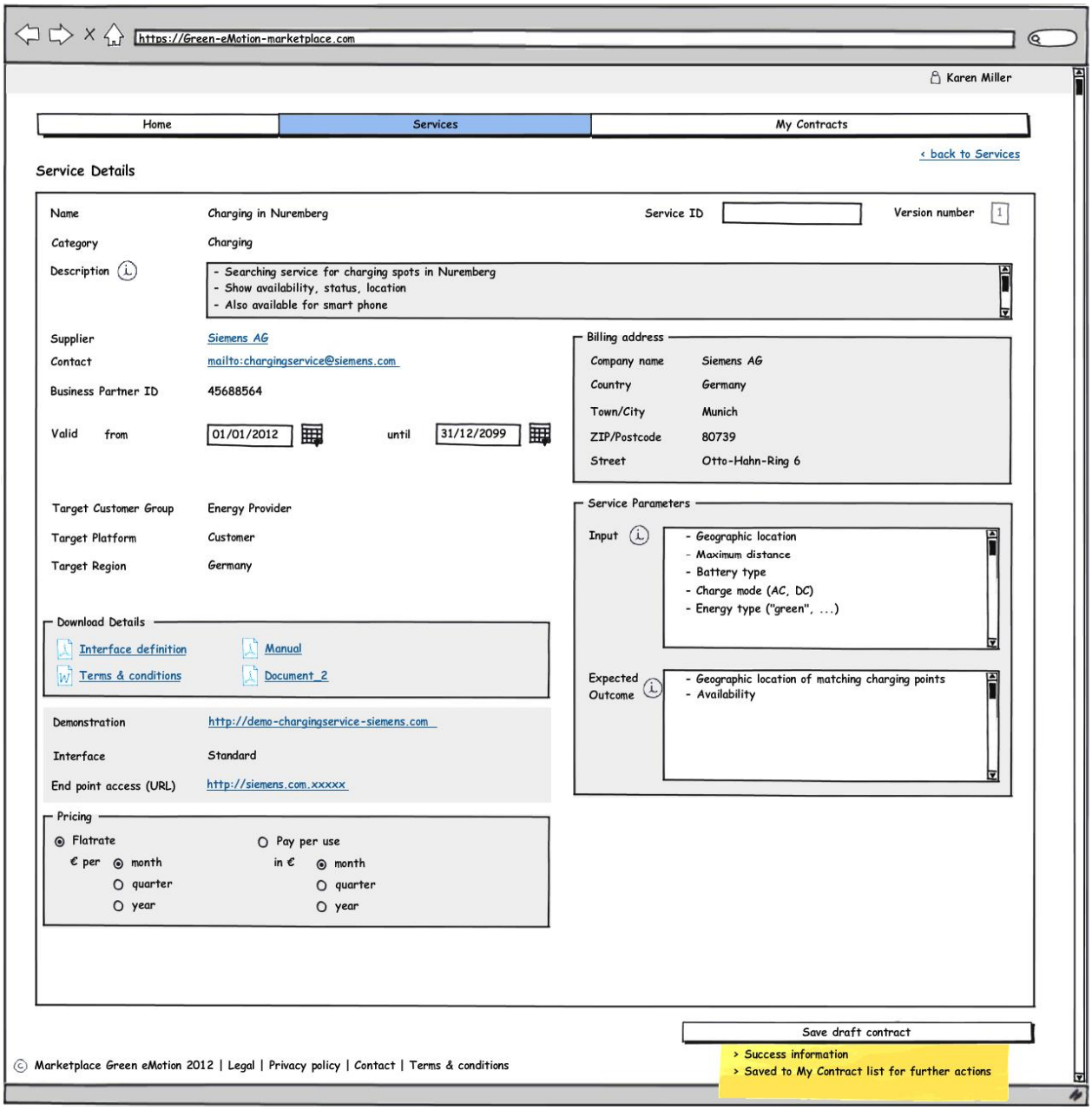
Only after determining the contract options the contract can be exported to be shown to legal (after saving it to contracts) or subscribed.(no one click buy as in B2C marketplaces).

2407: BC Service Converter
This BC was exported before.

1487: UC Download Service Specification

1966: BC Business Services
This BC was exported before.

1981: SKT Service Details



Restrictions

Company link is sufficient to get provider details. No additional company description will be displayed.

Terms and Conditions are placed on a separate page only and include the pricing conditions and billing address (not editable, taken from registration information).

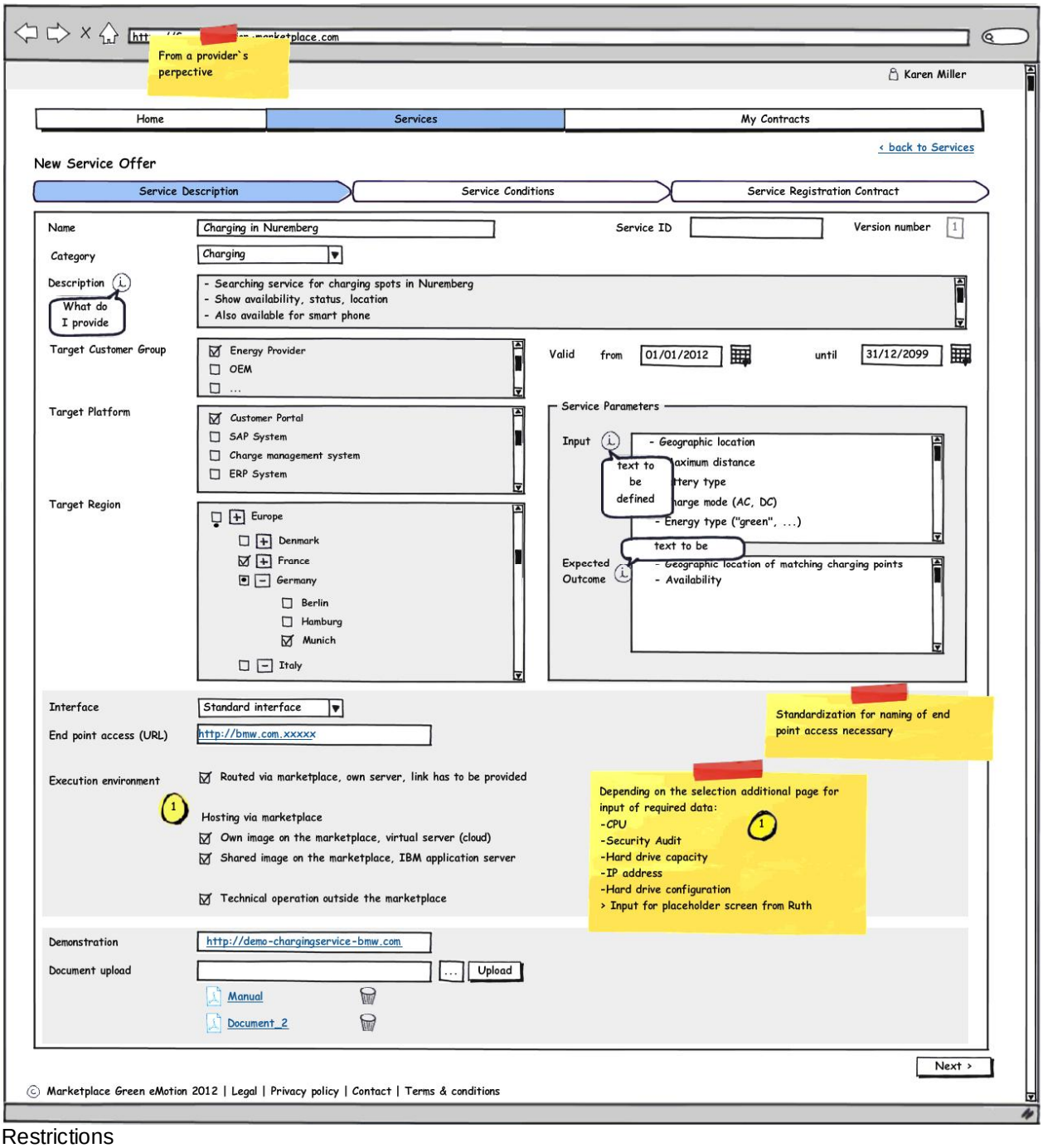
Only after determining the contract options the contract can be exported to be shown to legal (after saving it to contracts) or subscribed.(no one click buy as in B2C marketplaces).

1488: UC Upload Service Content

1966: BC Business Services

This BC was exported before.

1979: SKT New Service Offer



Restrictions
Execution Environment will be handled as radio group (no checkboxes).
Company information is inherently given by the logged in user.
The price is not part of this screen but is handled as part of the Service Conditions.

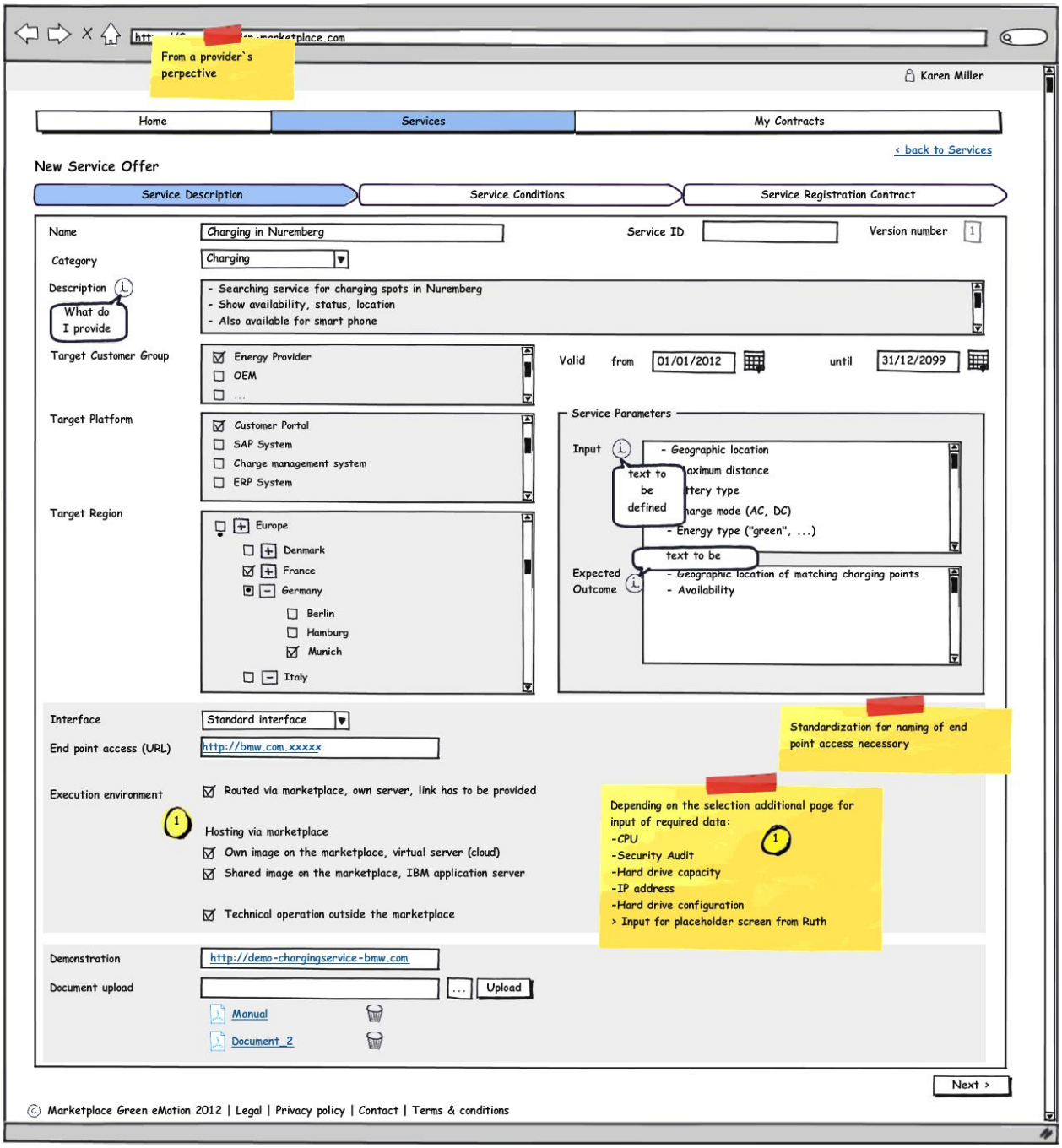
Dropdowns for "Target business customer", "Target platform", "Target region" will allow multiple selections.

1489: UC Upload Service Specification

1966: BC Business Services

This BC was exported before.

1979: SKT New Service Offer



Restrictions
Execution Environment will be handled as radio group (no checkboxes).
Company information is inherently given by the logged in user.
The price is not part of this screen but is handled as part of the Service Conditions.

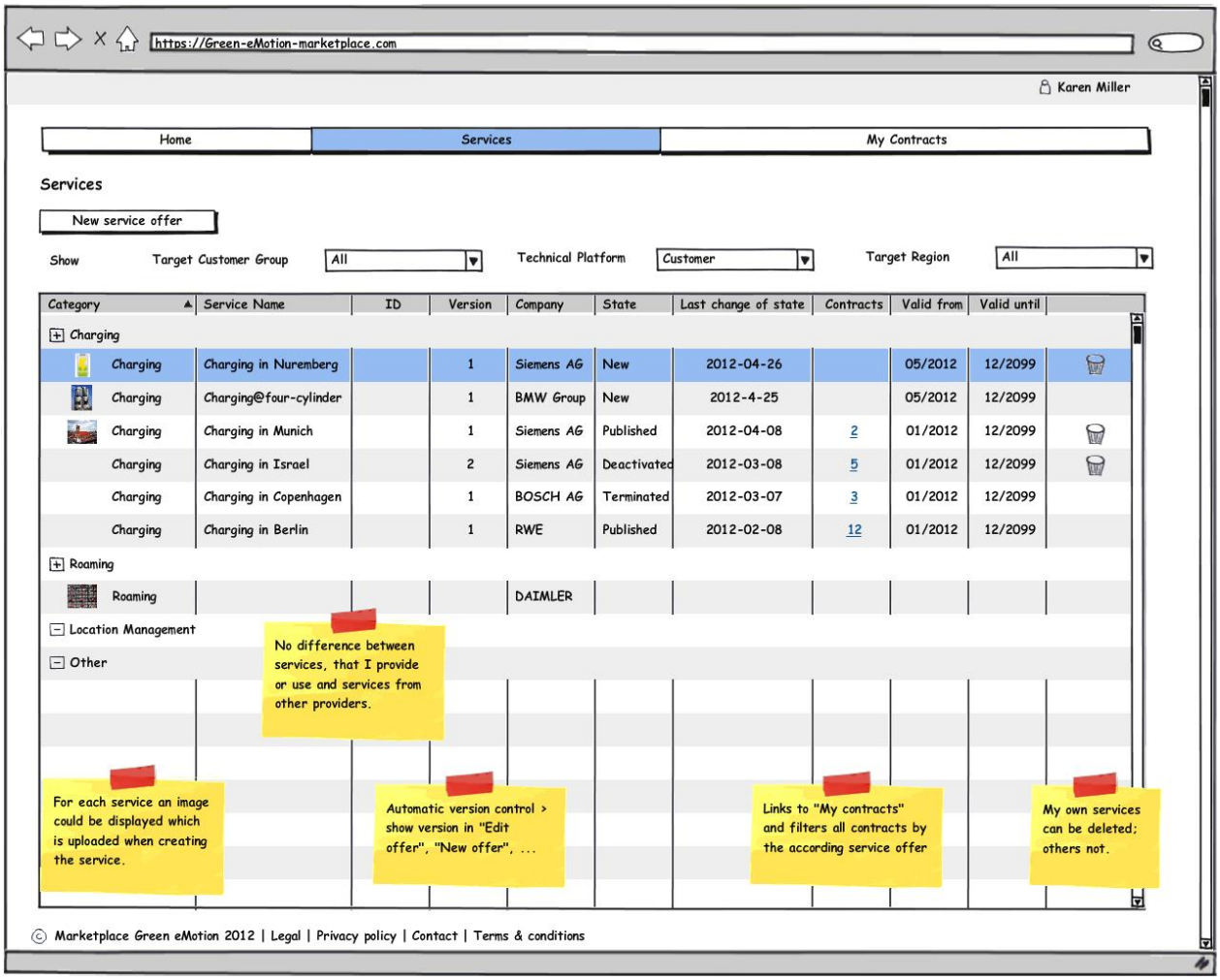
Dropdowns for "Target business customer", "Target platform", "Target region" will allow multiple selections.

1494: UC Publish Service

1966: BC Business Services

This BC was exported before.

1980: SKT Service Offers



Restrictions
There will be no version management in Release 1 for Services.
For Release 1 only the states "Submitted" and "Published" are supported.

2407: BC Service Converter

This BC was exported before.

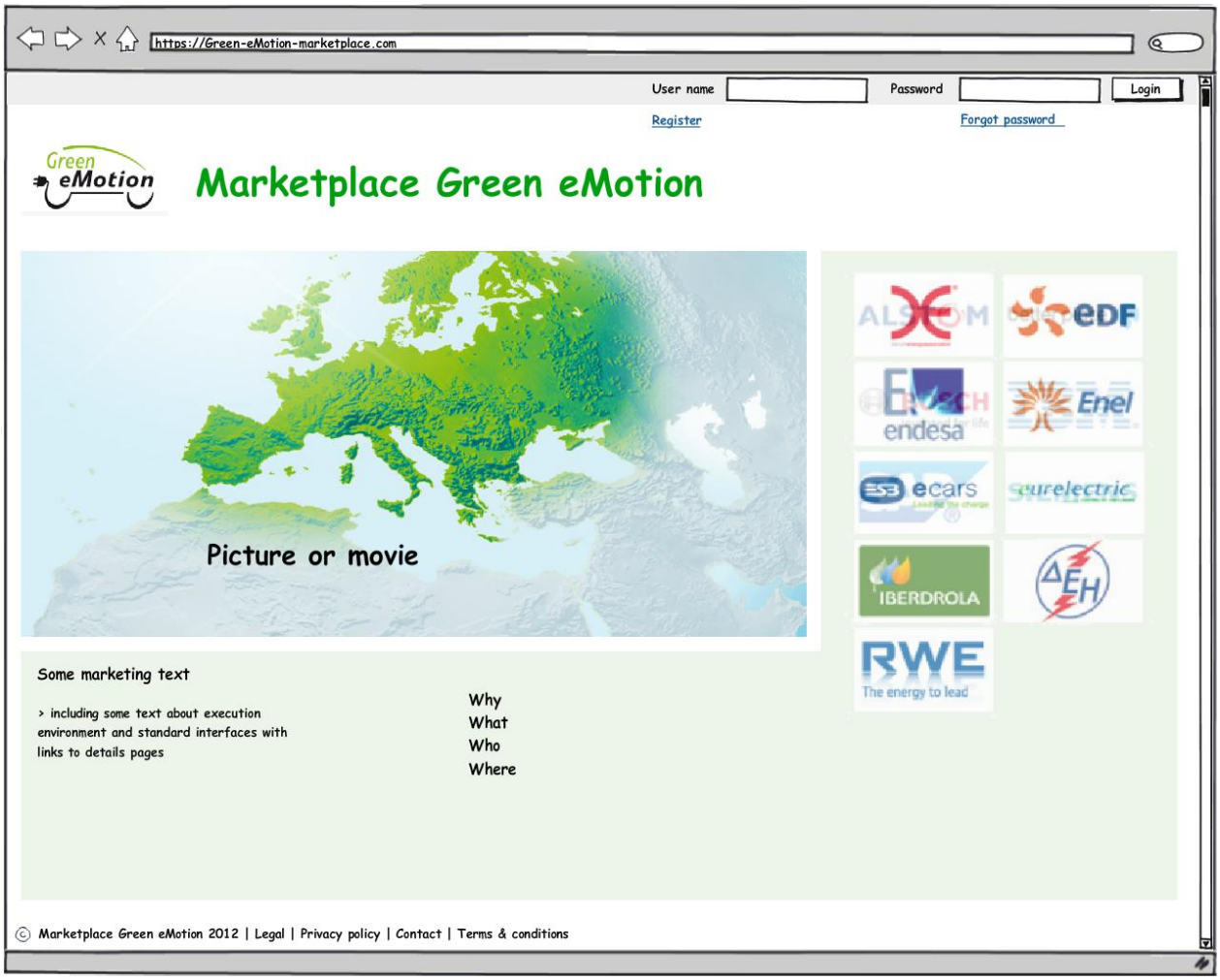
1618: UC Marketplace - Login

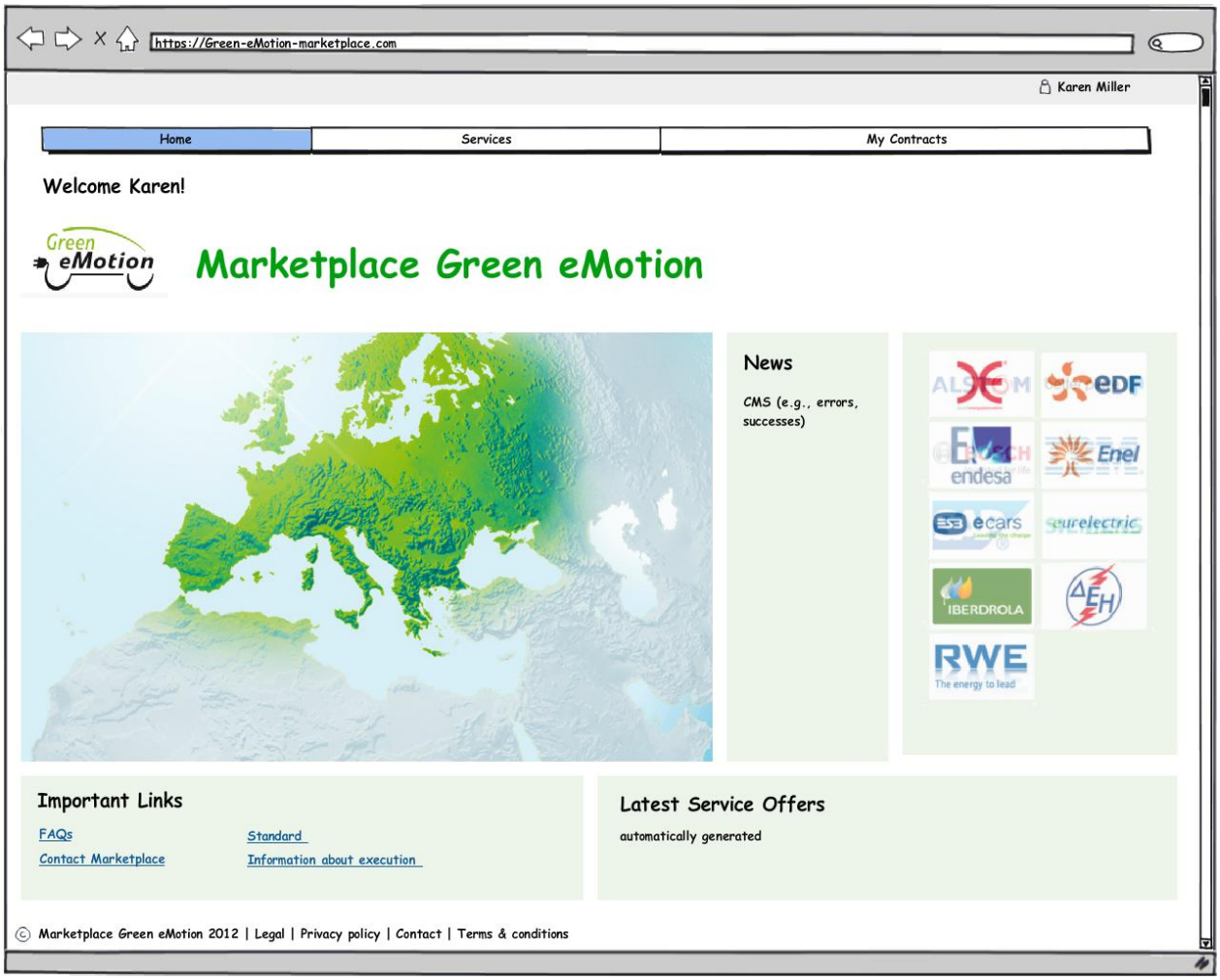
1966: BC Business Services

This BC was exported before.

2089: SKT Marketplace - Login

The login functionality will be realized by a form providing input fields for the user credentials (i.e. username and password) on the start page.





1619: UC Marketplace - Logout

1966: BC Business Services

This BC was exported before.

2088: SKT Marketplace - Logout

The logout functionality provides no own user interface, but is realized by a link or a button in the right upper corner on each individual screen.

1687: UC Notify Service Requesters of own Service

1966: BC Business Services

This BC was exported before.

2090: SKT Notify Service Requester

The functionality will be shown in a popup screen providing a form with the following elements:

- Text field: Subject
- Text area: Message
- Button: Send Message
- Button: Cancel

2407: BC Service Converter

This BC was exported before.