Funded by
the European Union

Horizon Europe
EUROPEAN COMMISSION
European Climate, Infrastructure and Environment Executive Agency (CINEA)
Grant agreement no. 101056765

ev4eu

Electric Vehicles Management for carbon neutrality in Europe

Deliverable D5.3
High-level design of Open V2X Management Platform (O-V2X-MP)

Document Details

Due date	31-04-2023
Actual delivery date	06-05-2023
Lead Contractor	Public Power Corporation (PPC)
Version	1.0
Prepared by	Alexios Lekidis (PPC)
Reviewed by	Igor Mendek (UL), Mario Nunes (INESC ID)
Dissemination Level	Public

Project Contractual Details

Project Title	Electric Vehicles Management for carbon neutrality in Europe
Project Acronym	EV4EU
Grant Agreement No.	101056765
Project Start Date	01-06-2022
Project End Date	30-11-2025
Duration	42 months

Document History

Version	Date	Contributor(s)	Description
0.1	17/02/2023	PPC	Table of contents
0.2	13/04/2023	PPC	First version
0.3	26/04/2023	UL, INESC ID	Deliverable internal review
0.4	02/05/2023	PPC	Revised version based on review comments with additional improvements on the sections and in the images
0.5	03/05/2023	UL, INESC ID	Second internal review
1.0	05/05/2023	PPC	Revised version based on review comments

Disclaimer

This document has been produced in the context of the EV4EU[1] project. Views and opinions expressed in this document are however those of the authors only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the grating authority can be held responsible for them.

Acknowledgment

This document is a deliverable of EV4EU project. EV4EU has received funding from the European Union´s Horizon Europe programme under grant agreement no. 101056765.

Funded by
the European Union

[1] https://ev4eu.eu/

Executive Summary

The *High-level design of Open V2X Management Platform (O-V2X-MP), (Deliverable D5.3)* provides an overview of the architecture and the high-level requirements that will be used for the development of Open V2X Management Platform (O-V2X-MP). O-V2X-MP will be used for implementing the Vehicle-to-Everything (V2X) scenarios within the project. The architecture is based on the functional view and services that O-V2X-MP as well as the user groups it supports. Moreover, the requirements are listed for the backend implementation as well as the frontend. Furthermore, the platform includes a list of Application Programming Interfaces (APIs), to facilitate the integration of technical, social and environmental aspects.

The backend requirements provide the main mechanisms and interactions that the O-V2X-MP platform should have to accommodate different V2X scenarios and especially to facilitate all the use-cases that will be tested within the EV4EU project. In addition, the backend also includes APIs to facilitate the interconnection and integrations with all the other layers of the platform, and with external entities that would like to contribute to its open-source module developments. Additionally, the frontend requirements provide the user interface, including widgets and dashboards that the platform should have for different user groups and categories that are also listed as a part of this deliverable. Preliminary feedback for formulating these requirements in the scope of the environmental and user layer was considered based on deliverable D3.1 "EV Users' Needs and Concerns -Preliminary Report". Moreover, the security layer is also explained in this deliverable, and the specific security mechanisms are presented in high-level (will be detailed in deliverable D5.4 "Cyber-security and Privacy analysis for V2X services").

Furthermore, the deliverable also describes the main elements that the O-V2X-MP platform should have in order to enable V2X compatibility, such as the modes that should be supported as well as the interactions and communication flows with entities in the V2X ecosystem. The communication flows are presented in the form of Entity Relationship Diagrams (ERDs) containing all the classes of involved entities in the platform as well as their corresponding relationships.

The architecture that is described in deliverable D5.3 will be used as a main reference point for the development of the O-V2X-MP platform, that will be reported on February 2024 as part of deliverable D5.5 "Open V2X Management Platform". Finally, the deliverable D5.3 has been prepared and edited by the leader of Work Package (WP) 5 – PPC.

Table of Contents

List of Figures

List of Tables

Acronyms

AC	Alternate Current
API	Application Programming Interface
BUC	Business Use Case
CCS	Combined Charging System
CDR	Charge Detail Record
CHAdeMO	CHArge de MOv
CMS	Content Management System
CPO	Charge Point Operator
CRM	Customer Relationship Management
CSMS	Charging Station Management System
DC	Direct Current
DER	Distributed Energy Resource
DSO	Distribution System Operator
EMS	Energy Management System
eMSP	eMobility Managed Service Provider
ERD	Entity Relationship Diagram
ERP	Enterprise Resource Planning
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
GUI	Graphical User Interface
JSON	JavaScript Object Notation
OCPI	Open Charge Point Interface
OCPP	Open Charge Point Protocol
OICP	Open Intercharge Protocol
O-V2X-MP	Open V2X Management Platform
PCAP	Packet Capture
PIN	Personal identification number
POI	Point of Interest
RBAC	Role-Based Access Control
RFID	Radio-Frequency Identification
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
TSO	Transmission System Operator
UC	Use case
UUID	Universal unique identifier
V2G	Vehicle-to-Grid
V2X	Vehicle-to-Everything
VPN	Virtual Private Network
VPP	Virtual Power Plant
WP	Work Package

1 Introduction

1.1 Scope and Objectives

This deliverable provides the initial architectural design of the O-V2X-MP platform. It focuses on the functional modules and user groups as well as the requirements for the backend and frontend implementation. Then, it illustrates the current platform implementation and the operational modes that should be implemented in order to ensure V2X compatibility. Concretely, the objectives of this deliverable are as follows:

1) Specification of all high-level functionalities of the O-V2X-MP platform.
2) Design of a hierarchical architecture with functional layers and corresponding services.
3) Detailed description of the interfaces for the information exchange between the platform functional layers.
4) Detailed description of the communication with all other external parties necessary for conducting V2X scenarios and the provision of the energy service.
5) Social layer interaction with the frontend interface of the platform, having descriptions of Electric Vehicle (EV) users, charging profiles and market participants.
6) Encapsulation of environmental layer specific characteristics for the different geographical locations where the platform is used, reflecting mainly natural conditions such as weather conditions.

1.2 Structure

The current document is divided into six sections. Section 1 introduces and describes the D5.3. Section 2 provides insight on the main entities that are involved in EV communications and will be interacting with the O-V2X-MP platform. Then, Section 3 provides an initial view on the architectural design of the O-V2X-MP platform. Section 4 builds on the architectural design to derive the requirements that the O-V2X-MP should have and presents the current state of the platform that is developed. Section 5 examines compatibility issues of the platform with the V2X standards and regulations. Finally, Section 6 presents overall conclusions and considerations about this deliverable.

1.3 Relationship with other deliverables

Deliverable D5.3 describes the design of the O-V2X-MP platform and provides requirements for compliancy to the V2X and specifically the Vehicle-to-Grid (V2G) connectivity and communication standards. Hence, it includes the V2G communication architecture and the standardization activities as input from the deliverable D5.2 “Standardization gap analysis for new V2X related Business Models”. Additionally, it receives input for the environmental and user layer from deliverable D3.1 “EV Users’ Needs and Concerns -Preliminary Report” and it provides an initial view of the security mechanisms of the O-V2X-MP platform that will be reported in deliverable D5.4 “Cyber-security and Privacy analysis for V2X services”. Moreover, the design of the O-V2X-MP platform and the relevant requirements are used for the development of its functionalities which will be reported in D5.5 “Open V2X Management Platform”. Finally, the architecture and V2X compatibility of the O-V2X-MP platform will be tested in relation to the Business Use Cases (BUCs) of the project that are defined in deliverable D1.5 “V2X Use-cases repository” and the respective information exchange in the BUCs presented in deliverable D5.1 “Information Exchange needs to enable different UCs” [1].

2 Background

In the scope of the EV4EU project, a V2X platform, called O-V2X-MP, is being developed. The platform will include a frontend and a backend. The backend should implement Open Charge Point Protocol (OCPP) version 1.5, 1.6 and 2.0 further detailed in [2] to ensure the support of both charging and discharging scenarios and to provide backwards compatibility for EV chargers that do not support OCPP 2.0 yet. The current version of OCPP 2.0 [3] is being developed in 2.0.1, which also serve as the basis for the initial design of O-V2X-MP, described in this deliverable. Moreover, the frontend should be a high-level interface for operators and users to view their charging sessions and provide feedback to the platform administrator, hereby called *Platform Manager*.

To implement V2X scenarios, which allow the interactions between the vehicles, the grid, the environment and the EV users, the platform should first start with the development of the V2G communication mechanisms and scenarios. V2G is a technology that allows EVs to discharge energy to the grid, acting as a Distributed Energy Resource (DER) and providing services to the grid. This can include supplying power to the grid during periods of high demand and absorbing excess power during periods of low demand.

The main entities that are involved in EV communication architectures are: 1) the vehicle, 2) charging station and 3) the Charging Station Management System (CSMS). The CSMS is responsible for the remote control, monitoring and maintenance of charging station as well as the resolution of faults or issues in them. Furthermore, it can also obtain remote diagnostics from the charging stations regarding their health status, real-time availability and the audit logs.

The EV charging architecture entities along with their communication flows and the corresponding standards are illustrated in Figure 1. More specifically, Figure 1 presents the communication flows and the corresponding standards between 1) EVs and charging stations, 2) CSMS systems and 3) Different EVs in the V2X ecosystem.

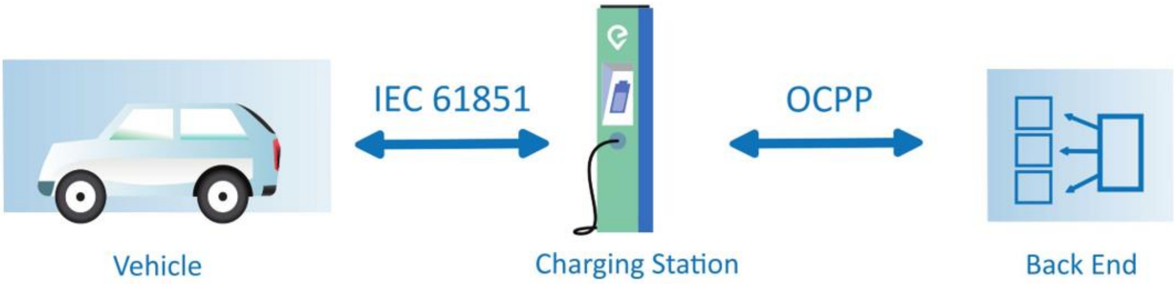

Figure 1 - EV charging architecture entities (detailed in [2])

Furthermore, Figure 1 is a simple example allowing the identification of the main standards used for the communication between the EVs and the charging station and between the charging station and the backend of a cloud system. However, this backend can be operated by different stakeholders such as Charge Point Operators (CPOs), flexibility operators, energy communities, Virtual Power Plants (VPPs), etc. This means that this architecture can be much more complex than the one presented in the figure [9].

As depicted by Figure 1 each entity invokes a different communication channel using standardized interfaces for data exchange. Specifically, the main standards that are currently used in EV charging infrastructures are the IEC 61851 [10] and OCPP [7]. These standards are further described in deliverable D5.2 “Standardization gap analysis for new V2X related Business Models” [2].

Within EV4EU a new and open CSMS platform is developed to comply with both charging and discharging scenarios in V2G. This platform is called O-V2X-MP and its architecture is illustrated in Section 3.

3 Architecture

The architecture of the O-V2X-MP platform is based on OCPP 1.6 as well as OCPP 2.0 that are defined in deliverable D5.2 "Standardization gap analysis for new V2X related Business Models" [2]. Since, OCPP 2.0 is a working standard document being currently readjusted within the V2X Working Group, we based our analysis and design of the architecture on generic modules and functionalities that are defined in version OCPP 2.0.1 [22]. This version consists of different parts, including:

- Part 0: Introduction
- Part 1: Architecture & Topology
- Part 2 Specification: Use Cases and Requirements, Messages, Data Types and Referenced Components and Variables
- Part 3: Implementation Schemas
- Part 4: Implementation Guide using JavaScript Object Notation (JSON) requests and responses in OCPP commands

These parts were taken into consideration, but since no actual V2G tests have been performed yet, to the best of our knowledge, in this deliverable we focus on the basic set of services to enable interactions in the V2X ecosystem. Thus, the most advanced functionalities that will be implemented in O-V2X-MP will be presented in the report D5.5 "Open V2X Management Platform" that will be delivered along with the software code platform, in February 2024.

Moreover, the architecture integrates the consumers' needs, pains, and potential gains since the design enables the participation of V2X flexibilities services managed by CPOs and VPPs, and provide interoperability in communications and data exchange between all stakeholders (EV owners, CPOs, VPP, Transmission System Operator (TSO), Distribution System Operator (DSO)) engaged in the development of next-generation energy services. Additionally, it facilitates the communications between vehicles and between the vehicles and the infrastructures. Finally, the architecture ensures scalability, to facilitate many consumers and services.

All these functionalities are combined in the different layers of the O-V2X-MP platform, which are presented in Figure 2. Specifically, the platform includes a backend and a frontend interface, which are detailed in Section 4 of this deliverable. These interfaces provide its core technical functionalities. Moreover, the platform includes a user layer, which aggregates descriptions and models of EV users, charging profiles, and market players. For instance, the social layer can include user segmentation into different categories, behavioural modelling, evaluation of users' interactions, and their social links. Additionally, it also includes an environmental layer, which reflects the impact of weather conditions and geographical location conditions to the scenarios that are studied through the platform. Finally, there is a cyber-security layer which protects the platform as well as the EV users against both cyber and physical threats and attacks using cyber-security and data privacy mechanisms.

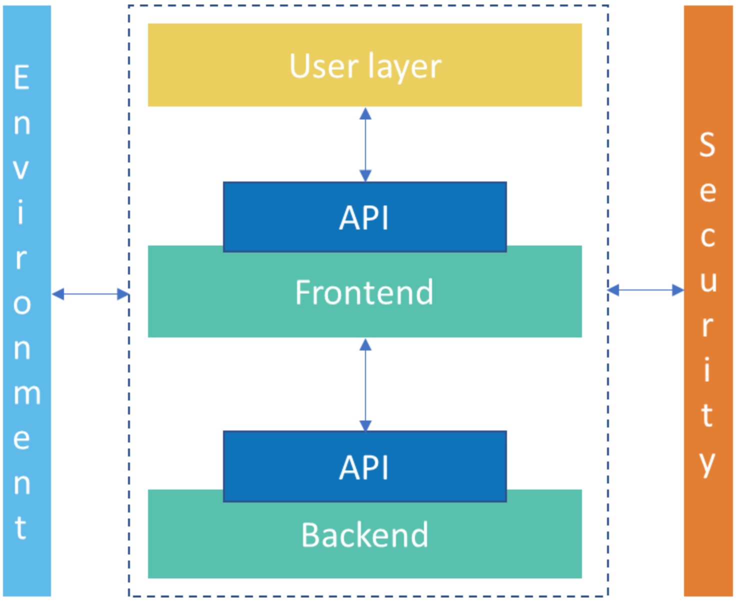

Figure 2 - High-level view of the O-V2X-MP platform

3.1 Functional services and modules

The main entities that are involved in the architecture are the 1) driver of an EV, 2) the CPO who owns and manages a charging station and 3) the emobility Managed Service Provider (eMSP) who holds the relationship with the Driver and provides an Radio-Frequency Identification (RFID) card. CPO is an entity operating a pool of charging points. CPO's responsibilities lie in ensuring that the EV charging network is operational, available and stable, round the clock. This includes:

1) Regular remote and on-site maintenance
2) Running diagnostics
3) Price settings
4) Point of Interest (POI) data management

eMSP are entities offering an EV charging service to EV drivers. Specifically, eMSPs create value for its customers (EV drivers) by providing access to multiple charging points around a geographic area. Additionally eMSPs offers Customer Relationship Management (CRM) / Enterprise Resource Planning (ERP) integration - Customer and invoicing integrations. eMSPs may use either the Open Charge Point Interface (OCPI) [14] or the Open Intercharge Protocol (OICP) [15] protocol to connect to CPO networks. Specifically, OICP is based on Simple Object Access Protocol (SOAP) and uses an object-based approach. OCPI is based on JSON and supports authorization, charge point information exchange (including live status updates and transaction events), charge detail record exchange, remote charge point commands and the exchange of smart-charging related information between parties. Both protocols support connections between eMSPs who have EV drivers as customers, and CPOs who manage charge stations. This protocol is free to use and independent. It can work both bilateral as well as in combination with roaming hubs. Everyone can participate in the development of the OCPI.

The design of the O-V2X-MP is flexible and scalable to accommodate both Platform Manager, CPO and eMSP functionalities that are depicted in Figure 3.

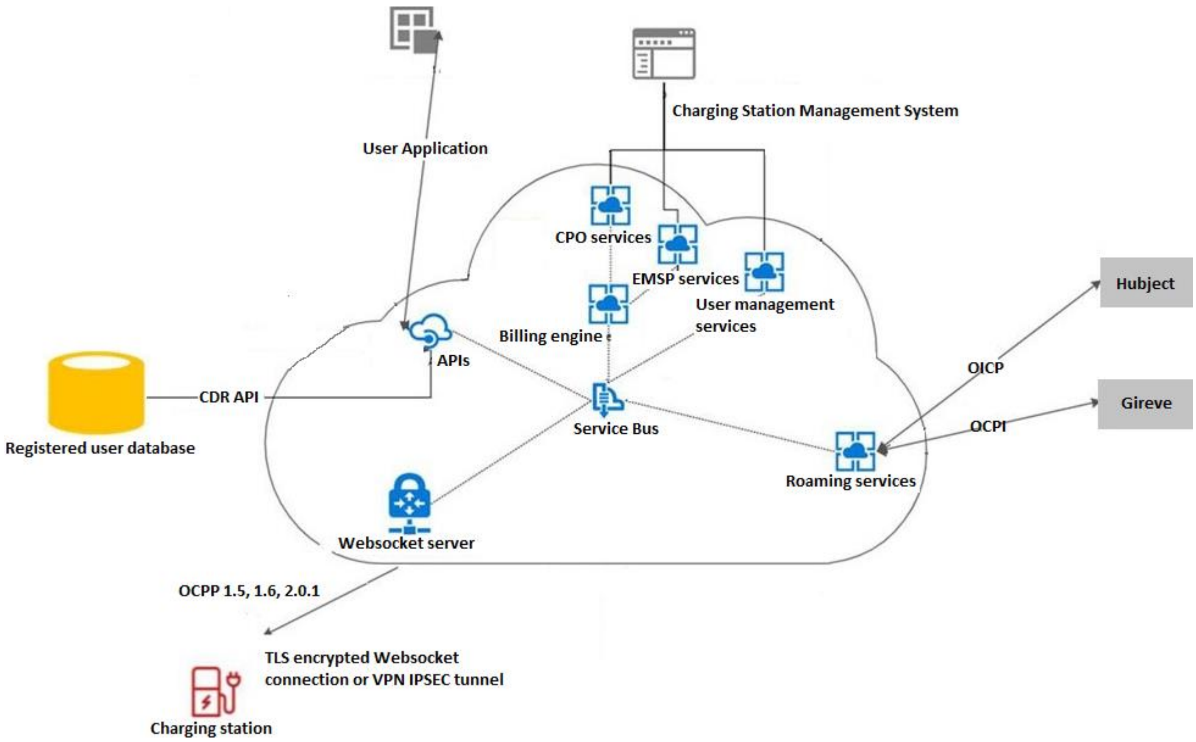

Figure 3 - Service-based architecture of the O-V2X-MP platform

Figure 3 includes the services offered by the O-V2X-MP platform, specifically:

1) *CPO services* related to the charging station management including diagnostics and operations to ensure their real-time availability.
2) *eMSP services*, related to the provision of access to charging points as well as CRM, ERP and invoicing integrations for a specific area.
3) *User Management* related to the user groups that are described in Section 3.2.
4) *API services* related to the APIs that the O-V2X-MP platform has which are described in Section 4.1.2.
5) *Secure WebSocket layer* for the communication with the EV chargers using OCPP, which ensures the protection against cyber-attacks using Transport Layer Security (TLS) WebSocket connection [22] or a Virtual Private Network (VPN) IPSEC tunnel [22].
6) *Roaming services*, which offer transaction possibilities with Roaming partners throughout the Europe, so that EV drivers have the possibility to charge in the network of partners, but also the drivers of other eMSPs to charge into the charging network. The integration with these services is based on the OICP or OCPI protocol, which allows the connection with roaming hubs as Hubject [12] and GIREVE [13] respectively.

3.2 User groups

The platform shall include access control policies, to provide different access and views for each user. These policies will be implemented through dedicated Role-Based Access Control (RBAC) policies for each individual user role. RBAC relies on the role to simplify the specification and management of access rights within an organization [6]. A role comprises a set of permissions needed to carry out a certain job function. Users are assigned to roles and inherit the permissions assigned to their roles. Roles are often organized in a role hierarchy, which defines the inheritance of permissions between roles. The list of roles and corresponding users that have access to the platform is:

- *Platform manager*, who is responsible for the entire platform including its operation
- *CPO manager*, who is responsible for the operation of the charging stations
- *Billing rule manager*, who is responsible for the definition of the billing rules
- *eMSP manager*, who is responsible for the services that are provided for the customers
- *1st line support engineer*, who provides services for the users in order to identify charging or eMSP service issues
- *2nd line support engineer*, who provides advanced support for the users and resolves charging or eMSP service issues
- *Charge Point Vendor*, who is the manufacturer of the charge points
- *Charge Detail Record (CDR) Manager*, who is responsible for customer billing and hence has access to all the customer EV charging sessions
- *CMS Manager*, who manages the frontend interface of the platform as well as the interactions with the users

4 O-V2X-MP Requirements

The architectural design and functional modules that are described in high-level in Section 3 are translated into the requirements of the following sections for the backend and the frontend. Then, the current platform implementation is also presented as a part of this section.

4.1 Backend

The backend of O-V2X-MP shall include libraries to interpret the OCPP 1.5, 1.6 and 2.0.1 messages, in order to allow the communication, remote control and maintenance of charging stations from the O-V2X-MP platform, which implements all the features and services of a CSMS. Specifically, the backend requirements include:

1) The backend should implement "Device Management / Monitoring" functionalities of OCPP 1.5, 1.6 and 2.0.1 in order to manage and monitor charging station configuration parameters.
2) The backend should implement the basic OCPP 1.5 and 1.6 commands [7], from which the most important include:
    - *Boot Notification:* This command is initiated by the charging station when it starts up and connects to the CSMS. The station sends a message to the central system to announce its presence and provide some basic information about its capabilities.
    - *Authorize:* When a customer plugs in their vehicle, the charging station sends an Authorize message to the central system to check if the customer is authorized to use the station. The central system responds with a message indicating whether the customer is authorized.
    - *Start Transaction:* If the customer is authorized, the charging station sends a Start Transaction message to the central system to begin the charging session. The message includes information about the charging station, the customer, and the charging parameters.
    - *Stop Transaction:* When the charging session is complete, the charging station sends a Stop Transaction message to the CSMS to end the session. The message includes information about the station, the customer, and the charging parameters.
    - *Heartbeat*: The charging station sends periodic Heartbeat messages to the CSMS to confirm that it is still connected and operational.
    - *Firmware Update:* If the charging station requires a firmware update, the CSMS sends a Firmware Update message to the charging station to initiate the update process.
3) The backend should implement the following OCPP 2.0 commands:
    - *ReserveNow*: This command allows an EV to reserve a specific charging point for a specific period of time. This is useful in V2G applications where the EV needs to be charged at a specific *time in order to provide grid services.*
    - *CancelReservation*: This command allows an EV to cancel a previously made reservation.
    - *GetCompositeSchedule*: This command allows the EV to request a schedule of the charging station's availability for a specific period of time, including information about its current state, planned maintenance, and any reservations that have been made.
    - *SetChargingProfile*: This command allows the EV to set a charging profile for a specific period of time. This can be useful in V2G applications where the EV needs to charge at a specific rate in order to provide grid services.

Additionally, OCPP 2.0.1 defines further authorization methods other than the ones listed in OCPP 1.6. Specifically, the backend should implement the basic OCPP 1.6 authorization option (i.e. Remote Start) for charging as well as additional OCPP 2.0.1 authorization options possible:

Radio-Frequency Identification (RFID) based on ISO 14443-3 [4], ISO 15118 Plug and Charge [19], Personal identification number (PIN) Code, Simple start button, Credit / Debit Card, Server Generated ID. Table 1 further describes the authorization methods that are supported in both OCPP 1.6 and OCPP 2.0.1.

Table 1 - Authorization methods in OCPP 1.6 and OCPP 2.0.1

OCPP 1.6	OCPP 2.0.1
RFID only.	RFID (ISO 14443-3), ISO 15118 Plug and Charge, PIN Code, Simple start button, Credit / Debit Card, Server Generated ID.
No identification of the EV possible. No mapping between EV ID and EV driver ID.	Identification of EV possible through EVCC ID (Unique identifier of Electric Vehicle Communication Controller). Hence, mapping between EV and EV driver becomes possible.

Along with the authorization methods OCPP 2.0.1 introduces additional authorization options for them, which do not exist in OCPP 1.6. These options are further described in Table 2:

Table 2 - Additional OCPP 2.0.1 authorization options for the introduced methods

New OCPP 2.0.1 functionalities	Description
Master Pass	E.g., Emergency personnel like the fire department, etc. can stop your transaction using the Master Pass.
Preferred Language	The EVCS that you are charging at, recognizes your preferred language and changes to that language automatically.
Personal Message	E.g.,: Personal Greeting, “Not authorized person”, Personal tariff information
Charging Priority	Gives priority to specific EV drivers.
Additional authorization information possible	You can add additional information to the authorization request. E.g., you can add to the authorization request additional parameters that are not part of the OCPP 2.0 directly.

4) The backend should include the capability to map an EV to a specific driver.
5) The backend should have integration support with the ISO 15118 protocol [19] to provide support for improved Smart Charging and Plug and Charge. ISO 15118 gives extra capabilities for:
    - *Plug & Charge: Charging without using RFID*
    - *Certificate Management (installing certificates on the charging station and the EV)*
    - *Smart Charging*
    - *Metering*
6) The backend should be scalable to support large CPO networks and include improved transaction handling.

4.1.1 Billing engine

The billing engine shall include all the mechanisms to perform the calculation for the cost of the charging session based on tariffs. Tariffs are modules which define charging-condition (e.g. AC 11 kW) for a charging process. A tariff generally defines at what time, at which charging station, to which customers, etc. a cost rate should be applied. This allows you to specify that a customer who is assigned to this tariff, for example, use the AC cost rate for AC charging connections, but the DC charging rate for DC charging connections. This structure of tariff management enables a dynamic cost structure. The platform should be able to create cost rates for the following categories:

a) One-time charging session fee
b) Consumption-based fee for the charging session (price per kWh)
c) Time-based fee for the charging session
d) Combinations of fees listed above

The CDR is the description of a concluded charging session and the only billing-relevant object. A CDR is the cumulative Session object combined with the relevant tariff information. Following a successful charging session, the CPO will send a CDR to the eMSP where the driver is registered with. The eMSP collects payment from the Driver and pays the CPO the difference.

Figure **4** illustrates how the CDR is derived. Specifically, during an active session, MeterValues are collected using the OCPP 1.6 respective command.

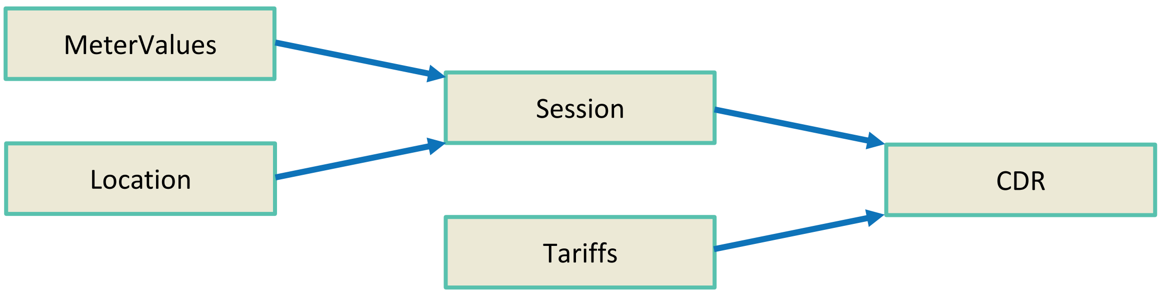

Figure 4 - Charging session billing through the CDR

The requirements that are relevant for the billing engine are:

1) A different tariff should be available for AC chargers and for DC chargers, as they have a faster charging rate.
2) The CDRs should be exportable in a user-friendly format (e.g., Microsoft Excel sheet).
3) Tariffs should be flexible and adapted based on the CPO and Platform Manager needs within the project.

4.1.2 APIs

4.1.2.1 Location API

Initially, the platform shall implement a location API. Through this API all the locations where charging stations exist may be retrieved. For EV4EU these are the São Miguel island (Azores) in Portugal, Krško and area of the DSO Elektro Celje for the Slovenian pilot, the Mesogeia Area (Attica) for the Greek pilot, and Risø and Campus Bornholm in Rønne (Bornholm Island) for the Danish pilot.

The Location API allows to retrieve the available charging stations through a map. Acording to their status, they can be devided in:

1) Public locations are those that are publicly available.
2) Private locations are not publicly available but only to specific group of users that have RFIDs to authorize to them.
3) Semi-public locations can also be indicated as the locations with parking restrictions (e.g. parking lot that closes during the night).

The map for the locations is using information through this API, in order to present them in the user frontend, described in Section 4.2.

4.1.2.2 CDR API

The platform shall include a CDR API, from where the complete session details will be exported in different formats as CSV or JSON. The session details include the charging station ID, the Location ID, starting and stopping time, total energy consumption (in kWh), the total duration and cost of the session as well as the RFID tag that was used with the user details (only if the user is previously registered in the platform).

The API allows to execute GET, POST, PUT and DELETE commands. An example JSON output for the response of the CDR API is presented in Table 3.

Table 3 – CDR API response in JSON format


```
{
"data": [
{
"id": "string",
"start_date_time": "2021-03-18T17:29:31.043Z",
"stop_date_time": "2021-03-18T17:29:31.043Z",
"auth_id": "string",
"auth_method": "AUTH_REQUEST",
"location": {
"id": "string",
"type": "ON_STREET",
"name": "string",
"address": "string",
"city": "string",
"postal_code": "string",
"country": "string",
"coordinates": {
"latitude": "string",
"longitude": "string"
},
"evses": [
{
"uid": "string",
"evse_id": "string",
"status": "AVAILABLE",
"capabilities": [
```

"CHARGING_PROFILE_CAPABLE"
],
"connectors": [
{
"id": "string",
"standard": "CHADEMO",
"format": "SOCKET",
"power_type": "AC_1_PHASE",
"voltage": 0,
"amperage": 0,
"max_electric_power": 0,
"tariff_id": "string",
"terms_and_conditions": "string",
"last_updated": "2021-03-18T17:29:31.043Z"
}

4.1.2.3 Tariff API

This API is linked with the Billing engine (detailed in Section 4.1.1). The O-V2X-MP platform will offer powerful mechanisms for creating political invoices and multi-category charges that can be assigned to different chargers or groups of chargers (e.g., city center charger – expensive and charger in the suburbs – cheaper) or even to different plugs of the same charger (different tariff on AC socket than DC CCS or CHAdeMO connectors). The following are the tariff categories supported:

1) *CPO tariffs (Wholesale EV charging session costs)*: They correspond to the charges applied by the administrator of the charger. E.g., CPO cost = energy cost + management cost (maintenance & platform) + profit. This tariff can be set in a unit of energy (price per kWh) with one or two digits (with corresponding second digit activation threshold) or in time unit (value per minute) again with two strands, or a combination of the two units.
2) *eMSP tariffs (Retail Packages)*: They relate to the costs of the eMSP. There are the same possibilities as CPO costs for the initial charge – energy – time. Here the Wholesale price factor feature is added: CPO costs can either be calculated and eMSP charges added: Either completely ignore the cost of CPO and set the charges completely flat of the eMSP (e.g., to create a tariff for subscribed users who pay a large monthly fee subscription and have a very low cost per charge).

4.1.2.4 CRM API

The CRM API is used to interact with the EV users and to receive feedback and input that will result in the improvement of the O-V2X-MP platform. The API contains all the available fields where the EV user enters his/her personal customer details, which are then managed by the Platform Owner or the eMSP.

4.2 Frontend

The frontend of the O-V2X-MP platform shall include features such as a map of available charging stations, the ability to search for specific stations by location or availability, and the ability to start and stop charging sessions. It shall also includes features for managing payments and monitoring charging activity in real-time. Moreover, it communicates with the backend of the system, in order to manage the charging stations and processes payments. Figure 5 illustrates a map of existing charging stations in the Attica area as well as specifically in Mesogeia region [11] where the Greek Use Case tests and demonstrations area will be taking place.

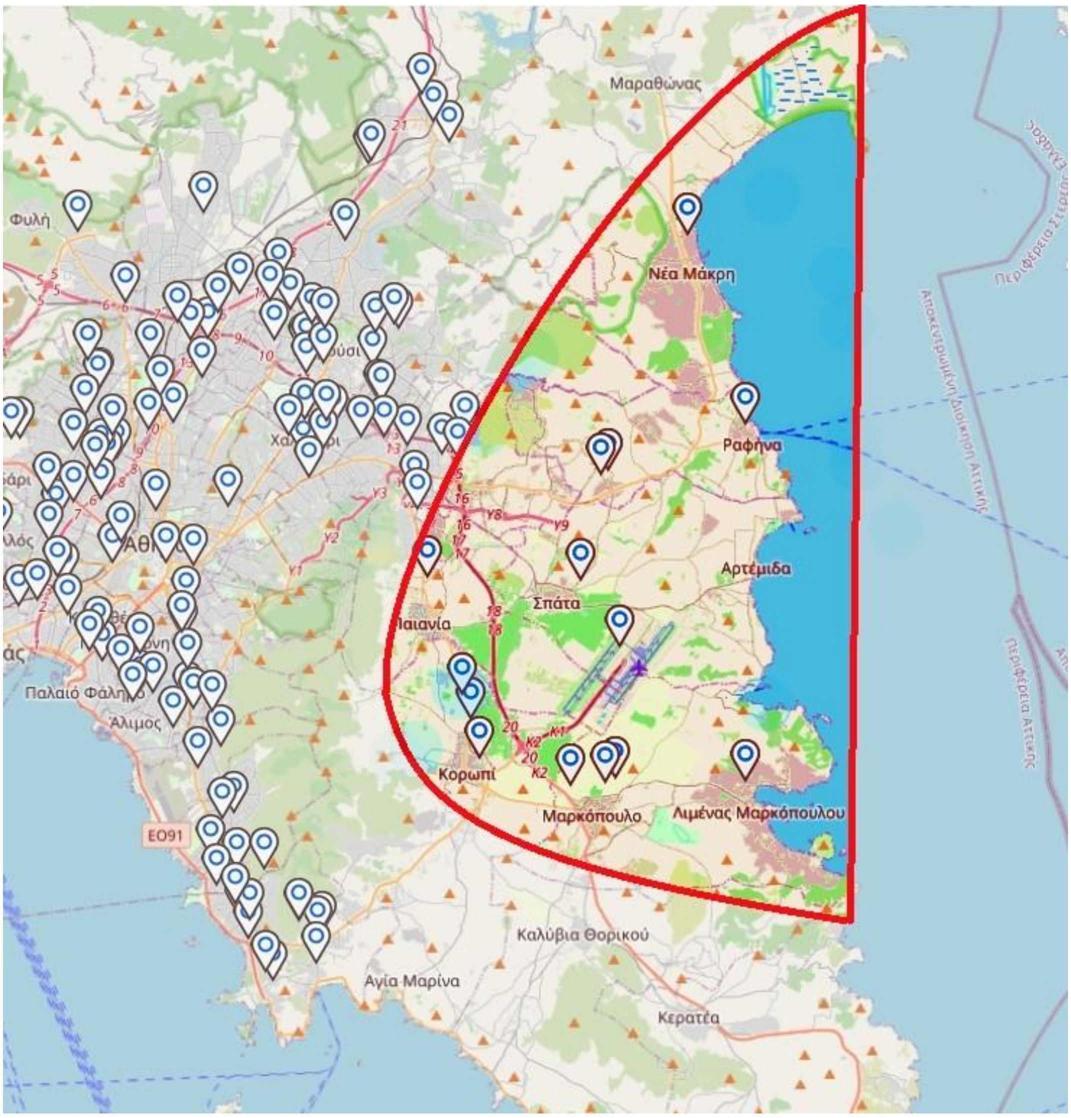

Figure 5 – Map of charging stations in O-V2X-MP platform for the Greek demo site in the Mesogeia region

Similarly, the EV chargers for the Slovenian demonstrator site at the business office building in Krško will appear in the O-V2X-MP frontend interface. An indicative map of the Krško area as it shall be visible in the O-V2X-MP frontend is illustrated in Figure 6.

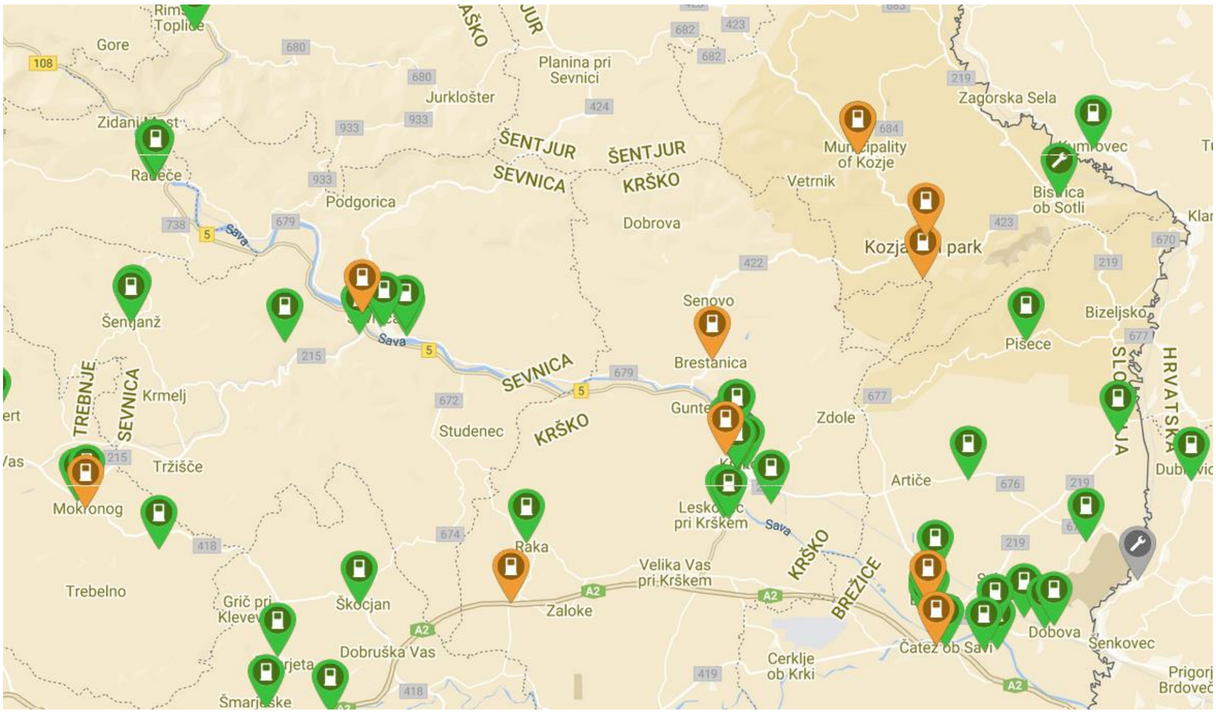

Figure 6 – Map of charging stations in O-V2X-MP platform for the Slovenian demo site

Additionally, the view for the Portuguese demonstrator in São Miguel, Azores that will be visible in the O-V2X-MP frontend interface is illustrated in Figure 7.

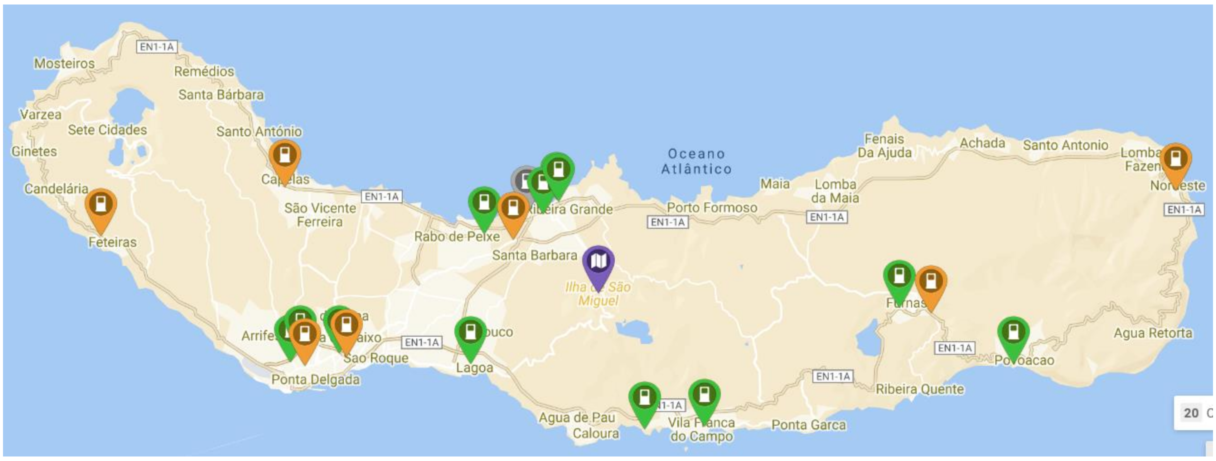

Figure 7 - Map of charging stations in O-V2X-MP platform for the Portuguese demo site

The O-V2X-MP platform will be also able to provide a street view for the locations where the chargers are installed. Such view of the envisaged interface is illustrated in Figure 8.

Figure 8 - Street view in O-V2X-MP platform for the Danish demo site (based on [11])

The O-V2X-MP frontend implementation is divided into an operational view where the Platform Manager as well as partners involved in the demos will be able to visualize the status of the connected charging stations in real-time, and a customer-centric view through which users of the platform can be able to visualize their charging sessions and provide feedback on the platform and its interfaces. The latter will be enabled through a CRM engagement tool that will allow you to interpret users' profile and get information about their frequency of use and customer behavioural analytics.

Specific frontend requirements are:

1) The operator should be able to add both OCPP 2.0.1, OCCP 1.6 and OCPP 1.5 charging stations.
2) The operator should be able to visualize the active charging sessions.
3) The Meter Values (current, voltage and active and reactive power in kWh) should be available during the charging session.
4) The frontend should provide the list of locations where the EV user can charge its vehicle based on the charging station map.
5) Dynamic tariff declaration and adjustment from the operator based on electrical load demand and availability in each EV charging location.
6) Charging station diagnostics shall be provided for each charging station.
7) The start and end time of charging sessions shall be available.
8) Measurement data of consumed energy for each charging session should be available and the operator shall be able to download them.
9) The Details of the token (RFID card number) used for user identification shall be visible.
10) Climate conditions and geographical location data for the EV chargers shall be reflected during the charging session as well as after it has finished.
11) A dashboard shall be available where widgets and statistics can be added an analysed such as the electricity usage (kWh) and the number of charging sessions in Figure 9.

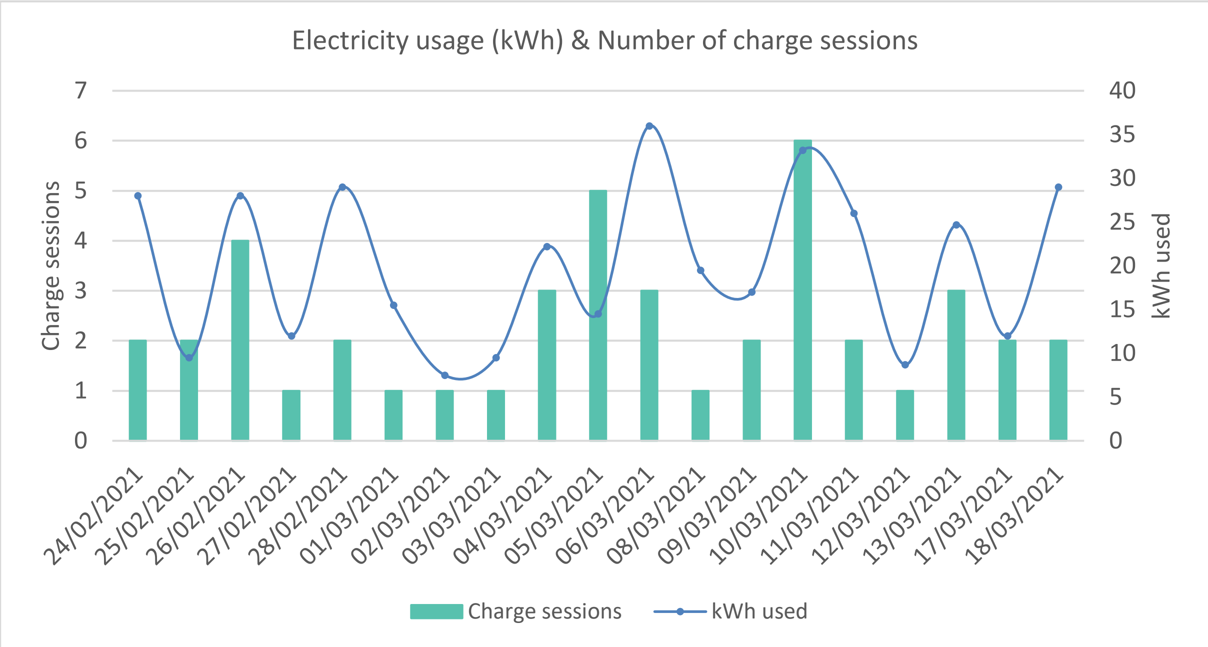

Figure 9 – Frontend dashboard widget with total electricity usage and EV charging sessions

4.3 Current platform implementation

The current version of the platform is based on SteVe [16] that is an open-source EV charging point administration platform developed in 2013 by research assistants in the IDSG group of RWTH Aachen University, that is part of the chair Informatik 5. The current version implements the following scenario for OCPP 1.6. Driver connects their EV to a charging station. The CPO checks whether this card is authorized to charge on this charge station. If it is, the system will start the charging session. As a frontend interface the current version has a very simple view containing the number of charging points, the number of received heartbeats as well as the connector status (Available/Unavailable) as illustrated in Figure 10.

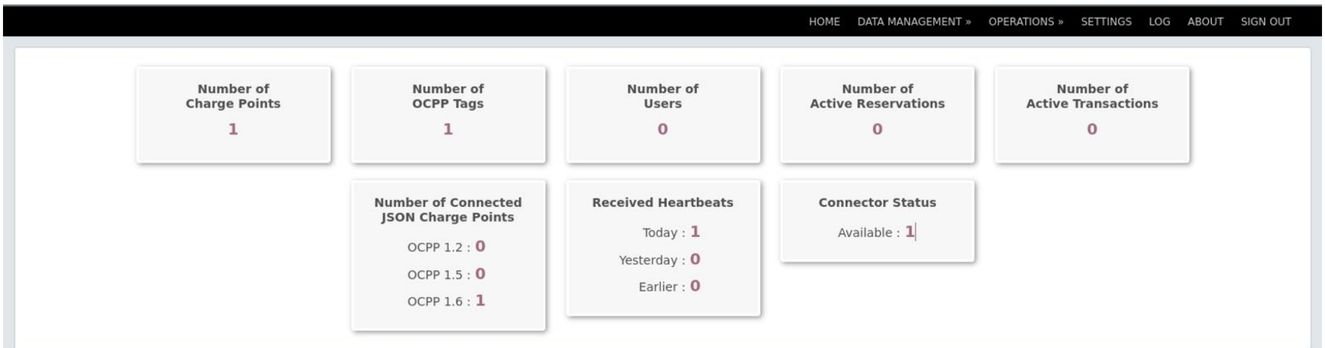

Figure 10 – Current frontend interface of the current O-V2X-MP platform based on [16] (to be updated)

Through the existing implementation the operator can obtain the details of a charging station as well as the active transactions or reservations based on OCPP 1.5 or 1.6. The details of an example charging station based on OCPP 1.6 from ABB charge point manufacturer are illustrated in Figure 11.

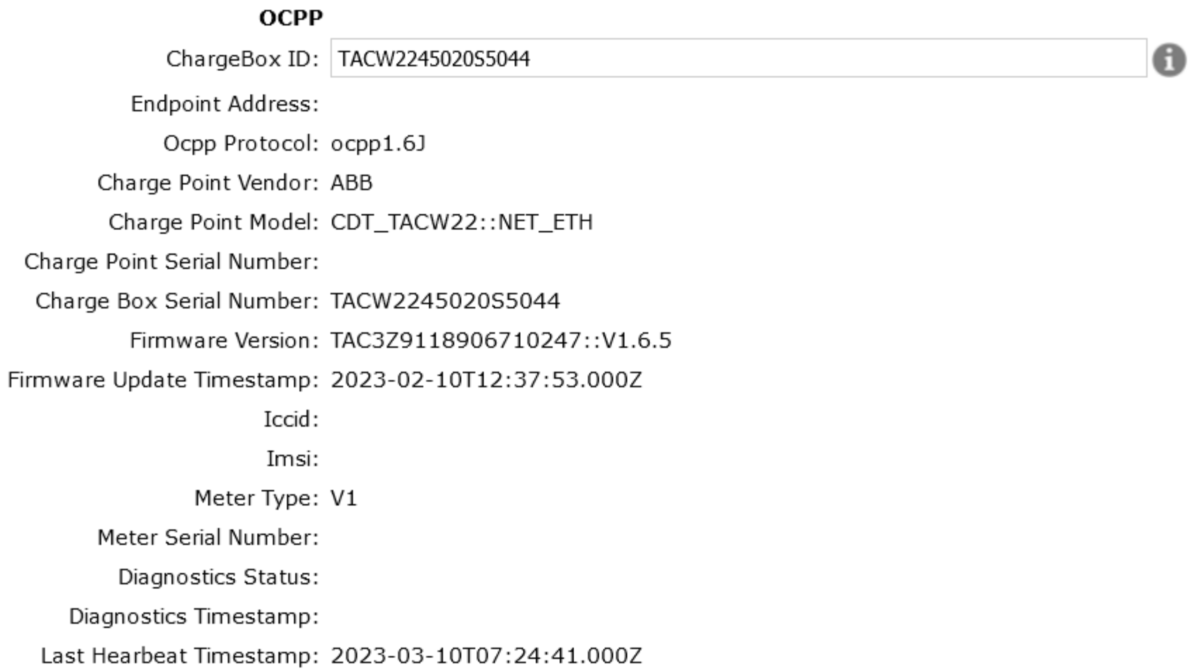

Figure 11 – Charging point details in the current O-V2X-MP platform

The messages and commands that are exchanged through the platform can be analysed through Wireshark and a plugin that was implemented in the scope of the EV4EU project to interpret WebSocket commands. Figure 12 illustrates the packet capture in Wireshark by analysing a "Hard reset" command through the O-V2X-MP platform towards the EV charging station.

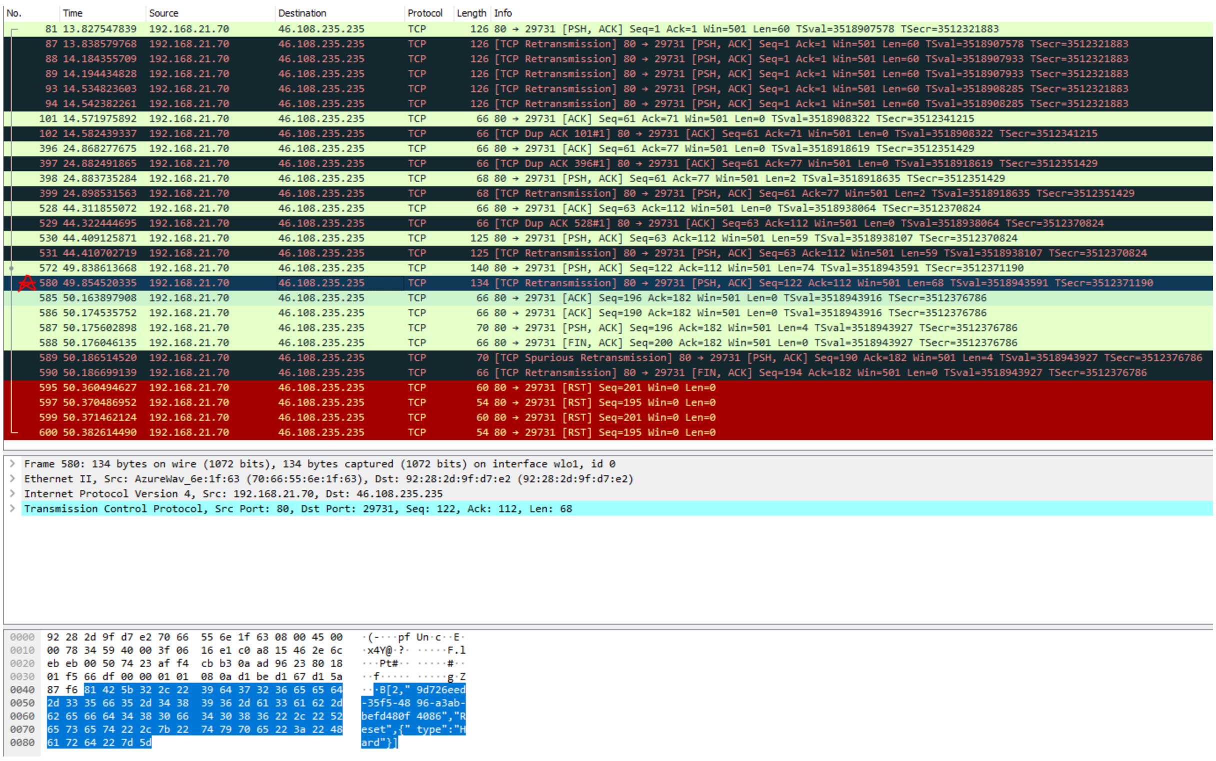

Figure 12 - OCPP network packet analysis of the current O-V2X-MP through Wireshark

The development on SteVe is mainly based on Java programming language [16] for the backend and uses JavaScript notations for visualizations as a part of the frontend. From the beginning of the EV4EU project till recently we have introduced extensions on SteVe, to improve the OCPP commands used for communication and to parse and encapsulate OCPP commands in JSON-based WebSocket messages, in order to be saved in PCAP format and analysed through network monitoring tools, such as the Wireshark network analyser [17]. However, during this period we have identified the following development difficulties:

1) The development environment is complex and cannot be easily extended, which is reasonable as it was initially developed as a research project.
2) Development issues were faced during by implementing the initial OCPP 2.0.1 commands, which require severe modifications on the Java based backend.
3) SteVe does not have an active development community, since 2019 the project is not actively developed within the university and research group.

Due to these issues we have decided in the course of project task T5.3 “Open V2X Management Platform - Architecture” to change the O-V2X-MP implementation to a Python based implementation for the backend, which will be based on the Django framework [5]. Furthermore, the frontend will be based on Bootstrap 5 [23]. The APIs will also be developed in Python and documented through the Swagger interface [18].

5 V2X compatibility

This section focuses on the main elements that the O-V2X-MP platform should have in order to enable V2X compatibility. Initially the modes that should be supported are described and then example of V2X communication flows are provided.

5.1 Supported modes

The O-V2X-MP platform will be designed to support the following modes:

1) *ChargingOnly*, which allows charging only, therefore is not V2X at all. It is added especially for Charging Stations that intent to operate in V2X but are unsure during energy service negotiation if the EV and EV user can operate in V2X at this time and location. It then starts in this operation mode, waiting for authorization of the V2X operation mode.
2) *ExternalSetpoint*, which tells the Charging Station that the setpoint parameter is to be determined by some external actor, such as an Energy Management System (EMS). CSMS submits the charging profile and leaves the setpoint parameter empty. Its value should then be received by the Charging Station from the external system through some other means of communication and not via OCPP. This scenario is further described in Deliverable D5.2 "Standardization gap analysis for new V2X related Business Models" [2].
3) *CentralSetpoint*, which is used by the CSMS to set a single setpoint or profile for charging and/or discharging, using the setpoint parameter. Positive setpoints requests for charging, while negative setpoints requests for discharging. A single setpoint (when only one entry) or profile might be defined by a secondary actor such as an aggregating party that relays the message through the CSMS. This scenario is further described in Deliverable D5.2 "Standardization gap analysis for new V2X related Business Models" [2].
4) *CentralFrequency*, where the setpoint for frequency support is determined by the CSMS, for instance when costly calibrated frequency measurements are to be used that cannot be installed in each Charging Station. The CSMS will have to continually update the setpoint when the frequency changes, using the setpoint field. When the Charging Station does not receive a new setpoint within time, it can fallback to local frequency operation mode.
5) *LocalFrequency*, where the power setpoint for frequency support is determined from a power/frequency table, based on the locally measured frequency.
6) *LocalVoltage*, where the reactive power setpoint for frequency support is determined from a voltage/power-factor (Q(U)) table, based on the locally measured voltage of the grid connection.
7) *LocalLoadBalancing*, which allows an EV to be utilized for load balancing, for example for a building that both consumes energy and produces energy from solar panels. If the Charging Station can read the smart energy meter of the building, then the Charging Station and EV setup can influence the resulting load on the grid connection. For this operation mode there are four configurable keys:
    a. *UpperThreshold*, threshold level for the measured load at which the EV will start to compensate the power of the load in order to avoid exceeding the threshold (by discharging).
    b. *LowerThreshold*, threshold level for the measured load at which the EV will start to compensate the power of the load in order to avoid passing below the threshold (by charging).
    c. *UpperOffset*, the amount by which the load is compensated more (or less) than the upper threshold.
    d. *LowerOffset*, the amount by which the load is compensated more (or less) than the lower threshold.

The determined setpoint in the V2X schedule entry is always to be limited by charge limit (when positive) or discharge limit (when negative).

8) *Idle*, which is used when the EV is to be idle, in order to allow to minimize the energy used by the Charging Station and EV for a period of time. The EV is able to go to sleep completely or can also be on standby. The difference between to sleep and on standby is that for to sleep it is not expected of the EV to react quickly on a new schedule, while for standby it is expected that the EV can quickly start charging or discharging when for example a new schedule is sent from the CSMS to the Charging Station.

These modes will be tested based on the BUCs of the project that are detailed in D1.5 "V2X Use-Cases Repository" [8].

5.2 Interfaces and communication flows

To aid in describing the interfaces and communication flows of the O-V2X-MP platform we first focus on providing an ERD with all the platform entities as well as their corresponding relationships. This diagram is illustrated in Figure 13.

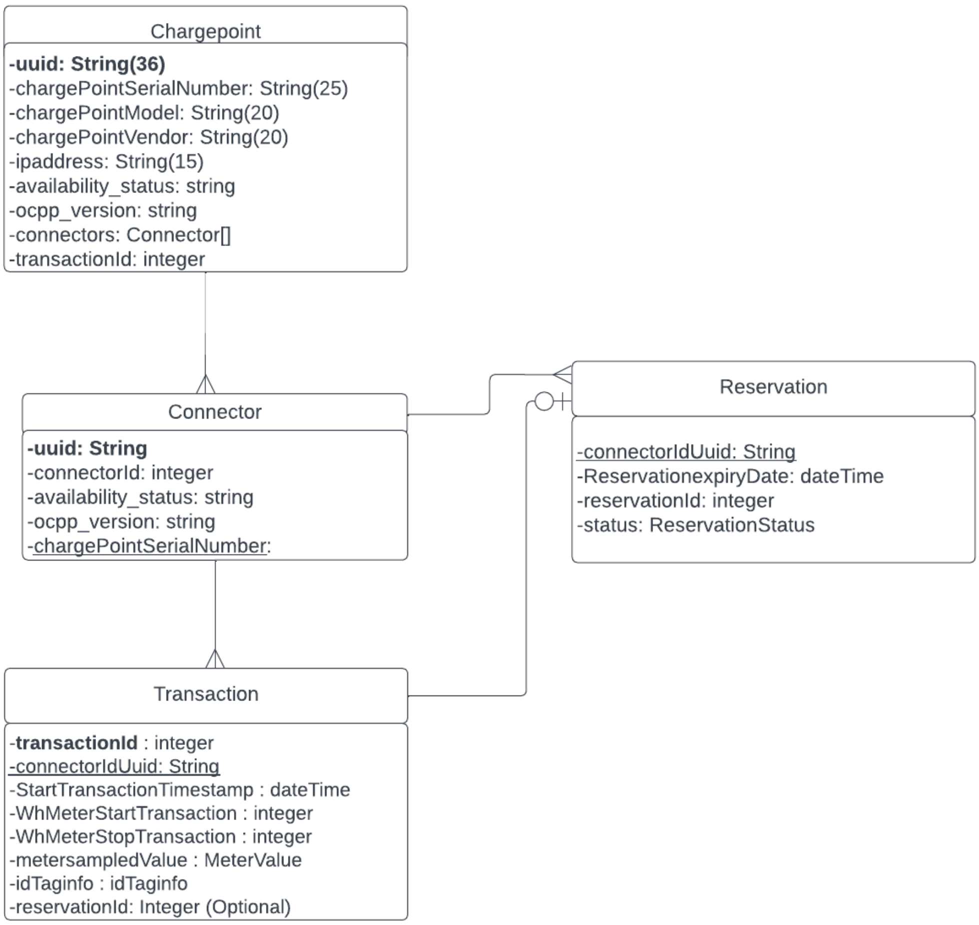

Figure 13 - ERD relationship diagram of the O-V2X-MP platform

The presented ERD explains the relationship between the strong classes which are ChargePoint and Connectors, and the weak classes which are Transaction and Reservation.

- *ChargePoint*: The main class ChargePoint represents the physical system where an EV can be charged. A Charge Point has one or more connectors. To be functional, Charge Point must have specific attributes. The first one is uuid (universal unique identifier) which is its unique serial number. The next two are chargepointModel and chargepointVendor which add up to the model and to the vendor of Charge Point. Next is IP address which is necessary for the communication of ChargePoint with the web. Another useful attribute is the availability status which adds up to the status of ChargePoint and it can take three values (Available, Unavailable, Faulted). Another attribute is ocpp_version indicating which OCPP version with values (1.6 or 2.0.1) is used for the communication between ChargePoint and CSMS and EV users. The last two attributes are connectorId and transactionId that indicate the ID of which connector is available and if there are any transactions about to start.
- *Connectors*: Is another strong class which represents an independently operated and managed electrical outlet on a Charge Point. Every connector has the following attributes. First and foremost is the uuid attribute that renders its id unique. Another attribute is the availability_status which takes values such as (Available, Preparing, Charging, SuspendedEVSE, SuspendedEV, Finishing, Reserved, Unavailable, Faulted). The chargePointSerialNumber is a part of the attributes which indicate who ChargePoint belongs to. The final attribute is the ocpp_version as described above.
- *Transaction*: The weak class Transaction represents the part of the charging process. Its basic attribute is the transactionId which is unique for every transaction. The transaction depends on the connectorIdUuid of the connector which executes the transaction. Other attributes are StartTransactionTimestamp and StopTransactionTimestamp which indicate the date and time when a transaction begins and ends. Included in the attributes are the WhMeterStartTransaction and WhStopMeterTransaction which show the Wh when the transaction begins and ends. Another attribute is the idTaginfo which indicates the user's ID who conducts the transaction and finally the attribute reservationId exists in case there has been a reservation.
- *Reservation*: The weak class Reservation represents a charger's function where the user can make a reservation for a future transaction. Included in the attribute is the connectorUuid which indicates the connector's uuid which executes the transaction. Another attribute is the ReservationexpiryDate which shows the date when the reservation expires. Furthermore, there is the reservationId that displays the reservation's ID which is unique for every case. Finally, there is the reservations' s status which takes values such as (Accepted, Faulted, Occupied, Rejected, Unavailable).

The relationships between the classes of Figure 13 are as follows:

- *ChargePoint* class presents the main attributes that are necessary for the operation and communication with EV users and CSMS of the ChargePoint. As we see a ChargePoint allocates several connectors, although a connector with a UUID maps to one and only ChargePoint.
- *Connectors***:** A connector with a UUID maps to one and only ChargePoint. Also, connector can occur many transactions, although a transaction can be occurred by only one connector.
- *Reservation* can be served by only one connector and in the same way can serve one transaction.
- *Transaction* can take place whether a reservation has been made or not, that's why it is optional. A transaction can be served by one reservation.

To better explain the classes as well as the corresponding relationships we also provide an example of the transaction functionality divided in steps. In order to easily recognize the charger, we allocate the ChargerPointSerialNumber = ChargerPoint uuid. The example then is as follows:

1) There is an EV driver who uses his RFID card [idtaginfo(RFID)] to login to the charger which has a serial number ChargePointSerialNumber(**670639**) and ChargePoint uuid(**670639**) and model ChargePointModel(**DCF1-212-84600000**) from the vendor ChargePointVendor(**com.wallbox**) which has the ipaddress(**190.186.188.134**).
2) When the EV driver completes the login, a message with the status of the availability of charger [availability_status(Available)] is displayed, as well as the status availability of the connector's [availability_status(Available)]. This connector which belongs to Chargepoint with ChargePointSerialNumber(**670639**), has an ID connectorId(ID=2).
3) The transaction with transactionId(**193082**) starts from connector with id[connectorId(ID=2)] at the date and time that is displayed [StartTransactionTimestamp(**2023-04-11T14:08:26+03:00**)] with the start value of Wh [WhMeterStartTransaction(**1.121 kWh**)].
4) The user is informed regarding the date and time that the transaction with id [transactionId(**193082**) is completed StopTransactionTimestamp(**2023-04-11T15:06:40+03:00**) with the end value of Wh[WhMeterStopTransaction(**0.817 kWh**)].

The communication flows that are developed in the platform will be defined based on communication flow diagrams linked with the UC of D1.5 as well as with the Actors and selected BUCs that are described in deliverable D5.1 "Information Exchange needs to enable different UCs". The flows will be developed in the UML language and an example involving the EV, charging station and CSMS actors for set the charging schedule is illustrated in Figure 14.

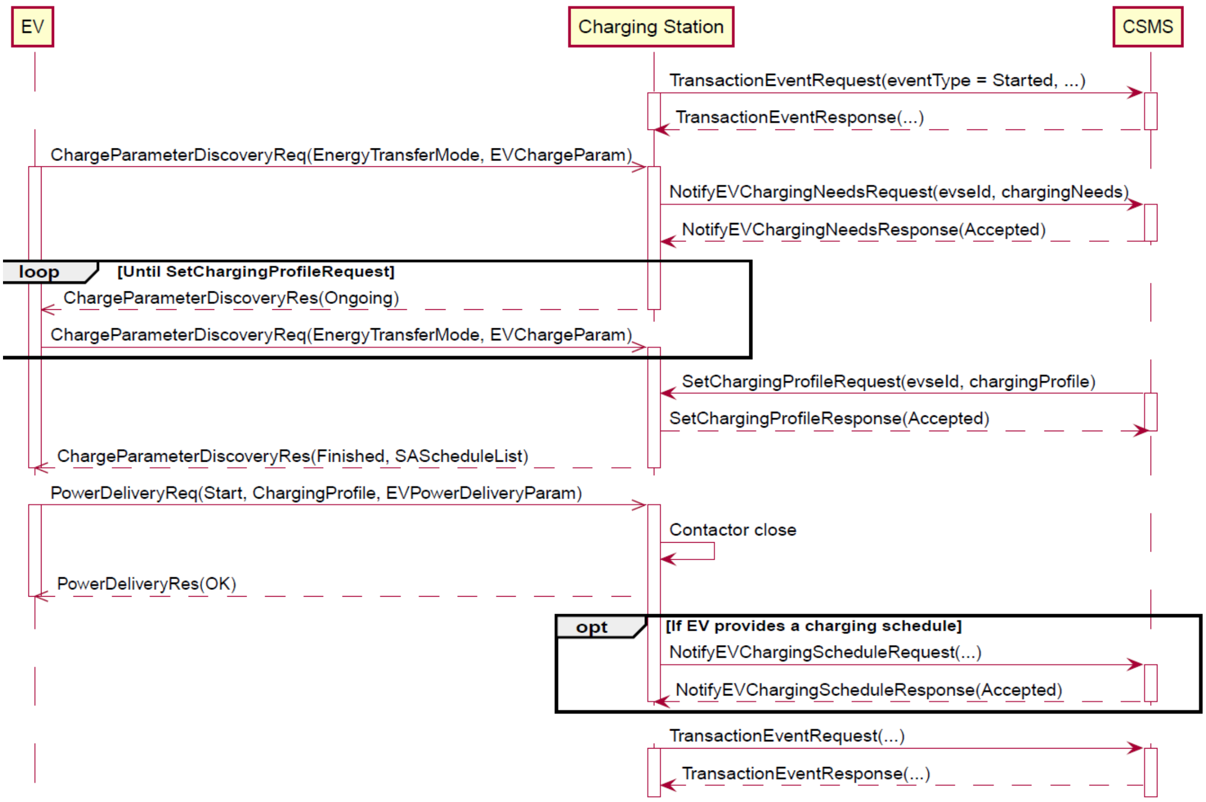

Figure 14 – Communication flows for the entities involved in the O-V2X-MP platform

6 Conclusions

This deliverable provides a high-level overview of the architectural design for the O-V2X-MP platform. The design is initially based on the functional architecture as well as the user groups that the under-development O-V2X-MP platform will include. Then, an overview of the high-level requirements is provided, which are split into the backend implementation of all the communication mechanisms and the frontend GUI interface and the list of all the APIs that will be part of the backend allowing the integration with further layers as the environmental layer, the security layer as well as the user layer. The interactions were also provided through interfaces, communication flows and relational classes in ERDs.

At the current phase of the project the list of requirements is not extensive and will evolve in the duration of the project and during the design and planning of the BUC scenarios. Hence, the emphasis was given to the communication interfaces and the interactions of the different actors in the V2X ecosystem, which translates to requirements for the backend, the frontend as well as APIs for the interactions with the rest of the layers. In certain cases, the interaction is present as in the security layer through the usage of security mechanisms in V2X communications and these mechanisms will be further detailed in deliverable D5.4 “Cyber-security and Privacy analysis for V2X services”.

The presence of the environmental and user layer require input from the UCs in different locations in Portugal, Greece, Slovenia and Denmark. Preliminary feedback has been taken into consideration from deliverable D3.1 “EV Users’ Needs and Concerns -Preliminary Report”, but further work will be done in the progress of the project as well as WP3. Additionally, the complete list of communication flows and ERD relationship diagram for all the BUC scenarios will be part of a report that will be also part of the open-source software tool and both will form deliverable D5.5 “Open V2X Management Platform”.

References

[1] M.Zajc et al, "Deliverable D5.1: Information Exchange to enable different UCs", Electric Vehicles Management for carbon neutrality in Europe (EV4EU) Horizon Europe funded project, Grant Agreement 101056765, Ref.Ares (2023) 1469491, 2023

[2] A. Lekidis et al., "Deliverable 5.2: Standardization gap analysis for new V2X related Business Models", Electric Vehicles Management for carbon neutrality in Europe (EV4EU) Horizon Europe funded project, grant agreement 101056765 (submitted), 2023

[3] Open Charge Alliance, OCA-Overview-OCPP-versions, 2021, Online: https://www.openchargealliance.org/uploads/files/OCA-Overview-OCPP-versions.pdf

[4] ISO/IEC 14443-3, "Cards and security devices for personal identification — Contactless proximity objects — Part 3: Initialization and anticollision", 2018, [Online]: https://www.iso.org/standard/73598.html

[5] Django framework, [Online] : https://www.djangoproject.com/

[6] Ravidas, S., Lekidis, A., Paci, F. and Zannone, N., 2019. Access control in Internet-of-Things: A survey. Journal of Network and Computer Applications, 144, pp.79-101.

[7] OCPP 1.6 Specification, [Online] : https://www.oasis-open.org/committees/download.php/58944/ocpp-1.6.pdf

[8] H. Morais et al., "D1.5, V2X Use-Cases Repository", Electric Vehicles Management for carbon neutrality in Europe (EV4EU) Horizon Europe funded project, grant agreement 101056765 (in preparation), 2023

[9] M. Mültin, "IEC 63110 - standardising the management of EV (dis-)charging infrastructures," 27-Apr-2017. [Online]: https://www.switch-ev.com/blog/iec-63110-standardizing-management-of-ev-charging-infrastructures.

[10] Van Den Bossche, P. (2010). IEC 61851-1: Electric vehicle conductive charging system-Part 1: General requirements. In *2* (pp. 1-99). IEC.

[11] Panagiotis Pediaditis et al., "Deliverable D1.3, Regulatory opportunities and barriers for V2X deployment in Europe", Electric Vehicles Management for carbon neutrality in Europe (EV4EU) Horizon Europe funded project, grant agreement 101056765 (submitted), 2023

[12] Hubject roaming platform, [Online]: https://www.hubject.com/

[13] Gireve roaming platform, [Online]: https://www.gireve.com/

[14] OCPI 2.2: Open Charge Point Interface, [Online]: https://evroaming.org/app/uploads/2020/06/OCPI-2.2-d2.pdf

[15] Hubject: Open Intercharge Protocol (OICP) for Charge Point Operators, Version 2.2 / March 2018, [Online]: https://assets.website-files.com/602cf2b08109ccbc93d7f9ed/60534f2e20d0f87be17ba21b_oicp-cpo-2.2.pdf

[16] SteVe EV charging station administration platform, [Online]: https://github.com/steve-community/steve

[17] Wireshark protocol analyzer, [Online]: https://www.wireshark.org/

[18] Swagger REST API documentation, [Online]: https://swagger.io/tools/swagger-ui/

[19] Mültin, Marc. "ISO 15118 as the Enabler of Vehicle-to-Grid Applications." 2018 International Conference of Electrical and Electronic Technologies for Automotive. IEEE, 2018.
[20] Open Charge Alliance, "OPEN CHARGE POINT PROTOCOL 2.0.1", [Online]: https://www.openchargealliance.org/protocols/ocpp-201/
[21] Lombardi, Andrew. WebSocket: lightweight client-server communications. " O'Reilly Media, Inc.", 2015.
[22] Snader, Jon C. VPNs Illustrated: Tunnels, VPNs, and IPsec: Tunnels, VPNs, and IPsec. Addison-Wesley Professional, 2015.
[23] Bootstrap 5 Introduction, [Online]: https://getbootstrap.com/docs/5.0/getting-started/introduction/