UPPSALA
UNIVERSITET

UPTEC ES 25029
Examensarbete 30 hp
Juli 2025

Direct Communication from DSOs to EVSE for Demand Response Solutions

Gustaf Ferntoft

Civilingenjörsprogrammet i energisystem

UPPSALA UNIVERSITET

Direct Communication from DSOs to EVSE for Demand Response Solutions

Gustaf Ferntoft

Abstract

The Swedish government has set ambitious climate goals, aiming for net-zero emissions by 2045. At the same time, Sweden is undergoing intense electrification, causing increasing strain on the power grid. One key solution to this challenge is the use of flexible resources, such as Electric Vehicles (EVs). To manage EVs efficiently, secure, standardized, and reliable communication protocols are essential. A particularly interesting opportunity lies in establishing direct communication between Distribution System Operators (DSOs) and Electric Vehicle Supply Equipment (EVSE), enabling more efficient and responsive grid management.

The aim of this project is to investigate communication protocols that facilitate the direct interaction between DSOs and EVSE, or indirectly through third parties, such as aggregators, for demand response purposes. Several existing protocols are analyzed and compared, and OpenADR is found to be the most suitable solution for implementation in Sweden. Interviews with key actors provide valuable insights into the potential of OpenADR in Sweden. Additionally, the OpenADR Alliance provides access to their OpenADR 3.0 Virtual Top Node Reference Implementation, serving as a foundation for system development.

System development in this project includes the implementation of both the OpenADR Business Logic client and the Virtual End Node client, each equipped with a user interface for clarity. The system is designed to support two test cases - charging curtailment and Vehicle-to-Grid (V2G) capacity limitation, both of which enable direct communication between a simulated DSO and the EVSE. These use cases are tested in a local environment, to evaluate the system's performance.

The results show that the EVSE responded correctly in both scenarios, reducing the charging current during curtailment and limiting export power during the V2G test, and returned to normal operation upon test case expiration. A key finding from the results is the viability of the V2G capacity limitation use case. The system operated as intended, demonstrating OpenADR's potential to support the deployment of V2G technology in Sweden. Additionally, the V2G capacity use case can also fulfill grid code requirements, further increasing its relevance and applicability.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: Ali Fotouhi Ämnesgranskare: Valeria Castellucci

Examinator: Carla Puglia

Populärvetenskaplig sammanfattning

Sverige har satt upp ambitiösa klimatmål, där landet ska ha 100% förnybar elproduktion till år 2040 och ha netto-nollutsläpp av växthusgaser till år 2045. För att nå dit pågår en omfattande elektrifiering av samhället. Enligt framtidsscenarier förväntas Sveriges elanvändning mer än fördubblas, från 132 TWh år 2023 till 310 TWh år 2045. Denna utveckling beror till stor del av elektrifieringen inom den svenska industrin, men även elbilarnas snabba framväxt spelar en betydande roll. Samtidigt som efterfrågan på el växer i rasande takt, går utbyggnaden och upprustningen av elnäten betydligt långsammare. Att förstärka eller bygga ut nya elledningar kräver ofta tillståndsprocesser som idag kan ta mellan sju och femton år. Resultaten av detta blir att elnäten på flera håll i Sverige inte klarar av att transportera all den el som samhället efterfrågar - ett problem som leder till kapacitetsbrist och hämmar elektrifieringen.

För att möta denna utmaning behöver Sveriges distributionsoperatörer (DSO:er), med andra ord regionnätsägare, ta i bruk smarta tekniska lösningar som gör det möjligt att styra elförbrukningen mer flexibelt. En sådan lösning är att använda elbilar som flexibla resurser. Då elbilar laddas på olika platser och vid olika tider under dygnet kan de, med korrekt styrning, bidra med så kallad efterfrågeflexibilitet, vilket innebär att anpassa sin elförbrukning efter elnätets behov. För att detta ska fungera krävs dock att kommunikationen mellan DSO:er och elbilsladdare är både säker och tillförlitlig, något som kan uppnås med standardiserade kommunikationsprotokoll. I dagsläget saknas ett gemensamt beslut kring en sådan standard i Sverige, vilket försvårar utvecklingen.

Detta projektet har därför undersökt vilket kommunikationsprotokoll som lämpar sig bäst för att etablera en direkt kommunikationslänk mellan DSO:er och elbilsladdare. En jämförelse mellan flertalet alternativ genomfördes, varifrån kommunikationsprotokollet OpenADR valdes som bäst lämpat för implementation i Sverige. OpenADR är ett internationellt erkänt protokoll som är särskilt utformat för efterfrågeflexibilitet. Protokollet används även redan i viss utsträckning i Sverige och har dessutom föreslagits som en industristandard av branchorganisationen Energiföretagen. För att fördjupa kunskapen om OpenADR genomfördes intervjuer med svenska DSO:er, nyckelaktörer i Sverige samt experter inom protokollet. Den insamlade kunskapen användes dels för att kartlägga hur OpenADR uppfattas av aktörer i Sverige, dels som underlag för att utveckla ett fungerade kommunikationssystem baserat på OpenADR.

Systemet som utvecklades simulerar kommunikationen mellan en DSO och en elbilsladdare. Eftersom detta projekt inte genomfördes i samarbete med en verklig DSO, byggdes systemet för att kunna simulera de funktioner en DSO vanligtvis står för. Systemet byggdes för att testa två olika praktiska scenarier. Det första scenariot testade att minska elbilens konsumtion genom att begränsa dess laddning, ett fall praktiskt vid hög belastning i elnätet. Det andra scenariot testade att begränsa elbilens export av el till elnätet genom tekniken Vehicle-to-Grid (V2G), en teknik som innebär att batteriet i en elbil är dubbelriktat och fungerar i båda riktningar, både som mottagare och leverantör av el. Dessa två scenarier testades individuellt i Volvo Cars laddningslabb, där en styrsignal skickades från den simulerade DSO:en till elbilsladdaren. I båda fallen kommunicerade styrsignalen information som elbilsladdaren anpassade sig efter genom att antingen minska elbilens konsumtion eller minska dess elexport, beroende på testfall.

Resultaten från bägge testfallen visade att systemet fungerade som förväntat. Elbilsladdaren följde styrsignalens instruktioner, både vid minskning av laddning och vid begränsningen av elexport. När styrsignalens tidsperiod var över övergick även elbilsladdaren tillbaka till ursprunglig drift. Detta bekräftar att OpenADR kan användas som kommunikationslösning i Sverige, även för mer avancerade tillämpningar som V2G. Dessutom visade testerna att lösningen uppfyller flera tekniska krav som anges i de nätkoder som reglerar elsystemets funktion,

ett viktigt steg mot att göra tekniken redo för storskalig användning. Dessa faktorer påvisar således att användning av OpenADR skulle kunna förenkla den framtida implementeringen av V2G i Sveriges elsystem.

Samtidigt framkom det tydligt i intervjuerna att svenska DSO:er anser att den mest effektiva kommunikationsvägen med en elbilsladdare vore genom en aggregator, snarare än genom direkt kommunikation. Detta beror på att en aggregator bidrar med resurser från många flexibla enheter, vilket ger DSO:er tillgång till en större kapacitet. Genom att ha en aggregator som mellanhand ökar även driftsäkerheten och den tekniska kompetensen. Aggregatorer möjliggör dessutom en mer flexibel och effektiv hantering, genom att kombinera olika typer av resurser som kan delta i olika efterfrågeflexibilitetsprogram.

För framtida utveckling och studier rekommenderas flertalet steg. Ett huvudmål skulle vara att genomföra ett pilotprojekt i samarbete med en svensk DSO. För att det ska vara möjligt behöver dock systemet vidareutvecklas, genom att införa auktorisering för ökad säkerhet och implementera funktionalitet för rapportering av realtidsdata. Det vore också värdefullt att anpassa systemet för att kunna styra flera elbilsladdare samtidigt, således möjliggöra aggregation, samt att skapa kompatibilitet med andra kommunikationsprotokoll som OCPP eller EEBus. På så vis kan systemets flexibilitet öka ytterligare och öppna upp för flertalet andra användningsområden, vilket gör systemet ännu mer användbart i framtidens elsystem.

Det som gör detta projektet särskilt intressant är att det inte enbart har undersökt hur elbilar kan styras i teorin, utan också testat tekniken i praktiken, med lyckade resultat. OpenADR används redan småskaligt i Sverige för att minska elförbrukningen, men inte i kombination med V2G. Detta projekt visar att en sådan lösning är teknisk möjlig. Ett nästa steg är att utveckla affärsmodeller som gör det möjligt för aktörer att ta tekniken i bruk, något som var utanför ramarna i detta projekt. Om OpenADR implementeras storskaligt i Sverige kan det både förenkla införandet av V2G och bidra till en mer effektiv elektrifiering av samhället, utan att elnätet behöver byggas ut i samma takt. Det skulle dessutom ge elkunder och aktörer nya möjligheter att bidra till balanser i elnätet, och samtidigt tjäna pengar på det.

Executive summary

This project investigates the direct communication between a simulated DSO and an EVSE. A comparison of several communication protocols is conducted, where OpenADR is proven to be the most promising communication protocol for adoption in Sweden. Based on this, a local communication system facilitating the direct communication using the OpenADR protocol is developed and tested for two key demand response use cases - charging curtailment and V2G capacity limitation.

The results demonstrate that the developed system operates as intended. In both scenarios the EVSE responded correctly, reducing its charging current during the curtailment test and limiting its power export during the V2G capacity limit test, before returning to default operation upon test case expiration. These outcomes confirm that OpenADR efficiently can support multiple use cases with minor payload changes, highlighting the flexibility and scalability of the protocol. Furthermore, the results confirm that OpenADR can support the V2G capacity limitation use case, providing the viability of the developed system and the protocol's potential to support future V2G deployment in Sweden. Additionally, this use case can fulfill V2G grid code requirements, further strengthening its relevance and applicability.

To build on this work, several next steps are recommended. A key objective would be to initiate a real-world test pilot in collaboration with an actual DSO. However, this would require further system development. Critical system improvements include implementing authorization through OAuth2 for enhanced security and implementing the report functionality to provide real-time data and performance insights to the DSO. Additionally, valuable future steps would be to extend the system to support multiple EVSE, enabling resource aggregation, and develop the system to support integration with other communication protocols, such as OCPP and EEBus. This added interoperability would increase system flexibility and scalability, allowing both direct and indirect communication pathways and expand its applicability across a wider range of use cases.

Förord

Med detta examensarbete är jag väldigt glad och stolt att kunna meddela att jag avslutar mina studier till civilingenjör i energisystem vid Uppsala universitet och Sveriges lantbruksuniversitet. Jag vill rikta ett stort tack till Volvo Cars i Göteborg som trodde på mig och gav mig chansen att utföra detta examensarbete. En speciellt tack till min handledare Ali Fotouhi, som genom hela arbetet bidrog med stöd, idéer och kunskap. Tack även till Johanna Trillkott som stöttat mig under resans gång, och till Virat Andodariya som gett mig ovärderligt stöd med systemutvecklingen. Jag vill också tacka min ämnesgranskare Valeria Castellucci för all vägledning och hjälp.

Jag riktar även ett stort tack till min familj och mina nära och kära för att ni funnits där för mig under min studietid. Till alla mina fantastiska vänner som jag träffat i Uppsala - tack för att ni gjort den här perioden i mitt liv oförglömlig.

Gustaf Ferntoft
Göteborg, juli 2025

Abbreviations

BL - Business Logic

CL - Customer Logic

CP - Charge Point

CPO - Charge Point Operator

DER - Distributed Energy Resource

DR - Demand Response

DSO - Distribution System Operator

e-MSP - E-mobility Service Provider

EV - Electric Vehicle

EVCC - Electric Vehicle Communication Controller

EVSE - Electric Vehicle Supply Equipment

HEMS - Home Energy Management System

HTTP - HyperText Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

IEC - International Electrotechnical Commission

IP - Internet Protocol

ISO - International Organization for Standardization

JSON - JavaScript Object Notation

PoC - Proof-of-Concept

RES - Renewable Energy Source

RESTful API - REpresentational State Transfer Application Programming Interface

SECC - Supply Equipment Communication Controller

UI - User Interface

U.S - United States

VEN - Virtual End Node

VTN - Virtual Top Node

VTN RI - Virtual Top Node Reference Implementation

VS Code - Visual Studio Code

V2G - Vehicle-to-Grid

V2X - Vehicle to Everything

XML - Extensive Markup Language

XMPP - Extensible Messaging and Presence Protocol

Contents

1 Introduction

In Sweden, the government has set the climate goals to achieve 100% renewable electricity production by 2040 and reaching net-zero emissions by the year 2045 [1]. At the same time, Sweden is undergoing rapid electrification, where the electricity demand is projected to increase significantly, from 132 TWh in 2023 to 310 TWh by 2045 [2]. This intensified electricity demand is primarily due to an increasing electrification in Swedish industry, but the growing adoption of Electric Vehicles (EVs) also contributes notably to the trend [3]. In parallel, Sweden's power grid experiences increasing challenges regarding grid congestion [4]. One solution to this problem is to reinforce and expand the grid infrastructure. However, this process is often slow and hindered by regulations and permits and can take between 7-15 years to get approved [5]. An alternative solution, both faster and more adaptable, is to utilize flexible resources, such as EVs, to manage the demand dynamically. Altogether, these challenges and opportunities display the urgent need for Sweden to restructure its energy system, transforming it from a conventional grid into a smart grid.

The smart grid is a concept which equips the conventional grid with the appropriate features and functions it needs to enable an efficient transition towards higher sustainability and increasing renewable energy integration [6]. The main goal of the smart grid is to improve the grid's reliability, efficiency, flexibility, sustainability and availability [7]. To fully utilize flexible resources like EVs, the smart grid intelligently integrates communication technologies that enhance the monitoring and controlling of electrical operations [6, 7, 8]. These technologies enable functionalities such as bidirectional power flows, enhanced automation, more efficient Demand Response (DR) mechanisms, and deeper Renewable Energy Source (RES) integration [9]. However, to ensure that these communication technologies function efficiently, standardization is essential. Using a variety of communication protocols can increase complexity, create issues with interoperability, and impair the integration of flexible resources into the power grid [10]. Thus, a clear incentive is to adopt a single, standardized communication protocol in Sweden to facilitate the scalable and effective integration of EVs into the power system.

1.1 Motivation

This project is driven by the need to accelerate Sweden's transition to a sustainable, flexible and reliable power system. As the electricity demand is estimated to more than double until 2045, due to industrial electrification and the increasing adoption of EVs, the effective integration of flexible energy resources becomes a top priority.

The adoption of EVs represents a rapidly growing and relatively unused source of flexibility. When managed correctly, EVs can contribute to grid stability by reducing load during peak hours, participate in various DR programs, and utilize the technology Vehicle-to-Grid (V2G). However, to efficiently access this potential, secure, standardized, and efficient communication protocols are essential. Despite this, Sweden currently lacks a widely implemented DR communication protocol for EV charging and flexibility integration.

A particularly interesting opportunity lies in enabling the direct communication between Distribution System Operators (DSO) and Electric Vehicle Supply Equipment (EVSE). Such an interaction would allow DSOs to access a larger pool of flexible resources, aiding the power grid's ability to balance supply and demand more efficiently. Thus, this project is motivated by the possibility of supporting the transition that Sweden is currently undergoing. By evaluating existing communication protocols and demonstrating a practical implementation, this project can provide insights that can guide the deployment of a scalable, secure, and interoperable solution for EV integration in Sweden's energy system.

1.2 Goals and purpose

The aim of this project is to investigate the communication protocols that facilitate direct interaction between DSOs and residential EVSE, or indirectly through third parties, such as aggregators, for DR purposes. The project also aims to implement and test the communication protocol best suited for adoption in Sweden.

To do so, the scope of the project is divided into three main focus areas: literature review, development, and testing and implementation. The literature review examines current DR communication systems and standards. It also includes a comparison of these standards to identify and choose the most suitable protocol for adoption in Sweden, based on criteria such as scalability, reliability, interoperability and ease of implementation. The development area aims to design a scalable and compliant communication system tailored for the Swedish market, which enables DSOs to interact with the EVSE, based on the identified protocols and technologies in the literature review. The testing and implementation phase focuses on setting up a reliable testing environment and test the chosen communication system to evaluate performance and compliance with DSO requirements.

2 Background

The following sections provide the necessary background for this project. Section 2.1 introduces Demand Response (DR) programs related to EV charging, explaining their main functions and mechanisms. Section 2.2 describes the Vehicle-to-Grid (V2G) technology and its relevance to the energy system. Following this, Section 2.3 identifies the main stakeholders involved in the charging infrastructure ecosystem, describing their roles and responsibilities. Section 2.4 reviews a selection of current communication standards, explaining how they function and is applied. Lastly, Section 2.5 highlights a selection of previous studies related to the work conducted in this project.

2.1 Demand Response Programs

DR programs play a key role in the electricity system, with the ability to maintain and restore balance between supply and demand, cut costs and integrate Distributed Energy Resources (DERs) [11]. DR is defined by the United States (U.S.) Department of Energy as "changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized" [12, 13]. Thus, DR is a solution which is providing several benefits for grid stability, by incentivizing users to change their electricity demand profiles. This is mainly accomplished by two mechanisms; incentive-based programs or price-based programs, which in turn include many different DR programs that benefit the grid [12, 14, 15]. In the following sections a number of relevant DR programs which utilizes EVs are presented.

2.1.1 Peak Shaving

Peak shaving, also called load shedding, is a strategy which serves to stabilize the electricity grid during peak hours. Short-term high electricity loads from large commercial or industrial consumers stresses the grid, as well as drives up the electricity prices during these peak periods [16]. To handle this, peak shaving can alter the load demand by discharging local resources, such as bidirectional EVs and stationary batteries, and utilize local Renewable Energy Resources (RESs), to minimize the electricity supplied from the grid, thus reducing the peak [17]. Apart from being beneficial to grid operators, the end-users also benefit from peak shaving, since the electricity is more expensive during peak hours [18].

It is worth to noting the challenges of peak shaving, which include sizing of energy equipment and response time. The supply-demand balance must be maintained by the grid operators to ensure grid stability, hence a quick response time is vital. It is also important to take into account the overall costs of energy equipment sizing to avoid oversizing [19].

2.1.2 Load Shifting

Load shifting is similar to peak shaving, but instead refers to smoothen the load curve by shifting the load demand to off-peak periods, when both prices and grid demand are lower [20, 21]. To be suitable for load shifting, the resource should be flexible and manageable. By utilizing smart charging, an EV can easily delay the charging and regulate the charging power, thus making EVs applicable for load shifting [22]. Similar to peak shaving, this strategy provides benefits for both the grid operator by flattening the load curve, and the end-users by reducing electricity costs [21].

Challenges associated with load shifting include end-users' reluctance to shift their activities from peak to off-peak periods, due to insufficiently reduced electricity prices or negatively affected comfort levels. A reliable information and communication technology infrastructure is also essential to implement load shifting successfully. Thus, the increased complexity of the overall system operation also creates a barrier [21].

2.1.3 Frequency Regulation

Maintaining a stable frequency is one of the cornerstones in the electricity grid. If there is an imbalance in supply and demand, a frequency deviation can occur. To counteract these deviations, frequency regulation can be applied. Frequency regulation is a strategy designed to stabilize the electricity grid frequency, by utilizing frequency regulating reserves to balance supply and demand [23]. There are several frequency regulation reserves, each with different purposes and activation requirements, depending on the magnitude and duration of the imbalance [24].

To operate as a resource in a frequency regulation reserve, certain requirements must be met. Key resource specifications include a fast response time and meeting the minimal capacity threshold for frequency market entry. A single EV achieves the former of these requirements but not the latter. However, if EVs are being aggregated, they fulfill both criteria [25]. Furthermore, a study by [20] demonstrates that EVs can participate as a resource in multiple frequency regulation reserves, responding to both upward and downward frequency deviations.

2.1.4 Congestion Mitigation

Grid congestion is one of the key issues regarding secure and reliable grid operations, and can occur in both the transmission and the distribution grid. Congestion arises when the grid has insufficient capacity to meet the demand of the electricity customers, creating overloads in the grid [26]. In the distribution and low-voltage grids, the widespread adoption of EVs places significant stress on the system, induced by high simultaneity of EV charging events [27]. Thus, DSOs could experience grid congestion and power quality problems more often due to the adoption of EVs. However, EVs can mitigate congestion by shifting the charging demand to off-peak hours utilizing smart charging, or inject electricity to the grid using the V2G technology [28, 29]. Furthermore, mitigating grid congestion using smart charging is more cost-effective compared to reinforcing the grid [28].

Utilizing EVs as a resource for congestion mitigation poses multiple barriers. First of all, it requires secure and verifiable communication between grid operators and many different decentralized actors. The implementation of these complex communication frameworks could lead to high costs. Another barrier is that EV users may be hesitant to allow their EV battery to be used for smart charging or V2G. Furthermore, grid operators must exercise caution in their EV smart charging algorithms to ensure that EV users do not face insufficient battery levels at departure, which is a risk due to the low forecasting accuracy today [28].

2.2 Vehicle-to-Grid

Vehicle-to-Grid is the most well-known application of Vehicle to Everything (V2X), and refers to utilizing the batteries in EVs to interact, support and provide value to the electricity grid in the form of one or multiple energy services, e.g. ancillary services [30]. The technology has bi-directional charge/discharge capabilities, supports smart charging, and engage in sophisticated communication, meaning that the EVs can receive signals and respond to real-time grid conditions by either varying charge or discharge power [30, 31, 20]. V2G can therefore be regarded as enabling the EV to act as an energy resource or energy storage device [32].

It is important to note that the V2G technology has some limiting elements. Due to the lack of large-scale commercial application of the technology, dedicated grid integration standards have not been developed [32]. The techno-economic aspect is also limiting, as the initial investments and cost for e.g. bidirectional chargers are high. Sociotechnical aspects are also in need of further investigation, including e.g. the motivation of EV owners utilizing V2G and their view on sharing data in charge/discharge events [31, 20]. Additionally, there is a minimum capacity that is required to be able to participate in energy markets, thus requiring an aggregator, see section 2.3.3, to coordinate a multitude of individual vehicles [30].

2.3 Involved Stakeholders

There are several stakeholders involved in the charging infrastructure ecosystem, each taking on different roles, communicating with one another, and exchanging energy and payments through the use of various protocols and interfaces. In practice, these roles can sometimes overlap; for example, a DSO may also operate its own charging stations [33]. The following sections present the key stakeholders in the charging infrastructure. It is worth noting that while other relevant parties may exist, they fall outside the scope of this thesis and are therefore not discussed.

2.3.1 Distribution System Operators

In Sweden, the DSOs are the owners of the distribution grid, which includes the regional- and local grid. The local grid is connected to the regional grid, which in turn is connected to the transmission grid [34]. There are about 170 different DSOs in Sweden, both smaller local DSOs and larger regional DSOs [35]. Regardless of size however, they have similar responsibilities, which are maintaining electricity quality and ensuring security of supply to their customers, connecting new customers to their grids, measuring power consumption and generation, provide flexibility services, as well as investing in and maintaining the infrastructure of the electricity distribution system. Thus, the DSOs have a vital role in the successful integration of DERs into the electricity system. With the current rapid expansion of DERs, this may force DSOs to take on a new, more active role as system operators [36]. Some of the largest DSOs in Sweden today are Vattenfall, E.ON, Ellevio and Göteborg Energi [37].

2.3.2 E-mobility Service Provider

E-mobility service providers (e-MSPs) are companies that offer EV charging roaming platforms, allowing EV drivers to seamlessly access various charging networks without requiring separate memberships. As the name suggests, e-MSPs fosters the growth and convenience of e-mobility, and play a major role in the functioning of EV charging ecosystems, providing smooth interaction between EVs and various charging networks, and an interconnected charging landscape [38]. In comparison with Charge Point Operators (CPOs), see section 2.3.4, the e-MSPs acts as the link between EV drivers and the charging infrastructure provided by the CPOs, thus creating a symbiotic relationship between the user-oriented services and the infrastructure backbone

of EV charging [38, 39]. The largest e-MSPs in Sweden are Fortum Charge & Drive, InCharge by Vattenfall, Clever, and Bee Charging Solutions [40].

2.3.3 Aggregator

Aggregation is defined by the Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU as "a function performed by a natural or legal person who combines multiple customer loads or generated electricity for sale, purchase or auction in any electricity market" [41]. A stakeholder engaged in aggregation, i.e. an aggregator, can therefore be seen as a flexibility provider, and may interact in different ways with parties both up- and downstream in the electricity market, such as DSOs upstream and prosumers downstream [42]. The main roles of the aggregator is to coordinate and aggregate the flexibility provided by the prosumers, offering the flexibility in markets, and arranging and maintaining contracts with prosumers [43]. Aggregators are recognized as third-party actors [36], who combine multiple smaller resources into one larger entity, enabling more extensive grid applications [31]. In Sweden, some major aggregators are Tibber [44] and CheckWatt [45].

2.3.4 Charging Point Operator

The CPO is a stakeholder which is responsible for the physical infrastructure of EV charging, including construction, installation and maintenance of the Electric Vehicle Supply Equipment (EVSE) or Charging Points (CPs), thereby forming the backbone of the EV charging industry [38, 39, 46]. CPOs have a rather flexible role, and are often collaborating with multiple EV charging stations vendors, providing the opportunity to control one or more public or private CPs from different manufacturers [47, 48]. Furthermore, CPOs can also own and operate their own CPs, or extend their services to third-parties, e.g., aggregators, see 2.3.3 [48]. In Sweden, the three largest CPOs by market share of the total CP volume are Vattenfall, Mer and Tesla [49].

Important to note is that the CPO relies on the important role of the Supply Equipment Communication Controller (SECC) within the EVSE. The SECC is essential for enabling communication both between the EVSE and the Electric Vehicle Communication Controller (EVCC) within the EV, and between the EVSE and the grid. The SECC gathers information on energy and tariffs from the grid, and communicates with the EVCC within the EV, supplying information about available charging capacity and pricing. Thus, to ensure efficient management of EV charging and energy consumption, the SECC must be acknowledged [50].

2.3.5 Electric Vehicle Owners

EV-owners are key stakeholders, as they own the EVs whose batteries are utilized as a resource through charging and discharging, and whose EVCC ensures safe and compliant communication with the SECC within the EVSE. To effectively implement EVs in the electricity grid, the interests of the EV owners should be considered. The study conducted in [51] proposes two important factors that needs to be fulfilled; the EV owners' charging demand and the maximization of their profit. The authors of the study in [52] agrees with this and mentions that the charging cost, traveling requirements and battery degradation also needs to be acknowledged. The paper also suggests that an optimized relationship between EV owners and EV aggregators could maximize economic benefits for both stakeholders. Furthermore, [53] proposes that such a relationship also could improve EV owner participation.

2.4 Existing Communication Standards

There are a wide range of communication standards existing today, each designed for specific applications and functioning differently. The following sections review a selection of standards relevant to the EV domain, chosen for their applicability in facilitating communication between key stakeholders. For each standard, its background, potential users, and way of operation are described.

2.4.1 OpenADR

OpenADR (Open Automated Demand Response) is an open, secure, and widely adopted Smart Grid communication standard used to manage DERs and automate demand response in the energy sector [54, 55, 56]. The protocol is developed in the U.S. and has since 2010 being maintained by the OpenADR Alliance, which is also U.S.-based [54, 57]. The OpenADR Alliance has members all over the world, meaning adoption of the protocol is not only occurring in many parts of North America close to development, but also around the world, including Europe, China, Australia, Japan and South Korea [55, 58].

OpenADR is a back-end protocol which enables two-way communication between various stakeholders in the power grid, such as service providers including DSOs, Aggregators, and CPOs, as well as energy users such as EVSE [58, 59]. To do so, OpenADR extends HyperText Transfer Protocol (HTTP), utilizes Extensible Messaging and Presence Protocol (XMPP) as a transport protocol, and uses public-key cryptography for security [58, 60]. XMPP is an application profile of the Extensible Markup Language (XML), which enables the exchange of extensible and structured data between any two or more network entities, in real time [61]. This provides a fundamental feature of XMPP, it facilitates the possibility to interconnect different services through appropriate gateways on the Internet. This feature is ideal for push and fast DR applications while also working for pull, making OpenADR compatible in both push and pull. Push means that a push client waits for data sent by the connected server; pull means that a pull client connects and requests data from the connected server [58].

These push-pull actions create OpenADR's communication process, which always occurs between a Virtual Top Node (VTN) and one or more Virtual End Nodes (VENs). The VTN is a central server component within OpenADR, and is operated on the utility or grid operators side. The VTN serves as a hub for managing interactions with DERs, such as DR events. The VEN on the other hand is a central entity which resides on the customer or resource side in the OpenADR framework. It interacts directly with the VTN, and acts as the endpoint which receives and responds to signals regarding DR or DER management [54]. On the VTN side, utilization of both HTTP and XMPP is mandatory, while either one is applicable on the VEN side [58]. Figure 1 shows an example of the interaction between the VTN and VEN with OpenADR implementation.

OpenADR is available in different versions, OpenADR 2.0a, OpenADR 2.0b and the latest release OpenADR 3.0. The OpenADR 2.0 Profile Specification has international support, since it was approved by the International Electrotechnical Commission (IEC) as IEC 62746-10-1 in 2018 [55]. The new release of the OpenADR 3.0 standard is not intended to replace the OpenADR 2.0a/b profiles, but instead to provide an additional and simplified way to add OpenADR functionalities in current, as well as in new and different scenarios [56]. Many of the general functions are the same between the two versions, where the main difference between the versions 2.0 and 3.0 is the utilization of JavaScript Object Notation (JSON) in 3.0. Another key difference is the removal of the Business Logic (BL), making OpenADR 3.0 more adoptable. The BL in 2.0 was defined by a clear mandatory message exchange pattern. In 3.0, the architecture is

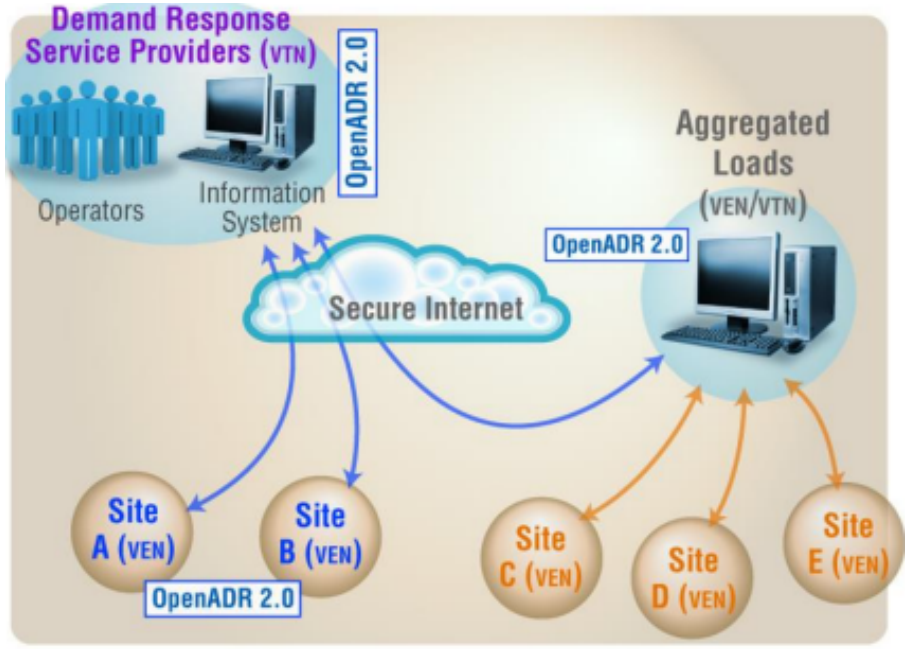

Figure 1: Example of the interaction between VTN and VEN with OpenADR implementation. The Demand Response Service Providers (VTN) represents the DSO and the different sites (VENs) represent multiple EVSE [62].

a REpresentational State Transfer Application Programming Interface (RESTful API), leaving the BL entirely to the application layer. This simplifies the implementation of programs, e.g., price communication for dynamic rates, enabling OpenADR 3.0 to easily be implemented in all types of systems and devices [63].

Worth mentioning is the OpenADR certification test tool, which provides the possibility to certify products such as devices and software, ensuring compliance and interoperability between vendors [54]. The certification process in OpenADR 3.0 will be simpler and more cost effective [63].

Regarding interoperability, OpenADR offers functionality with the communication protocol OCPP [54, 64]. OCPP is discussed in detail in section 2.4.3. The method behind this cooperation is that an OCPP central server aggregates a number of participating charging stations. The central OCPP server then registers as an OpenADR VEN, to an existing OpenADR VTN, turning all the EVs connected to the charging stations into DR assets [65, 66]. Further, [57] and [67] explains that OpenADR is compatible with smart charging. The author in [68] agrees with this, and also establishes that OpenADR is functional with the technology V2G.

2.4.2 IEEE 2030.5-2018

IEEE 2030.5 is titled "Standard for Smart Energy Profile Application Protocol" [69] and is a standard developed by several different stakeholders in the energy supply ecosystem [70]. It formally became an IEEE standard in 2013 and after being revised in 2018, it also incorporates IEEE 1547-2018 functionalities within the standard [71, 72]. The standard defines an application layer, which provides functions in the Internet and transport layers. These functions enable the utility to manage the end-users energy environment, including DR management [56, 69, 71, 72].

IEEE 2030.5 is a broad protocol with a focus on the communication between the utility, e.g. DSOs and end-users. There are many possibility for end-users, including an EVSE, a Home Energy Management System (HEMS), a home area network or an aggregator [57, 71]. To enable the communication between these participants, IEEE 2030.5 utilizes the REST architecture, the HTTP operations GET, PUT, POST, and DELETE, and the XML schema [60, 70, 71, 72]. IEEE 2030.5 also adopts elements from many existing standards, for example security measures from IEEE 1547-2018 [72] and the logical node classes for DER components from IEC 61850-7-420 to improve interoperability [57, 69]. IEC 61850 is described in more detail in 2.4.6. Furthermore, IEEE 2030.5 is compatible with smart charging [67].

2.4.3 OCPP

OCPP (Open Charge Point Protocol) is a back-end open source communication standard [59] developed in 2009 by the Netherlands-based foundation ElaadNL, and is now maintained by the Open Charge Alliance. OCPP is designed to offer a uniform solution for communication between an EVSE and a CPO or a central management system [67, 73, 74], regardless of the vendor of the EVSE [56, 57, 59, 65, 66]. The standard is the de facto open protocol for communication between EVSE and CPO or a central management system [66, 74] and is implemented in many parts of Europe and parts of the U.S. [57]. OCPP is IP-based [66] and utilizes WebSocket [75] and HyperText Transfer Protocol Secure (HTTPS) to communicate between different devices [60].

OCPP is available in different versions, OCPP 2.1, OCPP 2.0.1 and OCPP 1.6, where the latest version is OCPP 2.1. OCPP 2.1 offers improved and added functionalities compared to the former version OCPP 2.0.1, which includes support for ISO 15118-20 with bi-directional power transfer, improved smart charging, and enablement of bidirectional energy flows, allowing compatibility with V2X [74, 76]. ISO 15118-20 is discussed further in section 2.4.5. It is worth mentioning that OCPP 2.1 still is compatible with the former version OCPP 2.0.1, but not with OCPP 1.6. However, both older versions, 2.0.1 and 1.6, are still supported and available for implementation [76]. In addition, OCPP is as mentioned also compatible with OpenADR, see section 2.4.1.

To further promote interoperability and flexibility, the Open Charge Alliance offers a certification program [65, 73]. An OCPP certificate assures and validates OCPP compliance for both vendors and buyers of OCPP implementations. Both the older versions, OCPP 2.0.1 and OCPP 1.6 offers the opportunity to certify an OCPP implementation [77, 78].

2.4.4 EEBus

EEBus is an open-source communication protocol [79, 80] which was founded in Germany in 2012 [81]. The EEBus protocol enables extensive home interoperability by allowing energy-enabled devices from different manufacturers and technologies to interact and exchange information, thus building intelligent HEMSs [80, 82]. By doing so, it allows seamless communication between household appliances, EVSE, storage systems, heat pumps, energy producers, and the HEMS, which in turn can respond to external control signals such as DR signals from DSOs [79]. The study conducted in [83] explains further that by utilizing EEBus with another standardized communication protocol which integrates the DSO, e.g. OpenADR, the combination can enable a seamless bi-directional end-to-end communication system.

2.4.5 ISO 15118

ISO 15118 is developed by the International Organization for Standardization (ISO) and IEC, two of the world's leading standards organizations [84, 85]. It is a front-end communication standard which specifies the communication between EVs and EVSE [57, 59, 85, 86]. The standard is titled "Road vehicles - Vehicle to grid communication interface", meaning that it defines V2G communication, and it consists of several parts where each part covers different aspects of the communication between EVs and EVSE [85]. The latest version is called ISO 15118-20, which specifies and enables an IP based communication between the EVCC and the SECC [87]. Further the latest version also provides improvements and support for bidirectional power transfer [56, 87], smart charging, improved security, Plug and Charge (PnC), and both AC and DC charging [57, 88, 89].

ISO 15118 utilizes transport layer security, XML and V2G-TP to exchange data between devices [60]. It is worth mentioning the high complexity of the protocol, due to cybersecurity constraints [88], the usage of public key infrastructure and certification exchange [60]. However, [84] and [89] imply that the standard has high scalability and potential for future growth, despite its complexity.

2.4.6 IEC 61850

IEC 61850 is an international communication standard developed by a large group of experts from utilities, consultancy companies and manufacturers. First published in 2003, the standard is continuously evolving, with updates and extensions being released annually to address emerging technologies [74]. While it was originally designed for automation and control within electrical substations, its scope has extended to include the integration and control of DERs, including EVs and energy storage systems [90]. The main objective of IEC 61850 is to address interoperability challenges when devices from different manufacturers operates within the same system [91]. To overcome this, the protocol provides a uniform way of communication between devices from different manufacturers, in a reliable and vendor-neutral way [74].

IEC 61850 is not a single protocol, but rather a family of standards, each covering specific components and applications. Among its many extensions, IEC 61850-90-8 and IEC 61850-7-420 are two important members relevant in the context of EV integration. The first extension, IEC 61850-90-8, defines object models for EVs and scenarios involving reverse power flow, enabling V2G applications [92, 93]. The second, IEC 61850-7-420, provide information models for DERs, including EVs and their charging systems [90].

One of the strengths of IEC 61850 is its support for real-time communication, which is essential for dynamic applications such as smart charging. The protocol includes several communication services, including Generic Object-Oriented Substation Event, Sampled Values, and the Manufacuring Message Specification, allowing for structured data exchange between devices and Supervisory Control and Data Acquisition systems [90]. Furthermore, the protocol is designed to be interoperable with other communication standards, such as OCPP and ISO 15118, allowing EVSE managements and V2G applications [90, 74]. It also facilitates the communication between EVSE and DSOs or CPOs [57].

2.5 Previous Studies

In the study conducted by [33] the most widely used communication protocols, physical connectors and key stakeholders involved in private and public EV charging are presented. The communication protocols in the study are related to both vehicle-charging station communication and other communication paths, including ISO 15118, OCPP and OpenADR, among others. Regarding the physical connectors, the study introduces those found in the U.S. and in Europe. The involved parties that are presented are those that organize the charging market, such as grid operators, aggregators, e-MSPs, CPOs, and customers. The study further highlights some of the most relevant trends that are occurring in the charging industry today.

The study written by [56] provides an analysis in which the communication and data science for the success of V2G is presented. The study offers a comprehensive review of V2G systems, focusing on the role of communication, and delves into the communication technologies and standards in V2G systems, highlighting the key requirements for achieving efficient and reliable communication between EVs and the different actors involved. The study also discusses the challenges of communication within the V2G technology, both in its current state and in future trends. The study's key findings are that V2G requires interdisciplinary collaboration and stakeholder engagement to overcome specific challenges, where some of these challenges could to be solved through communication technologies.

The study conducted by [57] presents a discussion of a selection of EV related communication protocols, and provides the reader with a comparison based on the functionality supported by these protocols. The key functionalities include smart charging, roaming, communication between central system and charge point, as well as communication between EV and charge point. Furthermore, the properties maturity, interoperability, market adoption and openness of the protocols are investigated. The report also offers an analysis about how the protocols can be used in conjunction with each other, to provide information for enabling standard harmonization or enabling grid and system interoperability. The communication protocols analyzed in the report are many, and include OpenADR, IEEE 2030.5, OCPP, IEC 61850 and ISO 15118. To conclude, the study provides summarized tables outlining the key functionalities and properties of the investigated protocols, along with future recommendations to accelerate the adoption of EV related communication protocols.

In [59] a comprehensive analysis of communication protocols for EVs is provided. The study performs a thorough literature review, in which the key technical components of the EV ecosystem and an overview of communication protocols for EVs, both front-end and back-end, are presented in detail. Further, the study discusses the significance of protocols in EV charging, with regards to aspects such as communication and data exchange, interoperability, monitoring, smart charging and plug-and-charge capability. The study also investigate the different protocols adequacy for adoption in the EV business, based on the properties compatibility, maturity, market adoption and openness. The protocols in question are ISO 15118, OCPP, IEC 61850, OpenADR, IEEE 2030.5, among others.

The study in [67] presents an analysis of communication requirements of the EVSE to enable coordinate charge of multimodal EVSE. In the analysis the study investigates the players involved in the charging of EVs, different charging scenarios, and the communication functions and communication interfaces of the EVSE. The study also introduces a project, which includes designing a multimodal EVSE with the ability to improve the sustainability of EV charging and to minimize its impact on the power grid. Further, the study presents a simulation framework to validate the architecture of the multimodal EVSE. The key finding of the study indicate that the impact of EV charging on the distribution grid can be reduced by the proposed multimodal EVSE.

In [75], a prototyping of a multi-protocol communication system to enable interoperability for DR services is proposed. The paper brings light to the challenge of real-time communication of data among various resources, e.g. EVs, flexible residential loads and battery storage systems, with a key focus on interoperability issues for aggregators. To address this challenge, the study presents a testbed consisting of a microgrid model integrated with several controllers which utilizes industry grade communication protocols for DR, such as IEC 61850, OpenADR and OCPP. Further the study tests the developed interoperability solution, and confirms the feasibility of the proposed inter-protocol conversion.

The paper [94] presents a methodology to evaluate the performance of DR protocols for the smart grid, in combination with a DR strategy. The methodology demonstrates how to formalize a household scenario, reuse existing specifications of DR protocols, and strategies for this evaluation. The paper also utilizes a simulation environment where the protocols are being evaluated by observing a set of performance metrics. The results from the simulations are used to enhance the protocol behavior, by tuning the parameters. Furthermore, the paper validates the proposed methodology through a case study using the SEP2 communication protocol, which is incorporated within the IEEE 2030.5 standard.

The document from [95] provides an industry recommendation for IT communication between DSOs and customers with conditional grid connections in Sweden. The document identifies four potential communication protocols for evaluation: OSCP, OpenADR, IEEE 2030.5, and IEC 61850-7-420, and describes the requirements that serves as the basis of the recommendation. Based on these requirements, the document proposes OpenADR as the industry recommendation. Furthermore, a Proof-of-Concept implementation of the proposed OpenADR protocol is described, aiming to provide insights into how the protocol's integration might work in practice.

3 Method

This section presents the methodology used in this project. Section 3.1 presents a comparison of the different communication standards lifted in the background, concluding that OpenADR is the most suitable protocol for adoption in Sweden. Thus, to gather insights regarding OpenADR, several interviews were conducted with both important actors in Sweden and the maintainers of OpenADR, which is presented in Section 3.2. The next section, Section 3.3, describes the system architecture and high-level workflow in OpenADR, which served as a guiding structure in the development process. The following sections provide details about the development of the OpenADR system, beginning with the configuration of the Virtual Top Node (VTN) in Section 3.4. Section 3.5 then describes the development behind the Business Logic (BL), including the various object creations and the design of the Business Logic client's User Interface (UI). Section 3.6 continues with the development of the Virtual End Node (VEN), covering object creations and UI design. Finally, Section 3.7 explains the methodology behind the testing, describing the setup and execution of the implemented test cases.

3.1 Comparison of Communication Standards

After gaining information about different communication protocols in use today, a comparison among these protocols was conducted. First and foremost, a connection map between the various standards and key agents in the power grid was made, to investigate which of the standards enabled the communication between DSOs and Electric Vehicle Supply Equipment (EVSE), see Figure 2. From the figure it is possible to see that there are multiple pathways which facilitate the communication between DSOs and EVSE, whereas the communication protocols OpenADR, IEEE 2030.5, and IEC 61850 all enable direct communication from the DSO to the EVSE. Worth mentioning is that even though OCPP, EEBus, and ISO 15118 do not support the direct communication between the DSO and the EVSE, they still play an important role in mapping the whole system, demonstrating all the available communication pathways, from DSO to EV. However, since these protocols do not promote direct communication, the comparison continued between OpenADR, IEEE 2030.5 and IEC 61850.

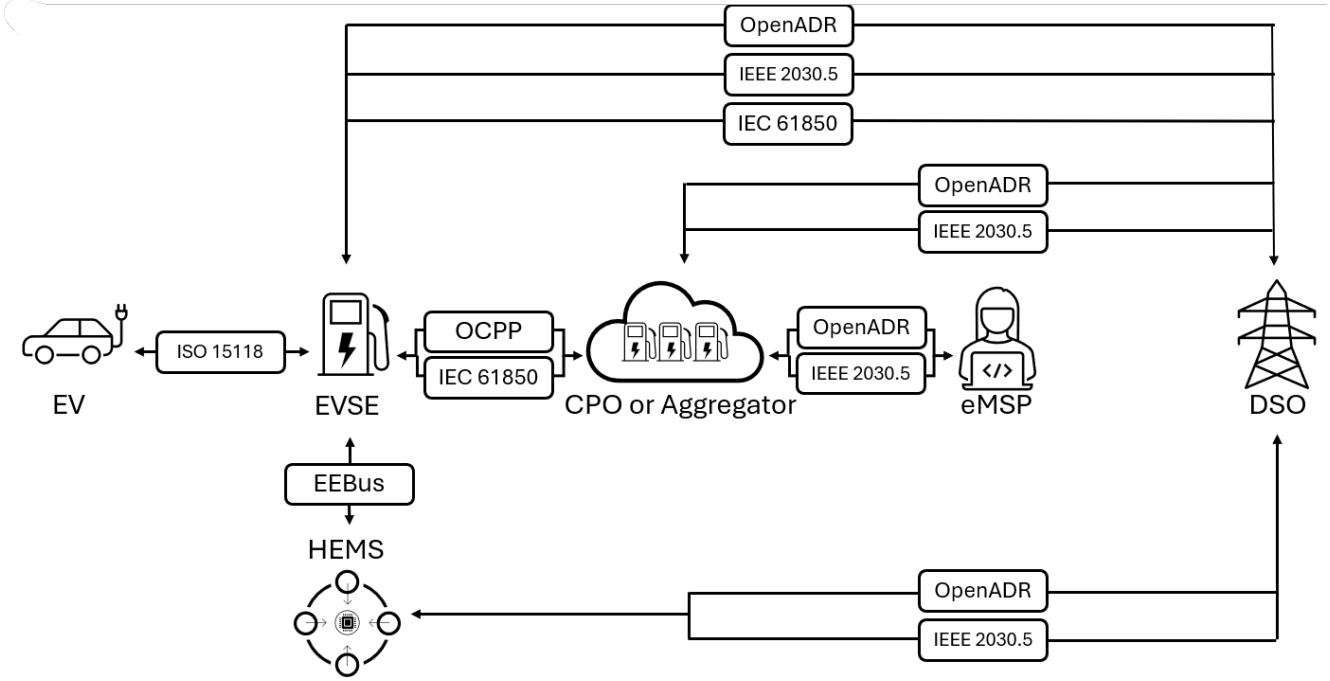

Figure 2: Connection map between key agents and various standards in the power grid.

Further on, as can be seen from figure 2, OpenADR and IEEE 2030.5 have a wide range of uses, and can be implemented to facilitate communication between many different actors, they gain an advantage over IEC 61850. Also, IEC 61850 is more connected to the integration of DER with a high level of safety and interoperability, meaning that it is not mainly used for facilitating communication for DR purposes. Thus, OpenADR and IEEE 2030.5 were chosen for the final comparison.

In the comparison between OpenADR and IEEE 2030.5 a lot of similarities can be found. Both of the protocols can be implemented at many different places in the system, by many different actors, have high scalability and security. However, IEEE 2030.5 has a higher complexity and is less flexible than OpenADR. The primary use case for OpenADR is also Demand Response (DR) programs, while IEEE 2030.5's primary use case is DER and IoT integration. However, IEEE 2030.5 can be implemented for DR, but since DR is the primary focus of OpenADR, the protocol is much more optimized for this particular use case [60]. In addition to this, OpenADR has been recognized in Sweden in a broader way than IEEE 2030.5. E.ON and Ellevio have implemented OpenADR into their systems, and Energiföretagen has proposed OpenADR as an industry standard, in both their industry recommendation and supplement to the industry recommendation [95, 96]. Since the purpose of this project is to investigate the communication for DR purposes in Sweden, OpenADR is seemingly the most promising communication protocol for adoption in Sweden. Therefore, a system which utilizes OpenADR for the direct communication between the DSO and EVSE will be developed.

3.2 Interviews

To gain an overview and deeper insight into OpenADR and its adoption in Sweden, several interviews have been conducted. These interviews followed a similar structure, beginning with participant introductions, followed by a presentation about the project, and concluding with a discussion based on prepared questions. Table 1 provides details on the interviews and their participants, which include representatives from the special interest organization Energiföretagen, the three largest DSOs in Sweden, and the OpenADR protocol maintainer, OpenADR Alliance. Additionally, the following sections outline the primary focus for each interview, and Table 2 presents all the prepared questions and in which interview they were discussed.

Table 1: Participant of the interviews.

Business Entity	Company	Current Role	Duration	Location
Special Interest Organization	Energiföretagen	E-mobility Manager	36 min	Teams
Distribution System Operator (DSO)	E.ON	Product Owner of SWITCH	48 min	Teams
	Ellevio	Solution Architect	63 min	Teams
	Vattenfall Eldistribution AB	Flex Hub Manager	60 min	Teams
Protocol Maintainer	OpenADR Alliance	Managing & Technical Director	69 min	Teams

3.2.1 Energiföretagen

Energiföretagen, also known as Swedenergy, is a non-profit industry and special interest organization representing companies in Sweden that sell, distribute, and store energy. Currently, the organization has nearly 500 members, ranging from state-owned and municipal entities to private companies and associations within the energy sector. Energiföretagen monitors industry development and promotes the interests of its members and the Swedish energy sector as a whole [97].

In October 2023, Energiföretagen published an industry recommendation for conditional grid agreements in Sweden [95]. The purpose of this interview was to examine the adoption of OpenADR within the Swedish energy sector, discuss the industry recommendation, and analyze the real-world case included in the recommendation as a Proof-of-Concept (PoC). Additionally, the representative from Energiföretagen provided an addendum to the industry recommendation, which was also discussed. Table 2 lists all the prepared questions and presents which were discussed in the interview with Energiföretagen.

3.2.2 E.ON

E.ON is one of the three largest DSOs in Sweden [37, 98]. They are owners of roughly a quarter of the electricity grid in Sweden, owning approximately 143 thousand kilometers of the electricity grid. In 2024, they had approximately 1,1 million customers connected to their power grid in Sweden [99]. In addition to Sweden, E.ON has operations and employees in 17 other countries in Europe [100].

E.ON were one of the companies involved in the PoC in the industry recommendation, and provided their already existing flexibility platform called SWITCH into the project. E.ON also implemented a VTN as part of the PoC. Thus, the objective of the interview was to discuss E.ON's part in the PoC, learn about their SWITCH platform and the SWITCH VTN, and investigate E.ON's take on OpenADR as the industry standard in Sweden. Table 2 lists all the prepared questions and presents which were discussed in the interview with E.ON.

3.2.3 Ellevio

Ellevio is one of the three largest DSOs in Sweden [37, 98]. Ellevio owns about 81 thousand kilometers of the electricity grid in Sweden, and had approximately 1 million customers in Sweden connected to their power grid in 2024 [101].

Following the publication of the industry recommendation, Ellevio integrated an OpenADR VTN into their system. Consequently, the interview aimed to discuss Ellevio's VTN implementation, gain insights into OpenADR implementations, and explore their perspective on OpenADR as an industry standard in Sweden. Due to the representative's role within the company, the interview had a more technical focus. Table 2 lists all the prepared questions and presents which were discussed in the interview with Ellevio.

3.2.4 Vattenfall Eldistribution AB

Vattenfall Eldistribution AB is one of the three largest DSOs in Sweden [37, 98]. They own and operate around 139 thousand kilometers of the electricity grid in Sweden, primarily located in the northern and central parts of the country, and had in 2024 approximately 1 million customers connected to their part of the power grid. Vattenfall Eldistribution AB has operations and customers in multiple countries in Europe, with their main markets in Sweden, Germany, Denmark, the United Kingdom, and the Netherlands [102].

Vattenfall Eldistribution AB participated in the PoC outlined in the industry recommendation, implementing a Virtual End Node (VEN) as part of the project. Prior to the interview, their representative also highlighted their involvement in an EU-project. The interview aimed to discuss the industry recommendation and Vattenfall Eldistribution AB's role in the PoC, gather their perspective on OpenADR as the industry standard in Sweden, and explore their participation in the EU-project. Table 2 lists all the prepared questions and presents which were discussed in the interview with Vattenfall Eldistribution AB.

3.2.5 OpenADR Alliance

The OpenADR Alliance, as the maintainer of the OpenADR protocol, plays a key role in its management and development. With the managing and technical director in attendance, extensive expertise about OpenADR was available in the interview. Thus, the primary objective of the interview was to take part in the deep knowledge of the protocol by asking questions and discussing OpenADR functionality, adoption, certification, implementation, and the differences between its various versions. Table 2 lists all the prepared questions and presents which were discussed in the interview with OpenADR Alliance.

Table 2: The prepared questions for the interviews and in which interview they were discussed.

Companies / **Questions**	Energiföretagen	E.ON	Ellevio	Vattenfall Eldistribution AB	OpenADR Alliance
In the industry recommendation it says that IEEE 2030.5 and IEC 61850-7-420 did not fulfill the requirements, which requirements did these fail?	X	X		X	
Is it possible to register a VEN to the SWITCH VTN/ Ellevio VTN for testing?	X	X	X		
Is it possible to find the SWITCH VTN version 2.0b and use it for testing in 2.0b?	X	X			
Have E.ON/Ellevio/Vattenfall looked into the possibility to certify their SWITCH VTN/Ellevio VTN/Vattenfall VEN as an OpenADR certified product?	X	X	X	X	
Is there any other Swedish DSO that are looking into the opportunity to certify their product with an OpenADR certificate?	X	X		X	
Which OpenADR communication case is the best/most promising for DSOs? Direct contact with DERs or through an aggregator?	X	X	X	X	
Do E.ON/Ellevio/Vattenfall have any ongoing projects in Sweden regarding OpenADR?	X	X		X	
According to you, what are the next steps to establish OpenADR as the industry standard in Sweden?	X	X	X	X	X
Do you have any contacts that you think could be good for me to get in contact with?	X	X	X	X	X
Do you have experience working with OpenLEADR?		X	X	X	X
Would you recommend using OpenADR 2.0b or OpenADR 3.0?		X	X	X	
Have Ellevio/Vattenfall implemented OpenADR in any of their systems?			X	X	
Can you briefly explain how Ellevios/Vattenfalls implementation was made?			X	X	
Which use cases does Ellevios OpenADR implementation manage?			X		
Is it possible to register to Ellevios VTN with both OpenADR 2.0b and 3.0?			X		
What would you say are the main differences between OpenADR 2.0b and 3.0?			X	X	

Table 3: The prepared questions for the interviews and in which interview they were discussed (continued).

Companies / **Questions**	Energiföretagen	E.ON	Ellevio	Vattenfall Eldistribution AB	OpenADR Alliance
Do you have experience/knowledge about the other protocols I am investigating?			X		
What is the EU project you were involved in about?				X	
Beyond conditional grid agreements, what are other relevant use cases for OpenADR in EV charging in different markets? What other use cases are being adopted in Europe?					X
What are the key advantages of using OpenADR 3.0 over 2.0b for EV charging systems?					X
Which certified products on OpenADR Alliance's website would you recommend me to investigate?					X
I have seen that many of the certified products have switched status to "no longer maintained". Why is that? Is it because of an update to 3.0?					X
When will OpenADR 3.0 testing and certification be released, and published on the website?					X
Do you know approximately how many OpenADR 3.0 products there are that are looking to be certified?					X
Does the EV charging system (EVSE) need to act as a VEN, or could the aggregator/cloud service be the VEN?					X
What would you say the different steps to building a VEN are?					X
Does the creator of the VEN have to implement the customer logic as well?					X
What would you say are the main differences when implementing a VEN for case 1 compared to case 2?					X
Is it possible to test a OpenADR VEN without assigning it to a VTN?					X
Are there any simulation tools or test servers to test VEN functionality? Does e.g. OpenADR Alliance have a VTN implementation one can use for testing of a VEN?					X
What are the minimal technical requirements for integrating an EVSE as a VEN?					X
What type of products can be OpenADR certified for EV charging applications?					X
Is it possible to get access to the "OpenADR 3.0 VTN Reference Implementation" without being a contributing member?					X
Which actors in Sweden have you been in contact with regarding OpenADR?					X

3.3 OpenADR System Architecture

The system architecture in OpenADR, as mentioned in section 2.4.1, depends on the version of the standard used for development. For this project, the decision was made to adopt OpenADR 3.0, the latest release, based on insights from the conducted interviews. Both DSOs that have implemented OpenADR into their systems are using version 3.0. Additionally, the OpenADR Alliance provided access to the OpenADR 3.0 Virtual Top Node Reference Implementation (VTN RI) - a resource typically restricted to contributing OpenADR members. This access provided the correct tools for development in OpenADR 3.0, as development and testing of a VEN require a functioning VTN.

OpenADR 3.0 is a RESTful API that uses JSON over HTTPS, and defines interactions among three core components: the Business Logic (BL) client, the Virtual Top Node (VTN), and the Virtual End Node (VEN) client, which includes the embedded Customer Logic (CL). Each of these clients has distinct roles and functionalities and communicates using the standard HTTP operations GET, PUT, POST, and DELETE. Figure 3 provides a visualization of how the RESTful architecture is applied in OpenADR.

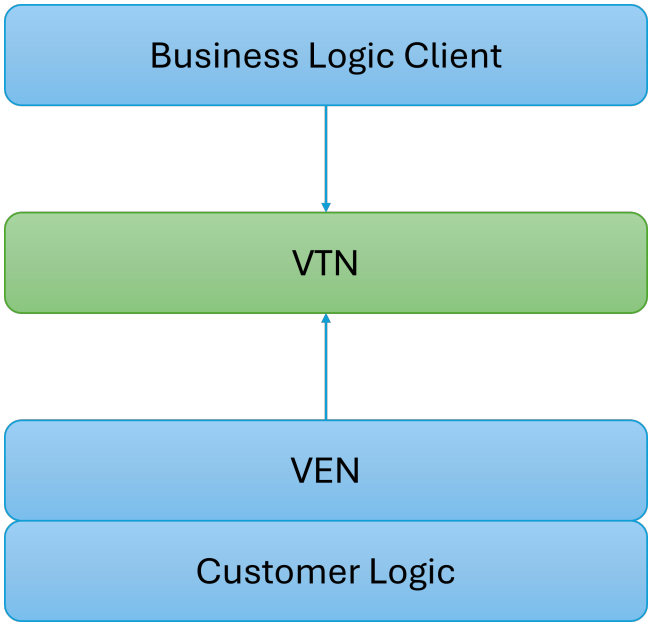

Figure 3: Visualization of the RESTful API application in OpenADR 3.0

The BL client, normally hosted by an energy retailer or DSO, interacts with the VTN and performs operations on behalf of this operator. The VTN functions as a resource server, providing mechanisms for the BL and VEN to exchange data. The VEN is a software client that interacts with the VTN and executes operations as requested by the BL. Embedded within the VEN is the CL, which interacts with local energy resources [96, 103]. This system architecture supports multiple configurations for facilitating the communication between the DSOs and VENs, offering flexibility in how information flows between the core components. However, the official OpenADR 3.0 specification provides a recommended setup in the form of a high level information workflow [104], presented in Figure 4. This recommended high level workflow acts as a guiding structure in the system development in this project.

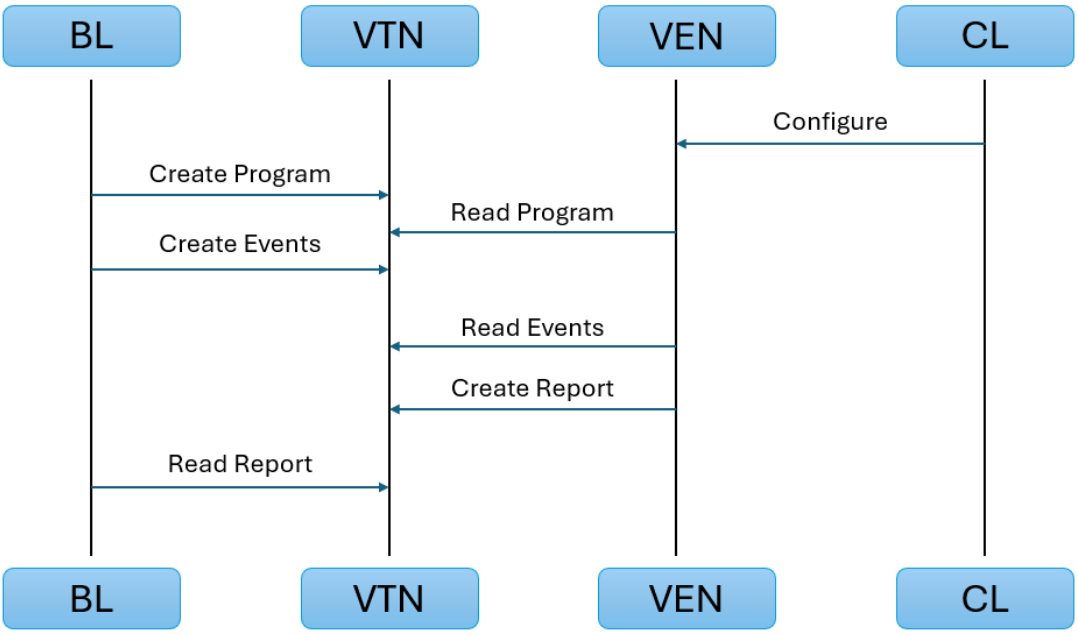

Figure 4: High level workflow in OpenADR 3.0.

Since this project is not conducted in collaboration with any Swedish DSO, the OpenADR 3.0 VTN RI is the only external resource available. As a result the BL client, VEN client, and the CL, are developed and operated locally. Sections 3.4, 3.5, and 3.6 describe the development in detail.

3.4 Configuring the Virtual Top Node

The VTN in this project is the OpenADR 3.0 VTN RI, provided by the OpenADR Alliance. The VTN RI is a fully functioning VTN server, entirely written in Python, using the popular "request" library to enable the JSON method. The OpenADR 3.0 VTN RI is accessed as a repository in GitHub, along with the OpenADR 3 API, accessible via SwaggerHub.

The first implementation step was to clone the repository to get it to run locally with Python. The cloned repository was then opened in the development environment Visual Studio Code (VS Code), followed by some configuration steps to get the VTN RI to run correctly. In the terminal window the commands "`virtualenv venv`" and "`source venv/bin/activate`" were executed to create a virtual environment, and to install the correct packages the command "`pip3 install -r requirements.txt`" was executed. Finally, to start the server, the command "`python -m swagger_server`" was executed, which produced the output shown in Figure 5, confirming that the server was up and running.

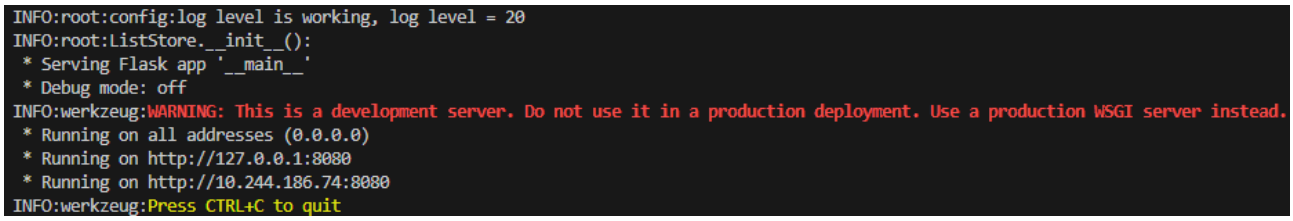

Figure 5: Terminal output confirming that the OpenADR 3.0 VTN RI is up and running.

The last step in configuring the OpenADR 3.0 VTN RI was to make sure that it was possible to interact with it while running. The VTN listens for requests on the port "`http://localhost:8080/openadr3/3.1.0`", and to confirm that the VTN is running and accessible the following command was executed: "`curl -H "Content-type: application/json" -H "Authorization: Bearer bl_token" http://localhost:8080/openadr3/3.1.0/programs`", which should result in the expected output "`[]`". Once the expected output was obtained the VTN was configured correctly and development of the BL and VEN could begin.

An important note is that the OpenADR 3.0 VTN RI uses HTTP, not HTTPS, meaning that it is not necessary to obtain API tokens through authorization. Instead, the VTN RI offers two pre-configured tokens, "`bl_token`" and "`ven_token`", when running test requests. Normally, when working with OpenADR outside of a test environment, authorization is needed for additional security. However, since the VTN RI provides these for a testing environment, implementing authorization was disregarded in this project.

3.5 Developing the Business Logic

The BL client in this project was developed in Python using VS Code. To support comprehension of the BL client and its operations, a UI was implemented with the Python library Streamlit. The BL client is responsible of creating programs and events, as well as reading reports. The created programs and events are sent to the VTN, which either confirms and acknowledges the data if the requests are correctly formatted, or denies them if they are not. The BL client utilizes the pre-configured token `"bl_token"` to authorize its requests to the VTN. The following sections describe the creation of programs, events, and the UI in detail.

3.5.1 Creating a Program

The program is an object created by the BL client that provides program-specific metadata from the VTN to the VEN. While the BL can create multiple programs, one program is generally sufficient for common use cases. Importantly, a program must be created before creating any events, as each event object must be associated with a specific program. The program can have any number of events associated with it.

There are several various parameters included in the program, which is presented in Table 3. The table includes the definitions of each parameter based on the official OpenADR specification [103], along with an indication whether the parameter was used in the development.

Table 3: The parameters included in the creation of a program object, their definition, and whether they were used during development.

Parameter	Definition	Used in development	
		Yes	No
id	VTN provisioned ID of this object instance.	X	
createdDateTime	Creation time for object, e.g. "2023-06-15T12:58:08.000Z".	X	
modificationDateTime	Modification time for object, e.g. "2023-06-16T12:58:08.000Z".	X	
objectType	Used as discriminator. PROGRAM	X	
programName	Name of program with which this event is associated, e.g. "ResTOU".	X	
programLongName	User provided ID, e.g. "Residential Time of Use-A".		X
retailerName	Program defined ID, e.g. "ACME".	X	
retailerLongName	Program defined ID, e.g. "ACME Electric Inc.".	X	
programType	User defined string categorizing the program, e.g. "PRICING_TARIFF".		X
country	Alpha-2 code per ISO 3166-1, e.g. "US".	X	
principalSubdivision	Coding per ISO 3166-2. E.g. state in US, e.g. "CO".		X
timeZoneOffset	An ISO 8601 duration that is to added to all interval.start values.	X	
intervalPeriod	The temporal span of the program, could be years long.	X	
programDescriptions	List of URLs to human and/or machine-readable content, e.g. "mple: www.myCorporation.com/ myProgramDescription".		X
bindingEvents	True if events can be expected to not be modified. [false]		X
localPrice	True if events have been adapted from a grid event. [false]		X
payloadDescriptors	An optional list of objects that provide context to payload types.		X
targets	An optional list of valuesMap objects.	X	

Although many of the parameters listed in Table 3 are optional and can be excluded to create a minimalist program, the choice was made to include many of the non-compulsory parameters to

enhance readability and clarity. Some parameters, such as *id*, *createdDateTime*, *modificationDateTime*, and *objectType*, are automatically assigned by the VTN when creating or modifying a program.

Among the parameters chosen in the development, certain parameters were configured through the UI developed by Streamlit, see Section 3.5.3, while others were given constant values in the code. The parameters *programName* and *retailerName* were set to be user-defined in the UI, while *retailerLongName* was set to "Volvo Cars". The *country* parameter was configured in the UI, allowing the user to choose between the Nordic countries. To calibrate the time to the Swedish timezone, the *timeZoneOffset* parameter was set to "PT1H", in accordance with ISO 8601. The start time and duration of the program, specified by the parameter *intervalPeriod*, was set to also be user-defined via the UI. Finally, the *targets* parameter was set to "RESOURCE_NAME" with the belonging value "EVSE"

It is also worth mentioning the parameters *bindingEvents* and *localPrice*, which were not utilized during development, and thus adopted their default value of [false]. Additionally, if the *payloadDesctriptors* parameter had been implemented in the program object creation, it would have applied to all events associated with that program, erasing the need to specify the *payloadDescriptors* in each individual event. However, in this project the decision was made to define *payloadDescriptors* within event object creation instead.

3.5.2 Creating an Event

The BL client is also responsible of creating the event object. The event object is always associated with a specific program, and it is used to communicate a DR request to a VEN. Thus, the event is one of the most important entities of OpenADR, since it contains the information for when and how the VEN should operate. The various parameters included in the event object, along with their definitions and whether they were used in development or not is presented in Table 4, based on the official OpenADR specification [103].

Table 4: The parameters included in the creation of an event object, their definition, and whether they were used during development.

Parameter	**Definition**	**Used in development**	
		Yes	**No**
id	VTN provisioned ID of this object instance.	X	
createdDateTime	Server provisions timestamp on object creation, e.g. "2023-06-15T12:58:08.000Z".	X	
modificationDateTime	Server provisions timestamp on object modification, e.g. "2023-06-16T12:58:08.000Z".	X	
objectType	Used as discriminator. EVENT	X	
programID	ID attribute of program object this event is associated with.	X	
eventName	User defined string for use in debugging or UI, e.g. "price event 11-18-2022".	X	
priority	Relative priority of event. A lower number is a higher priority.	X	
targets	An array of valuesMap objects.	X	
reportDescriptors	An array of reportDescriptor objects. Used to request reports from VEN.	X	
payloadDescriptors	An array of payloadDescriptor objects.	X	
intervalPeriod	Defines default start and durations of intervals.	X	
intervals	An array of interval objects.	X	

Similar to the program object, not all the parameters are mandatory. However, to improve clarity, all parameters were included in the development. As with the creation of the program, the VTN automatically assign values to the parameters *id*, *createdDateTime*, *modificationDateTime*, and *objectType* at event creation and modification. The *programID* parameter links the event to its associated program, and was set in the code to reference the most recently created program.

In the UI, it was implemented for the user to choose the event type, either "Curtailment Event" or "V2G Capacity Limit Event", which also assign the chosen event type to the *eventName* parameter. The *priority* parameter was set to the constant value of "0", indicating the highest priority. As with the program object, the *targets* parameter was set to "RESOURCE_NAME" with the value "EVSE".

The *reportDescriptors* parameter contains an array of configurable objects. In this project, only the *payLoadType* was chosen to be defined, which was set to either "CURTAILMENT_ACKNOWLEDGEMENT" or "EXPORT_CAPACITY_LIMIT_ACKNOWLEDGEMENT", depending on the event type. All other elements in the *reportDescriptors* parameter were neglected, and thus adopted their default values.

The parameters *payloadDescriptors* and *intervalPeriod* are both connected to the *intervals* parameter. The *intervals* parameter typically contain one or more intervals, each which may contain one or more payload objects, which themselves consists of type and values attributes. The *payloadDescriptors* parameter provides context for these attributes, by defining the *payloadType* and corresponding unit. The decision was made to include one interval and depending on the chosen event type, the payloads in *intervals* were configured as follows:

- Curtailment Event: type = "DISPATCH_SETPOINT", values = user-defined in UI.
- V2G Capacity Limit Event: type = "EXPORT_CAPACITY_LIMIT", values = user-defined in UI.

The *payloadDescriptors* parameter was set to accordingly match the chosen type, and to provide context to the value by setting the unit to either amperes or kilowatts.

The *interval* parameter offers the opportunity to assign a separate *intervalPeriod* to each interval. However, this function is mostly used when each interval has different duration. Since it was chosen to use one interval, this function was neglected. Thus, the *intervalPeriod*, which defines the event's start time and duration, was implemented to be set by the user in the UI.

3.5.3 Building the Business Logic User Interface

To enhance the understanding and streamlining the creation of the program and event objects, a UI was implemented to support the BL, with the Python library Streamlit. The UI allows the user to configure and define certain elements in the program and event object creation. To launch the Streamlit UI, the command "`streamlit run openadr_bl_app.py`" was executed, which starts the BL client as a local web application.

The first step in the UI is to specify the VTN address. This configuration sets the BL client to communicate over the correct port and enables it to send program and event creation requests to the VTN. Figure 6 visualizes how this step is executed in the UI.

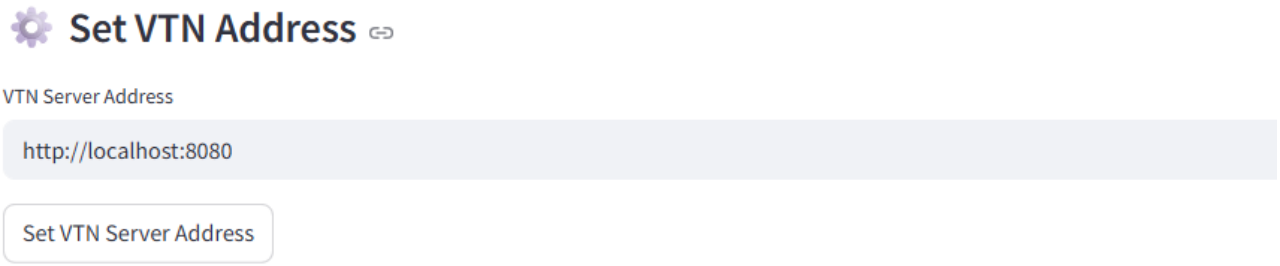

Figure 6: The BL UI which sets the BL client to the correct VTN address.

When the BL client is set to the correct VTN address, it is possible to create a program. As mentioned in Section 3.5.1, some of the parameters were set to constant values in the code, while others may be defined by the user. Figure 7 shows the parameters which are user-defined during the creation of a program in the UI. Once the "Create Program" button is pressed, the BL client sends a POST request to the VTN, containing the predefined constant values alongside the user-defined values. If correctly formatted, the VTN server confirms and acknowledges the create program request.

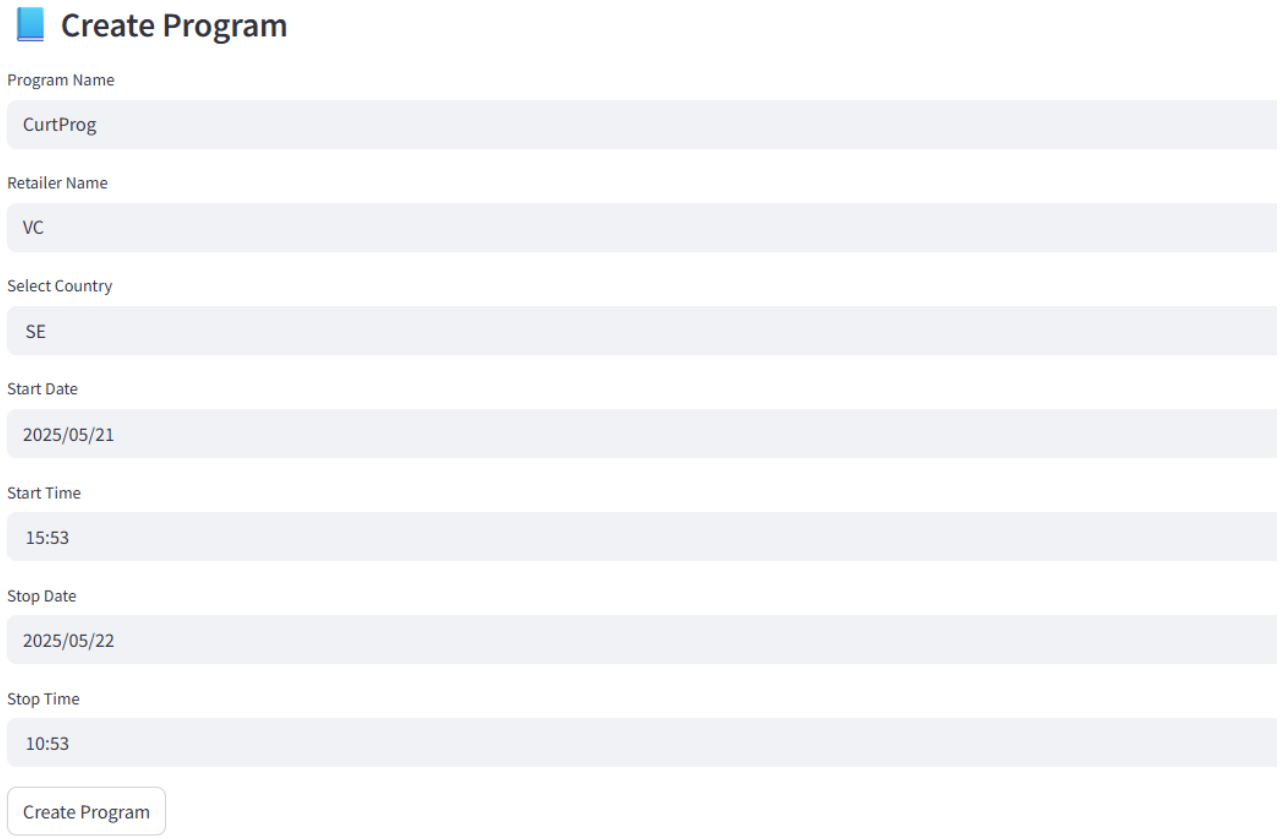

Figure 7: The BL UI which creates the program object.

When a program is created and confirmed by the VTN server, it is possible to create an event. Similar to the creation of the program, some parameters in the event creation are set to constant values in the code, while others are being defined by the user in the UI. Unlike the program though, it is possible to select the event type in the UI, choosing between "Curtailment Event" and "V2G Capacity Limit Event". Depending on event type, certain parameters change to correspond to the chosen type. Figure 8 show how these parameters change depending on event type, and presents the two event types besides each other for easier comparison. Once the "Create Event" button is pressed, the BL client sends an POST request to the VTN, containing the predefined constant values alongside the user-defined values. If correctly formatted, the VTN server confirms and acknowledges the create event request.

Figure 8: The BL UI which creates the event. Both the different event types "Curtailment Event" and "V2G Capacity Limit Event" are presented.

3.6 Developing the Virtual End Node

The VEN client in this project was developed in Python using VS Code. Similar to the BL client, a UI was implemented using the Python library Streamlit to improve the understanding of the VEN client and its operations. The VEN client is responsible for creating VEN objects, defining resources associated with these VEN objects, creating subscriptions, and generating reports. These different elements are sent to the VTN, which either confirms and acknowledges the requests if correctly formatted, or rejects them if they are not. To authorize the requests sent to the VTN, the VEN client utilizes the pre-configured token `"ven_token"`.

It is important to note that although report creation typically is part of the VEN client's responsibility, this functionality was not implemented in this project. This decision was made because all communication was conducted in a local testing environment, eliminating the need to exchange reports between the VEN and BL clients. However, in a production environment, reports are an important component, indicating whether a VEN has received and responded to a certain event. Thus, the report creation process is still described, to support further research and development.

The following sections provide a description of the creation of the VEN object, resources, subscriptions, reports and the VEN client UI.

3.6.1 Creating a Virtual End Node Object

The VEN object represents a client with the VEN role, and is created by the VEN client. While several parameters can be included in the creation of a VEN object, many of them are optional, allowing a basic VEN object implementation. The VEN object can have one or multiple resources associated with it. Table 5 presents the parameters included in the creation of a VEN object, along with their definitions and whether they were used in this project.

Table 5: The parameters included in the creation of an VEN object, their definition, and whether they were used during development.

Parameter	**Definition**	**Used in development**	
		Yes	**No**
id	VTN provisioned ID of this object instance.	X	
createdDateTime	Server provisions timestamp on object creation, e.g. "2023-06-15T12:58:08.000Z".	X	
modificationDateTime	Server provisions timestamp on object modification, e.g. "2023-06-16T12:58:08.000Z".	X	
objectType	Used as discriminator. VEN	X	
venName	String identifier for VEN. VEN may be configured with ID out-of-band.	X	
attributes	A list of valuesMap objects describing attributes.		X
targets	An array of valuesMap objects.		X
resources	A list of resource objects representing end-devices or systems.		X

The parameters *id*, *createdDateTime*, *modificationDateTime*, and *objectType* are automatically assigned by the VTN upon VEN object creation. In this project, the VEN object was purposely kept minimal, with only the *venName* parameter specified in addition to the VTN-assigned parameters. The *venName* parameter was implemented to be user-defined in the UI. The decision to keep the VEN object simple was made because of the possibility to define the optional VEN object parameters *resources*, *attributes*, and *targets* through the creation of a resource object. Since a resource object is directly associated with a VEN object, the possibility

to define these parameters in the resource creation enhances the clarity and flexibility of the development.

3.6.2 Creating a Resource

The resource object is created by the VEN client and represents an energy device or system subject to be controlled by a VEN. The resource is directly associated with a specific VEN object and provides additional information about the VEN object and its connected assets. The various parameters involved in the creation of a resource object are presented in Table 6, along with their definitions and an indication of whether they were utilized during the development.

Table 6: The parameters included in the creation of a resource, their definition, and whether they were used during development.

Parameter	**Definition**	**Used in development**	
		Yes	**No**
id	VTN provisioned ID of this object instance.	X	
createdDateTime	Server provisions timestamp on object creation, e.g. "2023-06-15T12:58:08.000Z".	X	
modificationDateTime	Server provisions timestamp on object modification, e.g. "2023-06-16T12:58:08.000Z".	X	
objectType	Used as discriminator. RESOURCE	X	
resourceName	String identifier for resource. Resource may be configured with ID out-of-band.	X	
venID	VTN provisioned on object creation based on path.	X	
attributes	A list of valuesMap objects describing attributes.	X	
targets	An array of valuesMap objects.	X	

Similar to the creation of the VEN object, the parameters *id*, *createdDateTime*, *modificationDateTime* and *objectType* are automatically assigned by the VTN upon resource creation. Furthermore, the *venID* parameter is also automatically set by the VTN, based on the URL endpoint included in the creation request. In this project, each resource object is upon creation associated with the most recently created VEN object.

Through the UI, the user can select from three different resource types, "EVSE", "Heatpump" or "RESS". Depending on this choice, the *targets* parameter is adaptively assigned to the corresponding resource. However, in this project, only the "EVSE" resource type was used. As a result, the parameter *resourceName* was given the static value of "EVSE", regardless of the resource selection in the UI. Lastly, the *attributes* parameter contains the two fields *type* and *values*. The type was set to the constant "MAX_POWER_CONSUMPTION", while the value was user-defined in the UI.

3.6.3 Creating a Subscription

The subscription object is created by the VEN client with the purpose of receiving notifications related to operations performed on objects. Thus, the subscription object serves as a mechanism that the VEN client can use to register interest in specific object operations and receive real-time updates via so-called "callbacks". In OpenADR this mechanism is known as a webhook, and it allows the VEN client to receive operations from the VTN, originally posted by the BL client. While OpenADR also supports an alternative polling method to receive these operations from the VTN, this project chose to only adopt the subscription method. Therefore, the polling functionality was excluded from the implementation. It is also important to note that the

subscription object is always associated with a specific program. The parameters included in the creation of a subscription object are presented in Table 7 together with their definitions and whether they were utilized during development.

Table 7: The parameters included in the creation of a subscription, their definition, and whether they were used during development.

Parameter	**Definition**	**Used in development**	
		Yes	**No**
id	VTN provisioned ID of this object instance.	X	
createdDateTime	Server provisions timestamp on object creation, e.g. "2023-06-15T12:58:08.000Z".	X	
modificationDateTime	Server provisions timestamp on object modification, e.g. "2023-06-16T12:58:08.000Z".	X	
objectType	Used as discriminator. SUBSCRIPTION	X	
clientName	User generated identifier	X	
programID	ID attribute of program object this subscription is associated with.	X	
objectOperations	List of objects and operations to subscribe to.	X	
objects	List of objects to subscribe to.	X	
operations	List of operations to subscribe to.	X	
callbackUrl	User provided webhook URL.	X	
bearerToken	User provided token	X	

As with the other objects created by the VEN client, the parameters *id*, *createdDateTime*, *modificationDateTime*, and *objectType* are automatically assigned by the VTN upon object creation. The remaining parameters were either defined by the user through the UI or statically defined within the code.

The parameters *programID*, *clientName*, and *callbackUrl* are configured to be user defined in the UI. The *programID* is specifically important, since the subscription object is associated with and must refer to an already created program object. Therefore the *programID* is recommended to be chosen as the given default value "0", which is the value the VTN automatically assigns to the first program created. The *callbackUrl* parameter, which defines the endpoint to which the VTN sends the webhook payload, is also possible to user-define in the UI. However, when launching the UI, a suggested callback URL is provided, which corresponds to an internal server hosted in the VEN client. The parameter *callbackUrl* should therefore be the suggested `http://localhost:8082/callback`, in accordance with the port number defined internally in the VEN client code.

The remaining parameters *objectOperations*, *objects*, *operations*, and *bearerToken*, were statically implemented in the code. The first parameter, *objectOperations*, serves as a container for the other three parameters, and is not directly assigned a value. The *objects* parameter defines which objects the VEN client subscribes to. In this project, the selected objects were "EVENT", "PROGRAM", RESOURCE", and "VEN", meaning that any operation affecting any of these object types, such as creation or updates, will initiate a callback to the VEN client. The *operation* parameter defines which types of object operations the VEN client subscribes to, whereas the selected operations were "CREATE", "UPDATE", and "DELETE". Together, the *objects* and *operations* parameters thus determine the scope of the subscription.

The last parameter, the *bearerToken*, represents a static authorization token used to authenticate the subscription. Normally this token is acquired through dynamic token generation, providing more security compared to a static token. However, since this project is performed in a local testing environment, a static token was deemed sufficient.

3.6.4 Creating a Report

The report is a mechanism which enables the BL client to request reports from the VEN client. The BL client specifies during event creation whether a report should be generated, and which data should be included within it. However, the actual report object creation however, is the responsibility of the VEN client. The report object is directly associated with a specific program and event, enabling the VEN client to communicate a response to the DR requests.

Table 8 presents the parameters involved in the creation of a report object, including their definitions and an indication of whether they were used during development. It is important to note that the report functionality was not implemented in this project, which is why none of the parameters were utilized.

Table 8: The parameters included in the creation of a report, their definition, and whether they were used during development.

Parameter	**Definition**	**Used in development**	
		Yes	**No**
id	VTN provisioned ID of this object instance.		X
createdDateTime	Server provisions timestamp on object creation, e.g. "2023-06-15T12:58:08.000Z".		X
modificationDateTime	Server provisions timestamp on object modification, e.g. "2023-06-16T12:58:08.000Z".		X
objectType	Used as discriminator. REPORT		X
programID	ID attribute of program object this report is associated with.		X
eventID	ID attribute of event object this report is associated with.		X
clientName	String ID of client, may be VEN ID provisioned during program enrollment.		X
reportName	User defined string for use in debugging or UI, e.g. "Battery_usage_04112023".		X
payloadDescriptors	An optional list of objects that provide context to payload types.		X
resources	An array of objects containing report data for a set of resources.		X
resourceName	User generated identifier. A value of AGGREGATED_REPORT indicates an aggregation of more than one resource's data.		X
intervalPeriod	Defines temporal aspects of intervals.		X
intervals	An object defining a temporal window and a list of payloads.		X

The parameters *id*, *createdDateTime*, *modificationDateTime*, and *objectType* are automatically assigned by the VTN upon report creation, as with the other objects created by the VEN client. The *programID* and *eventID* parameters refer to the program and event that the report is associated with. The *clientName* parameter refer to the name of the VEN client, and could for instance be the name of the VEN object or the client name specified in the creation of a subscription. The *reportName* parameter defines the name of the report, and creates clarity about which event the report is associated with.

The *payloadDescriptors* parameter provides context for the payload data, such as type and unit of the payload values. This context is essential for interpreting the contents of the *resources* parameter, which serves as a container for additional report object details. Within *resources*, the three parameters *resourceName*, *intervalPeriod*, and *intervals* are defined. The *resourceName*

parameter specifies the name of the resource being included in the report. The parameter *intervalPeriod* specifies the overall start time and duration of the report object, while the *intervals* parameter specifies the number of intervals and the payload included in these. Each interval contains one or more payloads, which include the payload type and corresponding value, which are given context in the *payloadDescriptors* parameter.

If the report functionality had been implemented in this project, a use case could be for the VEN client to notify the BL client that the DR event was received and executed. In such a use case, the *intervals* parameter could be set to the type "CURTAILMENT_ACKNOWLEDGEMENT" with a value of "EXECUTED", to indicate the action taken by the VEN client.

3.6.5 Enabling Connection with the EVSE

In order for the VEN client to receive events through the subscription mechanism and communicate them to the EVSE, a communication pathway needs to be established between the two. To enable this pathway, the laptop running the VEN client must connect to the EVSE via the appropriate access point, thus forming a local connection. This connection is done through HTTP/HTTPS, and the VEN client utilizes the RESTful APIs to communicate with the EVSE. It is also important to note that the EVSE operates as a separate hardware device, running its own server on a different port than the VEN client. However, as both the EVSE and VEN client are sharing the same IP-address, all communication is performed locally, as intended with the test environment setup.

Once the communication pathway is successfully established, the user is required to log in to the EVSE through the UI by providing the correct credentials. The credentials must correspond to a service-level user, and were provided by Volvo Cars. To streamline the UI design, these credentials are automatically suggested when launching the VEN client. Upon successful login, the VEN client is programmed to automatically send the event data to the EVSE. When receiving the data, the EVSE interprets and responds to the event. Furthermore, once logged in, the user is also able to fetch the EVSE charging current through the UI. See section 3.6.6 for a visual representation.

3.6.6 Building the Virtual End Node User Interface

To enhance understanding and to streamline the operations performed by the VEN client, a UI was implemented using the Python library Streamlit. This UI enables the user to configure and define certain elements of the VEN client in a more accessible way. To launch the Streamlit UI, the command "`streamlit run openadr_bl_app.py`" was executed, which starts the VEN client as a local web application.

As with the BL client interface, the first step in the VEN client UI is to specify the VTN address. By pressing the "Set VTN Server Address" button, as shown in Figure 9, the VEN client is configured to communicate over the appropriate port. This step must be done first thing, since it is essential for enabling the VEN client to send VEN object, resource, and subscription creation requests to the VTN.

Once the VTN address is configured, the user can create a VEN object with resources associated to it. As previously discussed, some parameters are user-defined via the UI, while others are statically defined in the code. For the VEN object, which is intentionally kept minimal, the only definable parameter in the UI is the VEN name.

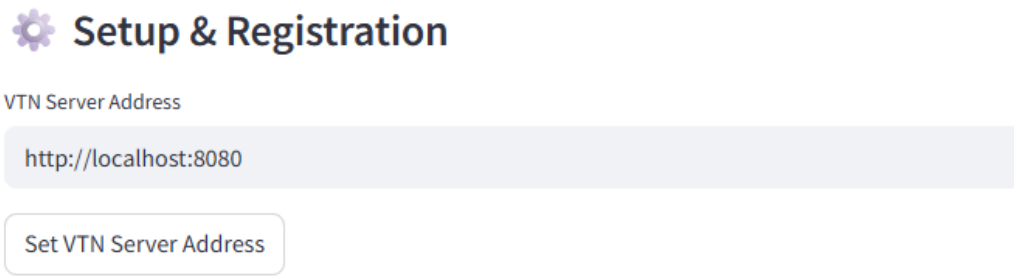

Figure 9: The VEN client UI which sets the VEN client to the correct VTN address.

For the resource creation, the user gets more flexibility. The UI allows the user to choose between three predefined resource types: "EVSE", "Heatpump", and "RESS". However, only EVSE was used in this project. Additionally, the user can choose the "Max Power" value of the implemented resource. Figure 10 illustrates the VEN client UI for creating a VEN object and associated resource. Upon pressing the "Register VEN" or "Add Resources" button, the VEN client sends a POST request to the VTN, containing the predefined constant values alongside the user-defined values. If the request is formatted correctly, the VTN server confirms and acknowledges the request.

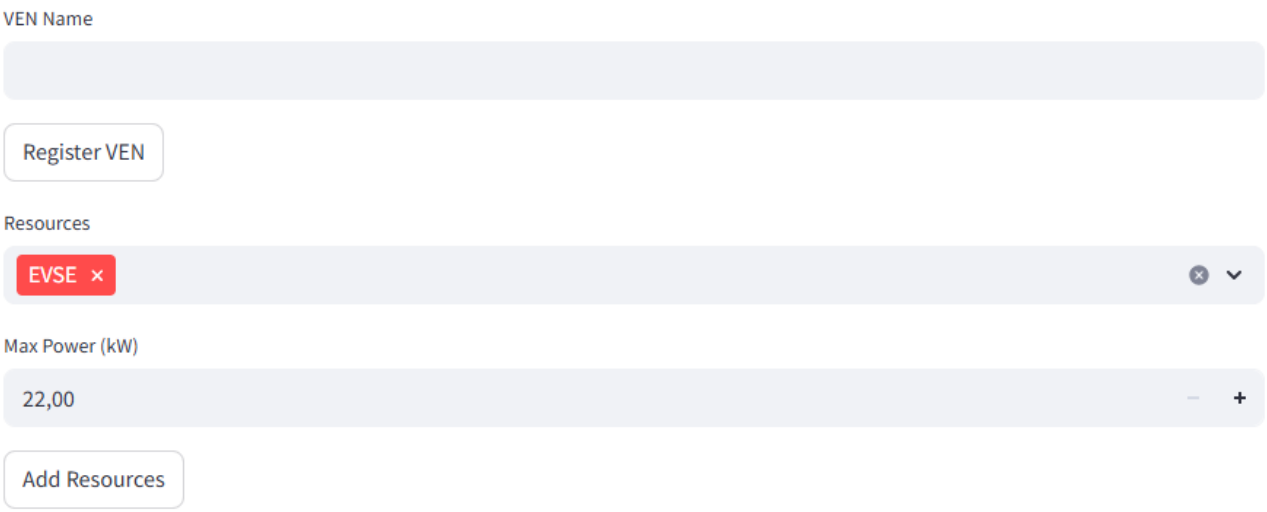

Figure 10: The VEN client UI which creates the VEN and resource objects.

Following the creation of the VEN and resource objects, it is possible to create a subscription. It is important to note that a program object must have been created in advance by the BL client, since the subscription object must be associated with an existing program. Without a preexisting program, the subscription creation will fail. Assuming a program is created in advance, the UI allows the user to define several parameters related to the subscription object, while other parameters are set to constant values in the code. Figure 11 displays the user-defined parameters in the UI when creating a subscription object. Notably, the "Program ID" should be set to "0" to correspond to the default ID automatically assigned the first program created by the VTN. Similarly, the "Callback URL" should be kept to the suggested default value displayed in the UI. Once the "Create Subscription" button is pressed, the VEN client sends a POST request to the VTN, which contains the user-defined values alongside the predefined values. If the request is correctly formatted, the VTN server confirms and acknowledges the subscription creation.

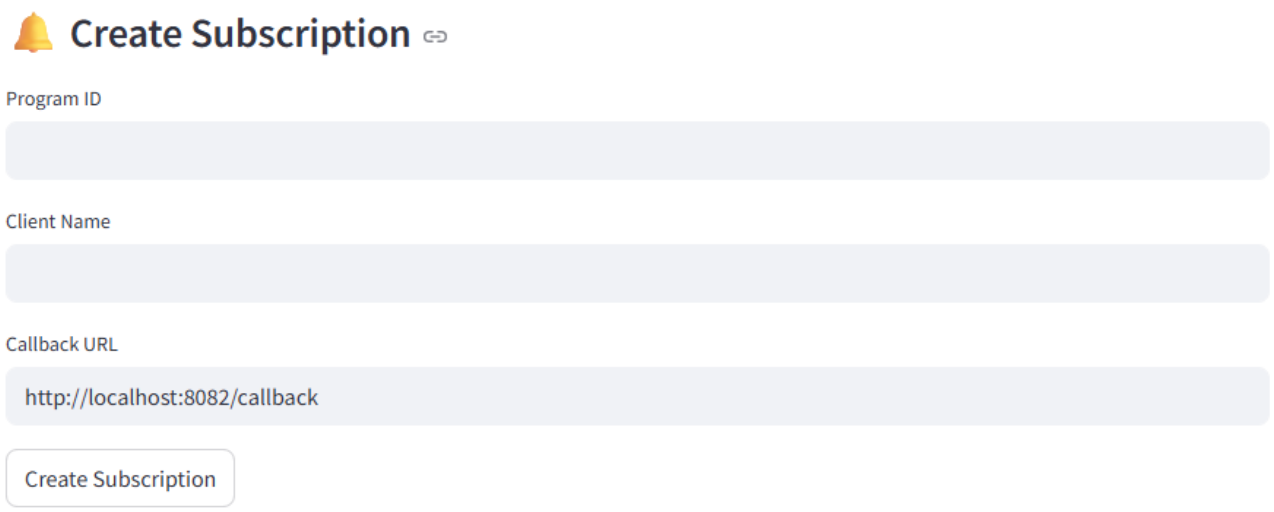

Figure 11: The VEN client UI which creates the subscription object.

The VEN client UI also offers the possibility to verify the connection to the EVSE and to login to it. There are three definable parameters that can be defined in the UI, the "Wallbox URL", "Username", and "Password". However, to ensure successful login, the user should retain the default values provided in the UI, since these default credentials have been verified to work with the EVSE used in this project. Figure 12 presents the VEN client UI used for logging in and verifying the connection with the EVSE. This figure also displays the "Get User Charging Current" option, which enables the user to obtain the current charging value from the EVSE. When pressing the "Verify Wallbox Connection" button, the VEN client attempts to login by sending the default credentials to the EVSE. If the credentials are valid, the UI confirms the login and the connection have been established successfully.

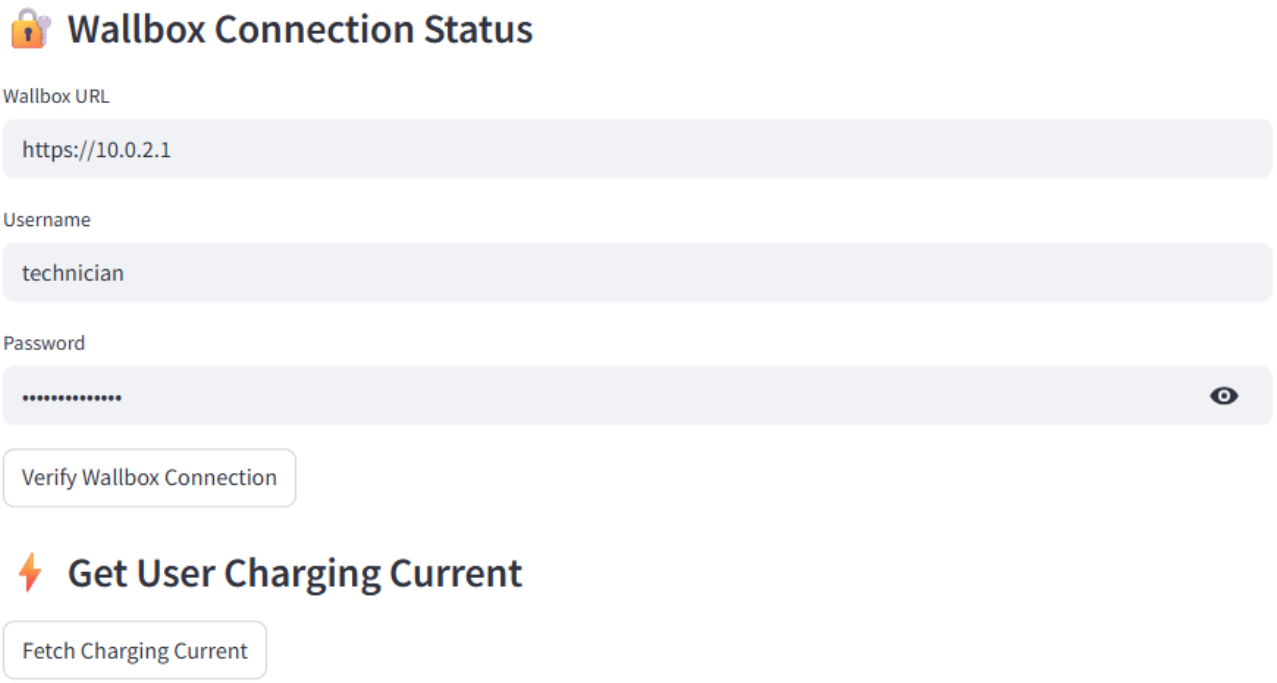

Figure 12: The VEN client UI which verifies the EVSE connection, performs the login authentication, and obtains the current charging value.

Additionally, the VEN client UI includes an integrated "Event Viewer". To get the Event Viewer to function, the VEN client must have created a subscription. When an event is created by the BL client, the VEN client retrieves the data through the active subscription and appends it to an internal list of events. This list is then made visible in the Event Viewer by pressing the "Refresh Events" button in the VEN client UI. The Event Viewer is implemented to improve transparency and provide a clear overview of both created and active events. Figure 13 illustrates the Event Viewer in two scenarios, one with no active events, and another with an ongoing curtailment event.

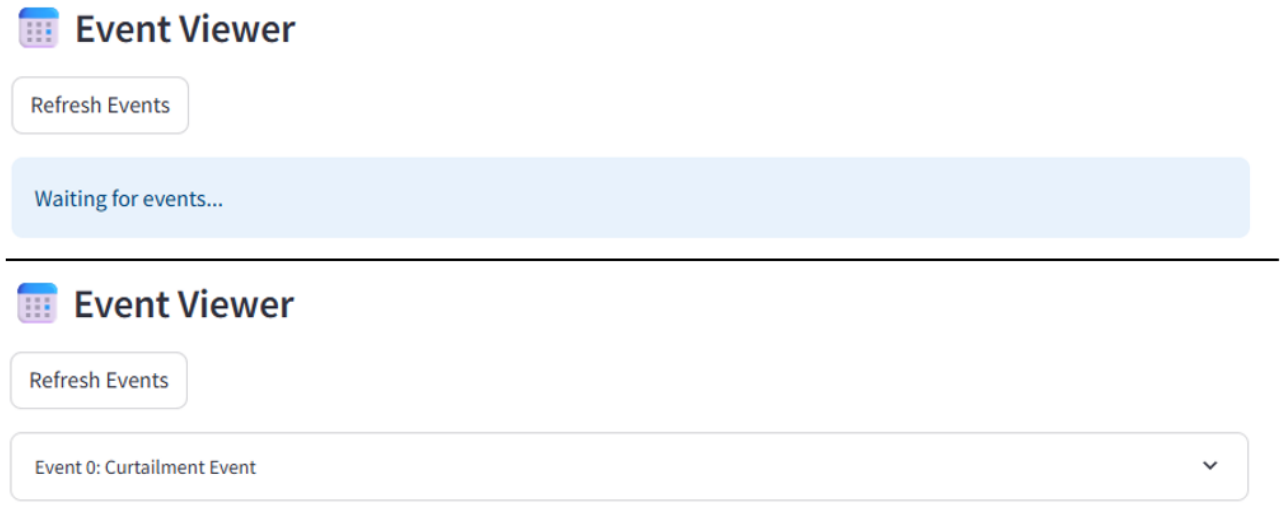

Figure 13: The implemented Event Viewer in the VEN client UI. Illustrates two different scenarios, the top one with no active events, and the bottom one with an ongoing curtailment event.

3.7 Testing and Implementation

To test the functionality of the developed system, two test cases were selected based on their ability to support the stability of the power grid. Due to the wide range of use cases and DR programs supported by OpenADR, the test cases were chosen to reflect scenarios which provide value for Swedish DSOs. Test Case 1 was chosen to be a curtailment event, simulating a situation in which the DSO issues a curtailment signal to an EV. The objective was to observe how the EV responds by reducing its power consumption in accordance with the issued event. Test Case 2 was chosen to simulate a V2G capacity limit scenario. In this case, the test simulated the DSO sending a V2G capacity limit event to an EV with bidirectional charging properties, requiring it to reduce the amount of power exported to the power grid. The following sections present the setup and execution of both the test cases.

3.7.1 Setup Test Case 1

The setup for Test Case 1 required four main components: a laptop, an EVSE, a charging cable, and an EV. The testing was conducted at the Volvo Cars Charging Lab, a facility that provided all necessary equipment and the appropriate environment for testing. The laptop used was the work computer provided by Volvo Cars. During testing, the laptop simultaneously hosted the VTN RI server, the BL client, and the VEN client. In addition, the laptop also maintained a local connection to the EVSE via the correct wireless access point. The EVSE used was a "Volvo Wallbox", and the EV connected for testing was a "Volvo C40". The charging cable was used to physically connect the EVSE to the EV. Figure 14 presents a visual representation of the complete setup for Test Case 1.

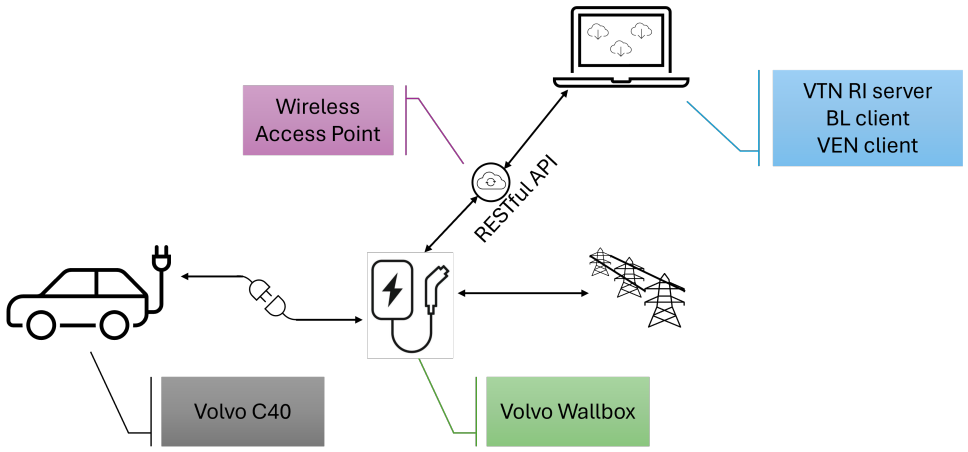

Figure 14: Test Case 1 setup, showing the laptop running the VTN RI server, BL client, and VEN client connected to the Volvo Wallbox EVSE, which in turn is connected to the Volvo C40 EV.

3.7.2 Test Case 1 - EV Curtailment

Once the setup for Test Case 1 was correctly assembled, the testing could begin. The total duration of the test was approximately one hour, during which two similar curtailment events were executed. Figure 15 presents a visual summary of the steps described below.

The initial step was to launch the VTN RI server, BL client, and VEN client on the laptop, ensuring that they ran simultaneously. Launching the BL and VEN clients also starts their respective UIs as local web applications. Once launched, the next step was to login to the EVSE via the VEN client UI and verify the connection between the VEN client and the EVSE. Upon successful login, the "Get User Charging Current" button was pressed in the VEN client UI, obtaining the current charging value of 16 A.

With the system prepared, the BL and VEN clients were configured to the correct VTN address, to ensure the communication occurred over the appropriate port. The BL client then sent a program creation request to the VTN. The request included the user-defined parameters: program name "DRProg", retailer name "Volvo Cars", and a program duration of one day, which was specified in the BL client UI. Once the program creation request was confirmed, a VEN object creation request was initiated in the VEN client UI, with the VEN name "VEN_TEST". Upon creating the VEN object, an associated resource was also created in the UI. This resource was specified as "EVSE", with a maximum power of 22 kW.

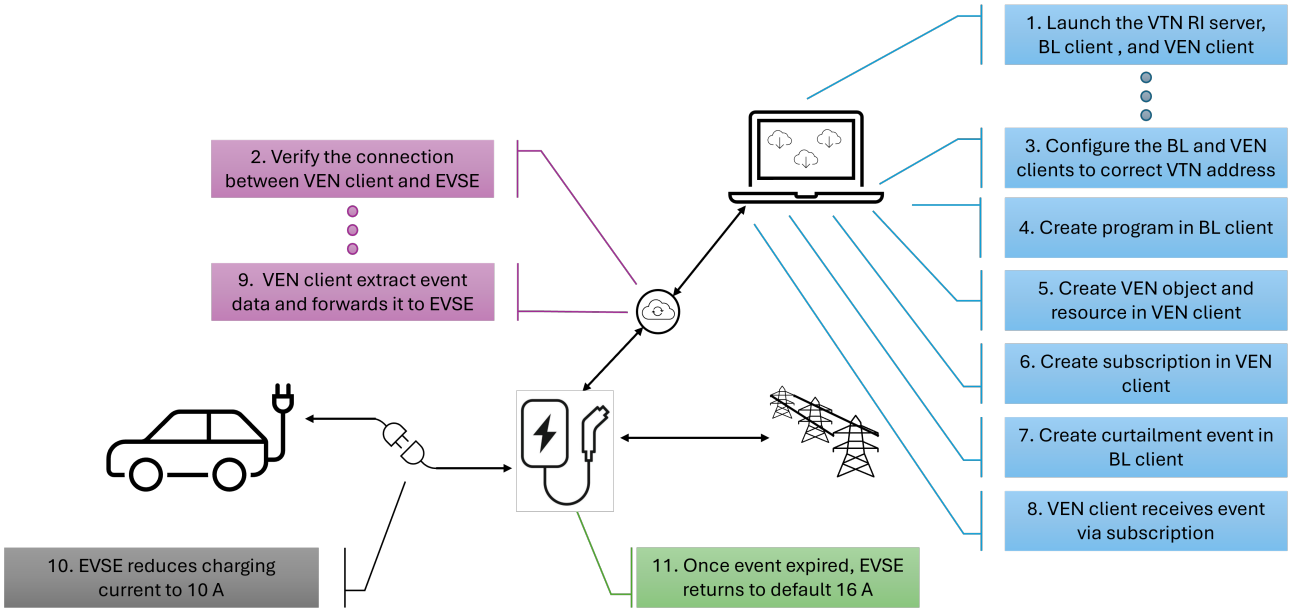

Figure 15: The workflow of the performance of Test Case 1.

When the VEN object and associated resource requests were confirmed, the next step was to create a subscription to enable the VEN client to receive events. In the VEN client UI the program ID was set to "0" to correspond to the most recent created program, the client name was set to "VEN_TEST", and the callback URL used the default value suggested by the VEN client. Once the subscription was confirmed, the system was ready to simulate the curtailment event.

To simulate a curtailment event sent by the DSO, the BL client sent a event creation request to the VTN. The event was defined in the UI to be the event type "Curtailment Event", with the event name matching the event type. The duration was set to be 15 minutes, during which the EVSE should reduce its charging current to the user-defined value of 10 A. The 15-minute duration was selected to verify both the EVSE's functionality to respond to the curtailment

signal, and its ability to return to default charging mode once the event expired. When the curtailment event was confirmed by the VTN, the VEN client automatically retrieved the event data via the subscription. It then automatically extracted the data and forwarded the relevant information to the EVSE. The EVSE then successfully interpreted the data sent from the VEN client, reduced its charging current to 10 A, and thus reduced its power consumption. Once the event duration had expired, the EVSE returned to the default charging value of 16 A.

To validate the system's reliability, a second similar curtailment event was conducted, where the only difference was the duration set to 11 minutes. The same procedure transpired, resulting in the EVSE reducing its charging rate to 10 A during the event. Upon event expiration, the charging current returned to 16 A. The consistent behavior of the system during both events confirmed that the system operated as intended.

3.7.3 Setup Test Case 2

The setup for Test Case 2 required six main components: a laptop, an EVSE, a grid simulator, hardware simulating an EV with bidirectional charging capabilities, a real-time energy meter, and the appropriate connection cables. The testing was conducted in the HIL room at the Volvo Cars Charging Lab, which provided access to all necessary equipment for testing this scenario. The laptop used was the work computer provided by Volvo Cars. During the test, it simultaneously hosted the VTN RI server, BL client, and VEN client. It also maintained a local wireless connection to the EVSE via the correct access point.

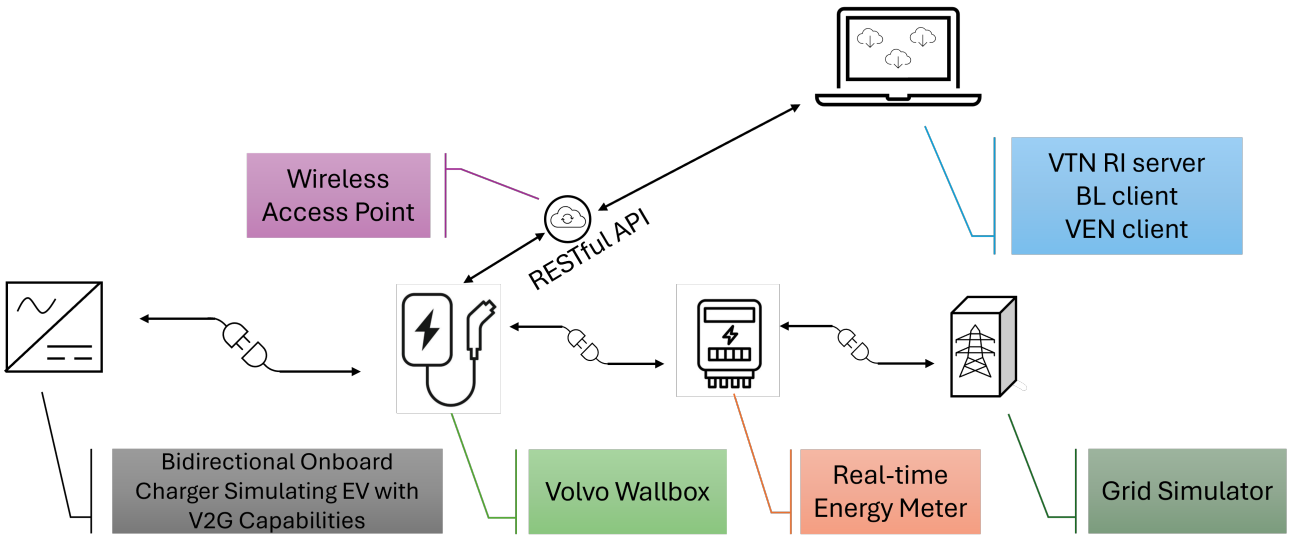

Figure 16: Test Case 2 setup, showing the laptop running the VTN RI server, BL client, and VEN client connected to the Volvo Wallbox EVSE, which in turn is connected to the grid simulator, real-time energy meter, and hardware simulating an EV with V2G capabilities.

The EVSE utilized was a "Volvo Wallbox". It was connected to a grid simulator capable of replicating grid behavior and authorized to accept power export from the EVSE. To simulate an EV with bidirectional charging capabilities, the pre-existing hardware setup in the HIL room was utilized. The EVSE was also linked to a real-time energy meter, which recorded the test session data using the DewesoftX data acquisition software. The connection cables were used to physically connect the EVSE to the simulated EV hardware, the real-time energy meter, and the grid simulator. Figure 16 presents a visual representation of the complete setup for Test Case 2.

3.7.4 Test Case 2 - V2G Capacity Limit

Once the setup for Test Case 2 was properly assembled, the testing could be conducted. The total test duration was approximately 22 minutes, during which three V2G capacity limit events were executed. Figure 17 presents a visual summary of the testing process described below.

Many of the steps for conducting Test Case 2 were identical to those in Test Case 1, as the main differences between the two test cases are the event type and the value sent in the event. The initial step was to launch the VTN RI server, BL client, and VEN client on the laptop, running them all simultaneously. Launching the BL and VEN clients also starts their respective UIs as local web applications. Once launched, the connection between the VEN client and the EVSE was verified by logging into the EVSE through the VEN client UI. Upon successful login and verification, the system was correctly prepared.

With the system prepared, both the BL and VEN clients were configured to communicate with the VTN over the appropriate port. This was done by setting the correct VTN address in both the BL and VEN clients UIs. The next step was to create a program, by sending a program creation request from the BL client to the VTN. The request included the user-defined parameters: program name "DRprog", retailer name "Volvo Cars", and a program duration of one day, specified in the BL client UI. When receiving program creation confirmation from the VTN, the VEN client sent a VEN object creation request to the VTN, with the name "VEN_TEST", defined in the UI. Upon creation of the VEN object, an associated resource was created in the VEN client UI. This resource was specified as "EVSE", with a maximum power of 11 kW, corresponding to the EVSE export capability.

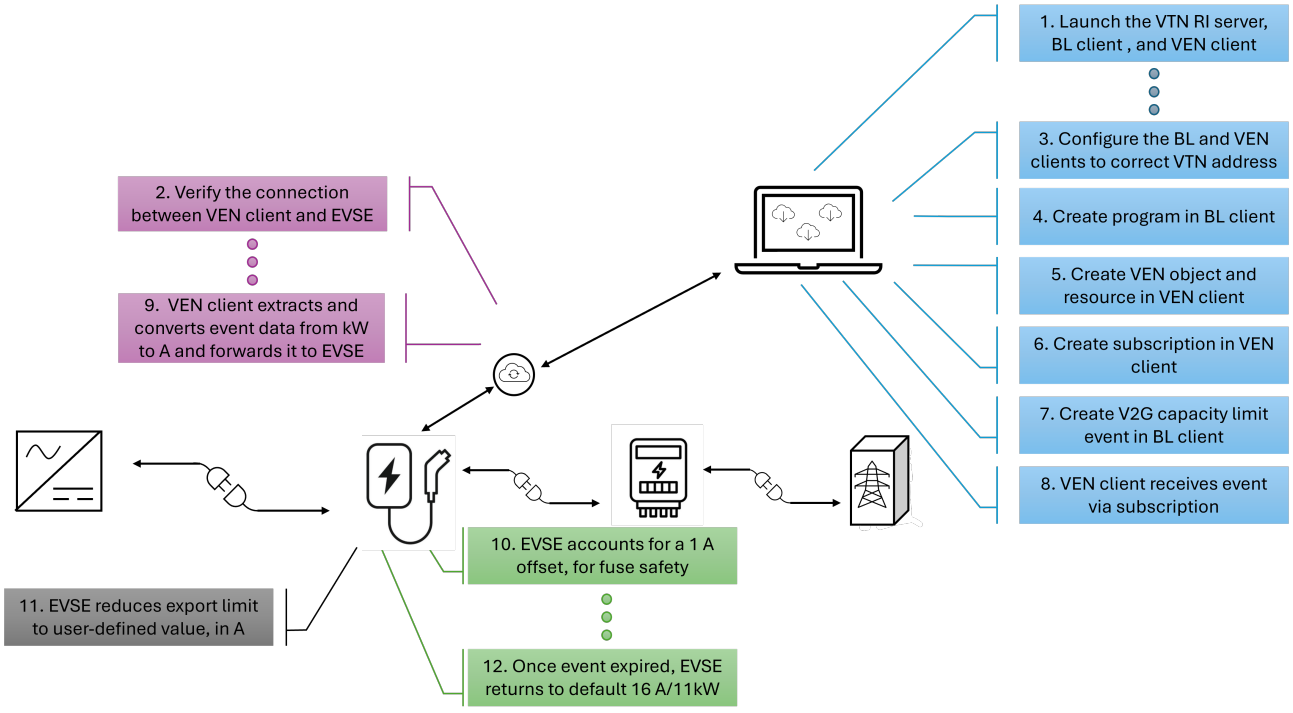

Figure 17: The workflow of the performance of Test Case 2.

Following the creation of the VEN object and associated resource, the next step was to create a subscription, enabling the VEN client to receive events issued by the BL client. In the VEN client UI, a subscription creation request was sent to the VTN with the user-defined parameters program ID set to "0", corresponding to the most recently created program, the client name set to "VEN_TEST", and the callback URL as the default value suggested by the VEN client. Upon subscription confirmation, the system was ready to simulate the V2G capacity limit event.

To simulate the first V2G capacity limit event sent by the DSO, the BL client issued an event creation request to the VTN. In the UI, the event type was set to "V2G Capacity Limit Event", with a matching event name. The duration was set to 6 minutes, during which the EVSE was expected to reduce its power export to the user-defined value of 6 kW. This duration was selected to test both the EVSE ability to respond to the V2G capacity limit signal and its capacity to return to default export rate at event expiration. When the V2G capacity limit event was confirmed by the VTN, the VEN client, through the active subscription, automatically retrieved the event data, extracted the relevant parameters, and converted the power value from kilowatts to amperes, as the EVSE only supports current via its RESTful APIs. After completing the conversion, the VEN client forwarded the data to the EVSE, which due to a built-in 1 A offset to protect the fuse, lowered the current data by the offset value. Therefore, upon successful unit conversion and the 1 A offset factored in, the EVSE reduced its power export to approximately 7 A, equivalent to about 4.7 kW, slightly lower than the expected 6 kW. Once the event expired, the EVSE returned its default export limit of roughly 16 A/11 kW.

A second V2G capacity limit event was then simulated. The event creation followed the same procedure, but with the duration set to 15 minutes and the export limit to 4 kW. Upon event creation confirmation, the same steps transpired, resulting in the EVSE responding to the event. This time, however, the EVSE reduced its power export to approximately 4 A/2.6 kW. This value was unexpectedly low even when accounting for the 1 A offset. Thus, an error in the code was suspected and the code was reviewed, leading to canceling the event after approximately 2 minutes. Upon event cancellation, the EVSE returned to its default mode.

The error was found and corrected, and was due to a type conversion error in the code. The code interpreted a float value as an integer during unit conversion from kW to A, further diminishing the current in addition to the 1 A offset. Thus, a third V2G capacity limit event was simulated with the corrected code. The third event was identical to the second one, but with the duration set to 9 minutes. This time, the EVSE responded with a reduced power export of approximately 5 A/3.2 kW, which aligned better with the expected output after accounting for the 1 A offset, and the test was considered successful. When the third event expired, the EVSE returned to its default export limit of approximately 16 A/11 kW.

4 Results

In the following section, the results from the project are presented. Section 4.1 begins with presenting the key insights from the interviews, showcasing their importance for several important decisions made throughout the project. Next, given that a major part of the project was the development of a functioning system utilizing OpenADR 3.0, Section 4.2 displays the establishment of the communication. It includes the payloads used in all the object creations and the confirming acknowledgment received from the VTN, demonstrating successful data exchange. Finally, Section 4.3 and 4.4 present the results from Test Case 1 and Test Case 2, respectively. These sections present that the EVSE responded in accordance with the events, adjusting its behavior as intended, based on the received event data.

4.1 Insights from interviews

As described in the methodology, interviews were conducted with the important actors Energiföretagen, E.ON, Ellevio, Vattenfall Eldistribution AB, and OpenADR Alliance. These interviews produced many valuable insights and contributions. Although all questions provided relevant information for this project, some offered key insights. Thus, these selected questions, along with their key insights, are summarized in Table 9.

The insights gathered from the interviews served as guiding tools throughout the development of this project. One important decision was the choice between OpenADR 2.0b and OpenADR 3.0. Based on the responses from the interviewees, it became clear that OpenADR 3.0 is the most preferred and scalable option. Additionally, the OpenADR Alliance provided access to their OpenADR 3.0 VTN Reference Implementation, making version 3.0 the natural choice for system development.

Another key decision was selecting which use cases to implement and test. A key insight was that Swedish DSOs already are exploring the use of OpenADR for conditional grid agreements, primarily for consumption reduction. Combined with information regarding additional possible use cases, this project decided on focusing on the curtailment event, which reflects current real-world applications and the V2G capacity limit event, highlighting the potential for utilizing OpenADR in future scenarios.

Additionally, although not listed in Table 9, the interviews also offered valuable insights and guidance for system development. Information about the implementation and functionality of E.ON's SWITCH VTN and Ellevio's VTN provided key insights into the general architecture of OpenADR and into how various payload objects should be structured and designed.

Table 9: A selection of the interview questions and the key insights gained from the responses.

Question	Insight/Contribution
Which OpenADR communication case is the best /most promising for DSOs? Direct contact with DERs or through an aggregator?	Swedish DSOs generally prefer communication through aggregators rather than direct communication with individual EVSE. This is due to benefits such as greater available capacity, enhanced security, and a higher level of professionalism. Aggregators also often manage a diverse set of resources, some of which are more or less suitable for different DR programs, allowing for a more efficient and flexible management.
According to you, what are the next steps to establish OpenADR as the industry standard in Sweden?	Collaboration and ongoing discussion among all Swedish DSOs are essential. Important to together define use cases where OpenADR could be utilized and reach agreement on appropriate deployment. Current challenges are coordination issues and possible business models. Thus, a proper standardization is required. A step in the right direction is the industry recommendation published by Energiföretagen.
What would you say are the main differences between OpenADR 2.0b and 3.0?	OpenADR 2.0b uses Simple Object Access Protocol (SOAP), has restrictive communication methods, and more complex programming. OpenADR 3.0 is a more intelligent and modern technology, using RESTful API and HTTPS, which is more accessible for developers.
Would you recommend using OpenADR 2.0b or OpenADR 3.0?	OpenADR 3.0 is the latest version and is expected to be the standard for future implementations. Using OpenADR 3.0 would reflect a real-world case, while version 2.0b would more of an academic case. In both Sweden and in Europe, OpenADR 3.0 is the version currently under discussion and adoption.
Beyond conditional grid agreements, what are other relevant use cases for OpenADR in EV charging in different markets? What other use cases are being adopted in Europe?	The Netherlands have significant problems with grid congestion. Thus, they are exploring using OpenADR 3.0 to solve the communication between DSOs and CPOs. In the U.S., several car manufacturers are utilizing OpenADR to delay EV charging to smoothen out peak loads. There is a broad range of use cases, including charging/discharging power control and dynamic energy price signals. Additionally, OpenADR itself does not define a business model, but works as an equipment which provides the communication between devices and actors. The customers come up with the business model themselves.
Is it possible to get access to the ”OpenADR 3.0 VTN Reference Implementation” without being a contributing member?	The interviewee provided access to the official GitHub repository containing the OpenADR 3.0 VTN Reference Implementation.

4.2 Establishing the communication

Developing the system was a major part of the project. This process included building the BL client and VEN client, and establishing their communication with the VTN server. By establishing this communication, it enabled the Bl and VEN clients to send object creation requests, and ensured that these objects were successfully created and registered on the VTN server. This section therefore illustrates the request payloads sent from the BL client and VEN client UIs, including both the statically defined and user-defined parameters. Each figure shows the full payload sent in the request along with a response confirming that the object was successfully created and registered with a automatically assigned ID on the VTN server. Furthermore, the VTN server also provisions each object with additional parameters, as explained in the methodology. However, these are not shown in the UI responses. The figures are presented in the same order that the objects were created during the execution of Test Case 1 and Test Case 2.

Figure 18 showcases the request for program object creation sent by the BL client. The payload includes user-defined values, statically set parameters, and default values. The figure also confirms that the program was successfully created and assigned the VTN provisioned ID 0.

Figure 18: The request for program object creation sent by the BL client to the VTN server. It includes the full payload, and the response confirms successful creation of the program with VTN provisioned ID 0.

As described in the methodology, the VEN object was kept simple. Figure 19 shows the VEN object creation request sent by the VEN client, where the only payload required is the user-defined VEN name. The figure also displays that the VEN object was successfully created, with the VTN provisioned ID 0.

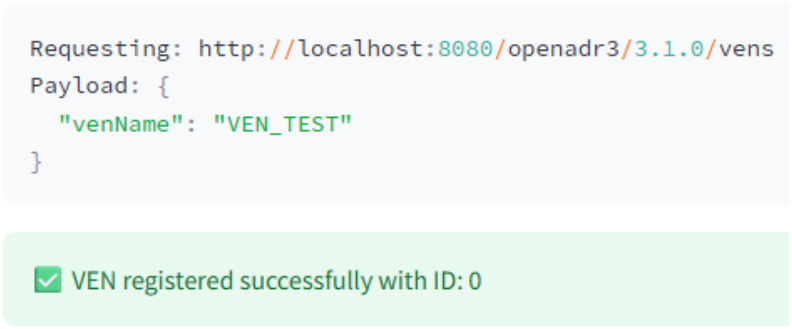

Figure 19: The request for VEN object creation sent by the VEN client to the VTN server. It includes the full payload and the response that confirms the creation of the VEN object with ID 0.

Figure 20 presents the request for resource object creation, also sent by the VEN client. As described in the methodology, the endpoint `"/vens/0/resources"` indicates that the resource is directly associated to the VEN object with ID 0. Thus, the payloads displayed in the figure are being incorporated in the VEN object as well. Furthermore, the figure also displays the response that confirms the successful creation and registration of the resource object, with the VTN provisioned ID 0.

Figure 20: The request for resource object creation sent by the VEN client to the VTN server. It includes the full payload associated with VEN object 0, and the response that confirms the creation of the resource object with ID 0.

Figure 21 shows the request for the subscription object creation, which is also sent by the VEN client. The subscription is associated to the program with ID 0, as indicated by the `"programID"` parameter. The subscription registers interest in the four object types `"EVENT"`, `"PROGRAM"`, `"RESOURCE"`, and `"VEN"`, and requests a callback on `"CREATE"`, `"UPDATE"`, and `"DELETE"` operations. Worth mentioning is also the parameter `"bearerToken"`, which is a randomly assigned static authorization token. The figure presents the response that confirms the successful creation and registration of the subscription object with the VTN provisioned ID 0.

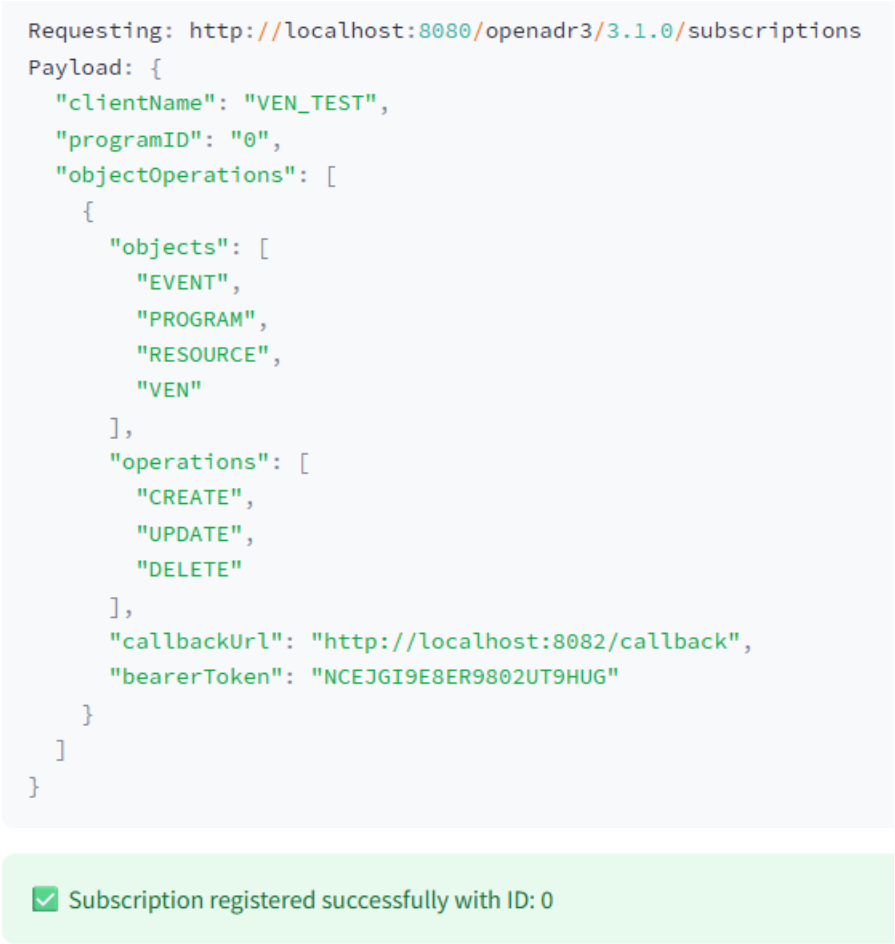

Figure 21: The request for subscription object creation sent by the VEN client to the VTN server. It includes the full payload, which is linked to program ID 0 and registers interest in several object types and operations. The response confirms the creation of the subscription object with ID 0.

Figure 22 illustrates the request sent by the BL client when creating the event object for Test Case 1 - the EV curtailment event. Similar to the subscription object, the event is associated to the program with ID 0, as indicated by the "`programID`" parameter. Since the event is an important entity in OpenADR, the payload includes a wide range of parameters. However, the most vital ones are "`values`", "`units`", and "`intervalPeriod`", the latter of which includes the "`start`" and "`duration`" parameters. These are the key parameters that instruct the EVSE of the curtailment value, what unit the value corresponds to, when the curtailment should start, and for which duration. As shown in the figure, the event is successfully created and assigned the VTN provisioned ID 0.

Figure 22: The request for curtailment event creation sent by the BL client to the VTN server. It includes the full payload and the response that confirms the creation of the curtailment event with ID 0.

Finally, Figure 23 shows the request sent by the BL client when creating the event for Test Case 2 - the V2G capacity limit event. Like the curtailment event, this event is also associated to the program with ID 0. The request differs from the curtailment event only in a few parameters, which results in a very similar payload and highlights one of the key strengths of OpenADR. Entirely new use cases can be implemented by modifying just a few parameters, showcasing the flexibility and standardization offered by the protocol. As with the curtailment event, the most important parameters are "`values`", "`units`", and the parameters within "`intervalPeriod`". As shown in the figure, the event is successfully created with the ID 0.

Figure 23: The request for V2G capacity limit event creation sent by the BL client to the VTN server. Similar to the curtailment, it includes the full payload, with minor parameter changes to support a different use case. The response confirms the creation of the V2G capacity limit event with ID 0.

4.3 Test Case 1 - EV Curtailment

The results from Test Case 1 were gathered from the EVSE UI, where screenshots were taken to document the test. Figure 24 shows a summary of the session, including the start time, the total energy consumption, and overall duration.

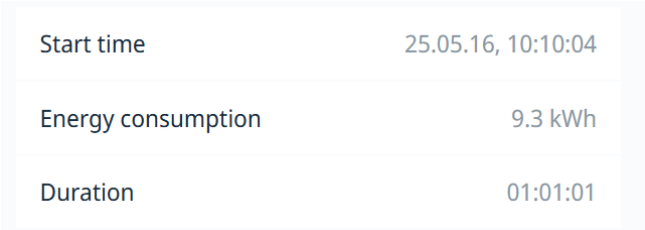

Figure 24: Overview from the EVSE UI displaying the test session's start time, total energy consumption in kWh, and total duration, including the two curtailment events.

The EVSE UI also provided real-time data about the connected EV and its charging behavior. Figure 25 presents this data during default conditions, when no curtailment event was active. It confirms that the EV was actively charging with a charging capacity of 10.92 kW and a maximum charging current of 16 A.

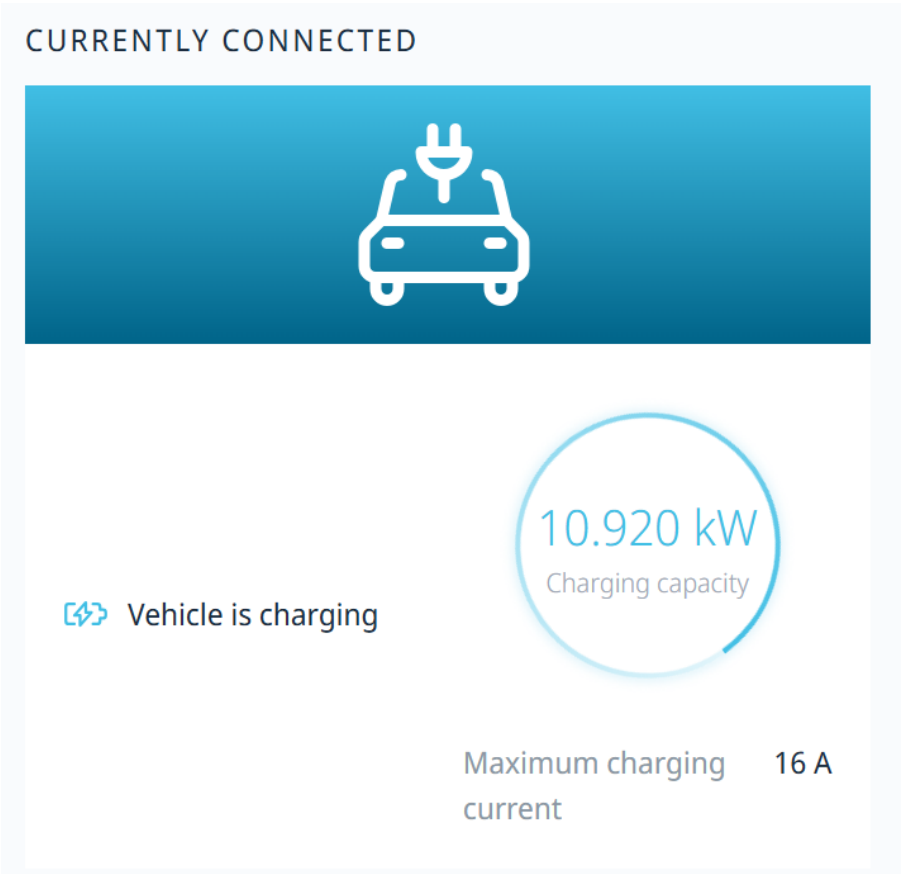

Figure 25: Charging status during default conditions, with a charging capacity at 10.92 kW and a maximum charging current at 16 A.

During the curtailment event, the EVSE adjusted its charging behavior as intended. Figure 26 presents the real-time data of the connected EV during the event. The EV remained connected and charging; however, the charging was limited to a charging capacity of 6.93 kW and a maximum charging current of 10 A, as specified in the event. This confirms that the EVSE responded to the curtailment event correctly.

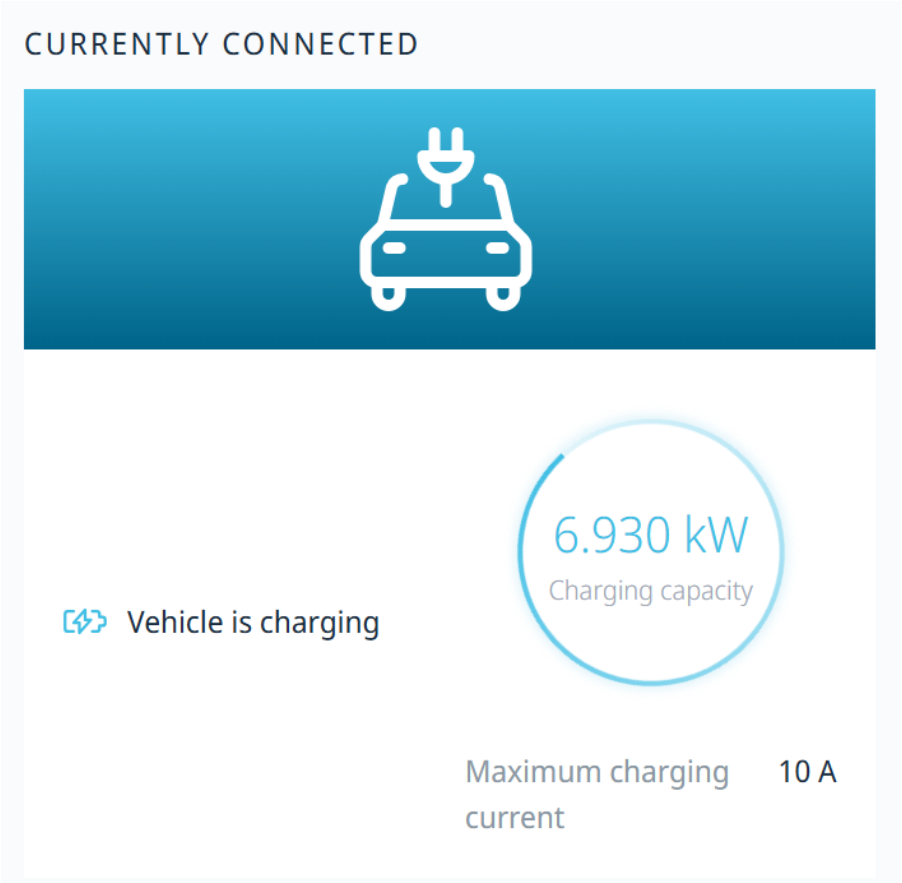

Figure 26: Charging status during curtailment event, with the limited charging capacity at 6.93 kW and maximum charging current at 10 A.

Figure 27 shows the consumption curve recorded by the EVSE UI, visualizing the impact of the two curtailment events during the charging session. The x-axis represents time, spanning between 10:00 to 11:15. In the graph, the first event has a longer impact period than the second, due to the duration of 15 minutes compared to 11 minutes. The y-axis represents the charging power in kW, with the default charging power at approximately 11 kW and curtailed charging power at around 7 kW, in accordance with the data presented in Figures 25 and 26. The key takeaway is that the EVSE successfully curtailed the charging during both events and returned to default operation upon their expiration.

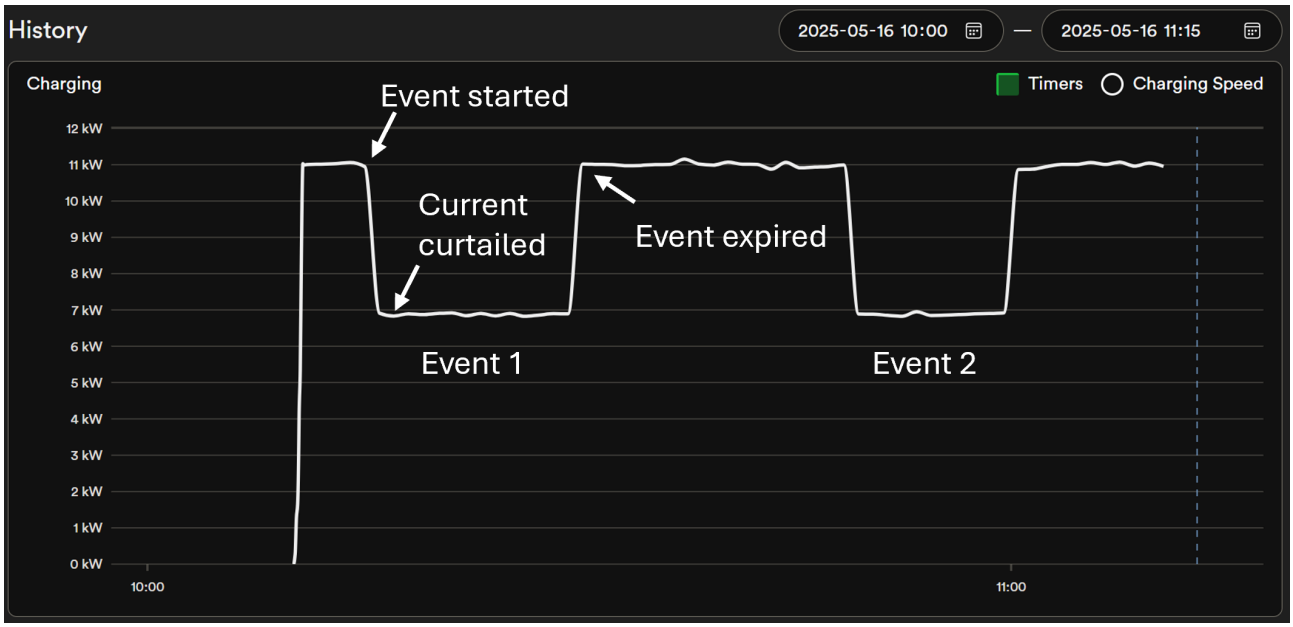

Figure 27: The EVSE consumption curve showing the impact of two curtailment events in the charging session. The charging power is curtailed during each event and returns to default levels upon event expiration.

Similar results are presented in Figure 28, which displays another graphical summary of the charging session and the impact of the two curtailment events. The x-axis represents the actual test duration, while the y-axis lacks a specified unit, presenting a limitation. However, the main takeaway remains clear, the EVSE successfully curtailed the charging during both events and returned to default operation after their expiration. Worth mentioning is that the energy consumption value is being displayed as 0 in the figure. This is because the screenshot was taken after the testing session ended and the EVSE had reset its measurements. The actual total energy consumption, as presented in Figure 24, was 9.3 kWh.

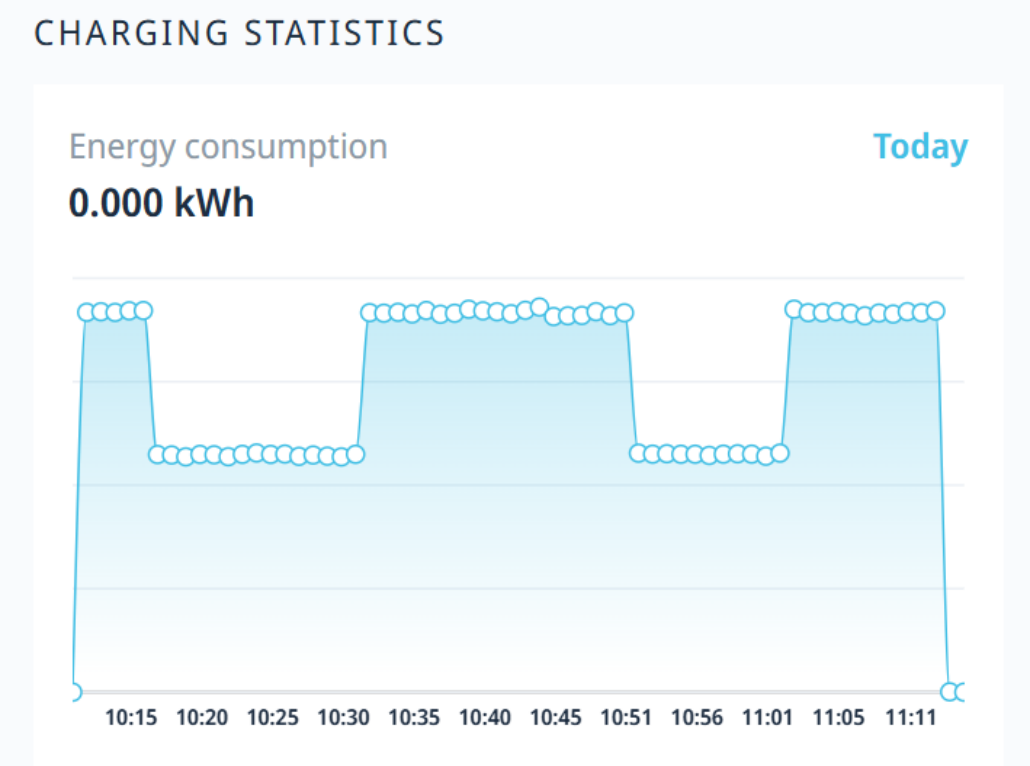

Figure 28: The EVSE UI charging statistics, illustrating the impact of two curtailment events in the charging session. While the energy consumption value is displayed as 0 due to measurement reset in the EVSE, the correct value is 9.3 kWh.

4.4 Test Case 2 - V2G Capacity Limit

The results from Test Case 2 were recorded using the real-time energy meter, which documented the test session data using the DewesoftX data acquisition software. The DewesoftX application can produce a great variety of figures, values and graphs. However, the key measurements from Test Case 2 are the power export and current, since the test focused on the EVSE response to the V2G capacity limit event.

Figure 29 displays the power export profile during the testing session. The EVSE successfully responded to the three V2G capacity limit events, limited its power export during each event, and returned to default export rate upon event expiration. The y-axis represents power export in watts, with negative numbers indicating power export from the EVSE. A larger negative number thus corresponds to a greater power export. The x-axis represents the total testing session duration, which lasted 22 minutes and 23 seconds.

Notably, the export limits do not perfectly match the 6 kW and 4 kW limits specified in the events. This deviation is caused by the EVSE built-in 1 A offset fuse safety function. Additionally, the power export values differ between the second and third event, even though they were specified as the same limitation of 4 kW in the event. This difference is explained by the coding error which was discovered during the second event. This error was corrected prior to the third event and, as a result, the third event aligns better with the expected value when accounting for the 1 A offset.

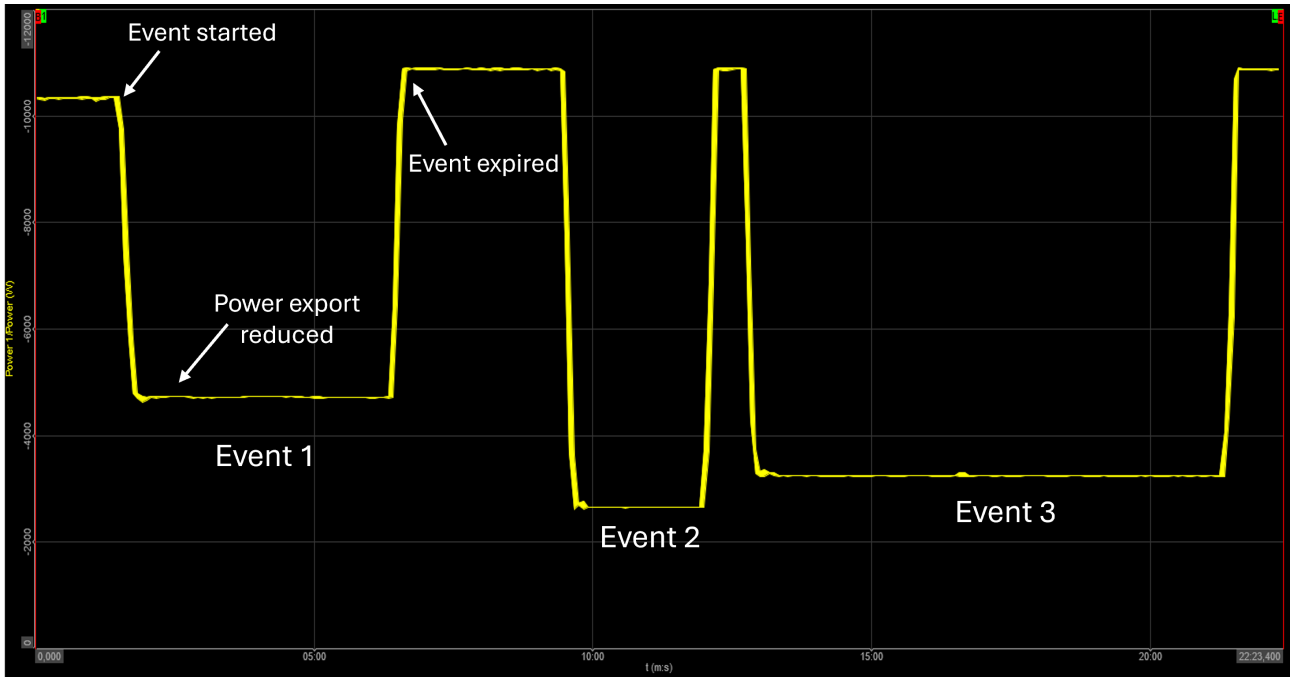

Figure 29: The power export profile during the testing session, showing the impact of three V2G capacity limit events. Power is shown in watts, with negative values indicating power export.

Figure 30 presents the current draw during the testing session. Similar to the power export graph, the current is limited during each of the three events, confirming that the EVSE successfully responded to the events and returned to its default mode upon event expiration. The y-axis represents the current in amperes, while the x-axis is the total testing session duration. The figure also highlights the deviation between the second and third event, where the current is lower during the second event due to the same coding error discussed previously. After correcting the error, the current during the third event aligned better with the expected value.

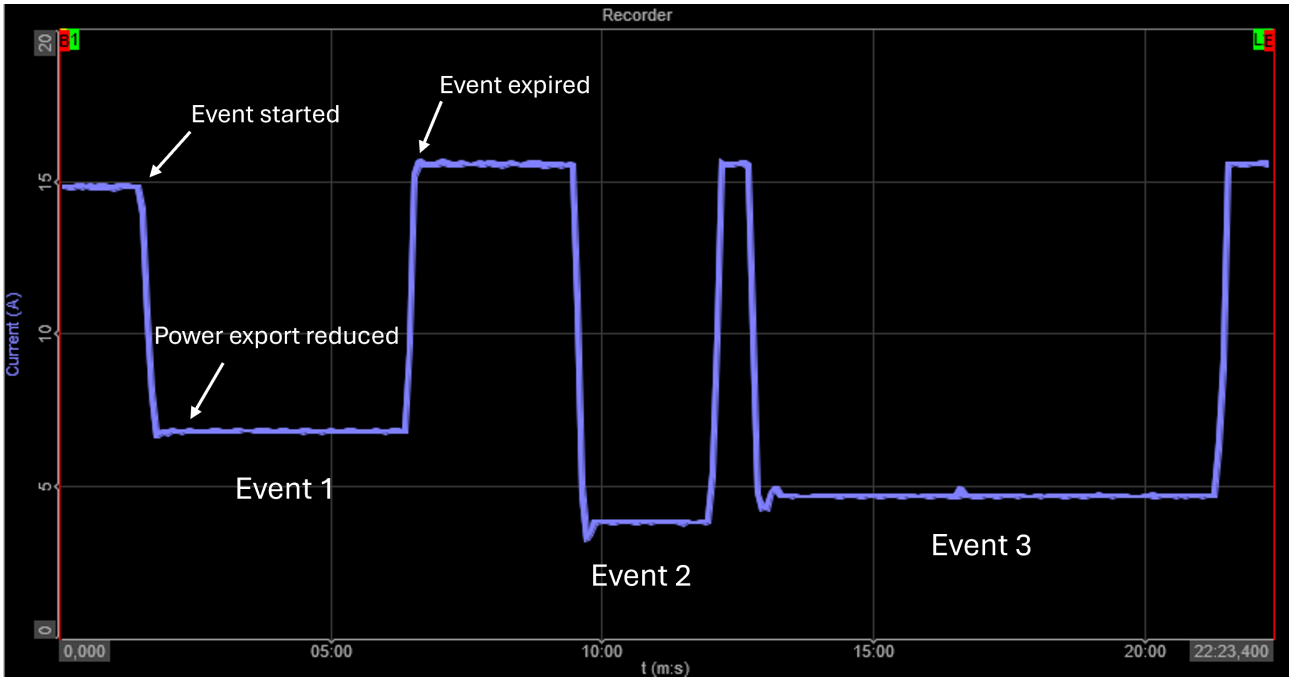

Figure 30: The current draw profile during the testing session, showing the impact of three V2G capacity limit events.

Finally, Figure 31 presents a snapshot summary of the real-time values for both the power export and current during default mode and each of the three V2G capacity limit events. The summary highlights the code correction clearly; in the second event the power export was limited to 2.65 kW, whereas after correcting the code, the third event limited the power export to 3.25 kW instead. Additionally, the summary highlights the current deviation caused by the coding error. The current is 0.8 A lower in the second event compared to the third, clearly showing the impact of the conversion error in the code. As before, the negative power export values indicate that power is being exported.

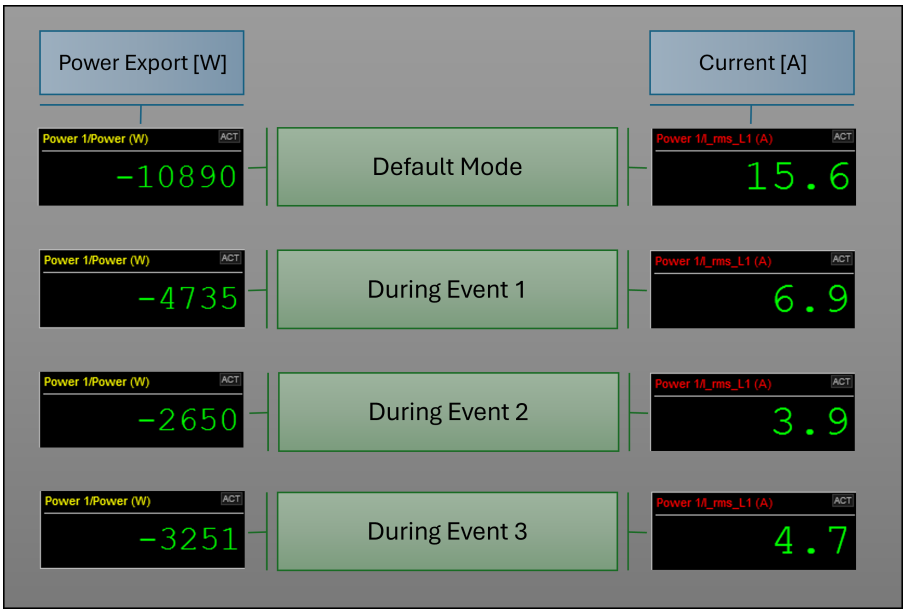

Figure 31: A snapshot summary showing the real-time values for both the power export and current during default mode and during each of the three V2G capacity limit events. Negative power values indicate power export.

5 Discussion

In Sweden, DSOs have begun utilizing OpenADR for the use case conditional grid agreements. These agreements allow customers to install and connect new resources, such as EVSE, to the power grid on the condition that they support the maintenance of grid stability when needed. This setup offers a mutually beneficial solution for both the DSO and the customer. The customers benefit from faster grid connection processes, while the DSOs gain access to a greater fleet of flexible resources that can be managed to enhance grid resilience and stability.

Currently, the Swedish DSOs E.ON and Ellevio have implemented conditional grid agreements using OpenADR, where E.ON's implementation focuses on reducing consumption or setting a production limit, and Ellevios focuses on reducing consumption. However, this project demonstrates that OpenADR can be utilized to reduce power export from the V2G technology as well. Beyond EV charging and discharging, OpenADR could also be used to manage a diverse set of resources, including batteries, heat pumps and thermostats. Events can be created to increase or decrease energy consumption or export, to pause resource operation fully, or to adjust the resource behavior based on energy price.

This proves the flexibility, modularity, and scalability of the OpenADR architecture. With small changes in the payload structure, the protocol can enable completely new use cases, as shown in this project. However, OpenADR does not define the business models of these use cases, but works as equipment for enabling communication. This poses a problem - while the technology may be ready, the business model and value proposition for both DSOs and customers must be discussed and defined. Questions such as "How does this use case benefit the DSO?" and "What does the customer gain from adopting OpenADR for this use case?" must be considered before OpenADR can be widely adopted in Sweden.

Regarding the use cases investigated in this project, the charging curtailment and V2G capacity limiting, only the curtailment use case has a defined business model through existing conditional grid agreements. Thus, it is important to discuss a business model that could be introduced for the V2G capacity limit use case. As described, conditional grid agreements can be recognized as a mutually beneficial solution for the DSO and the customer. To address this solution, one potential approach could be to introduce the new business model called "V2G conditional grid agreements."

For the customer this would mean that they are allowed to install and connect a bidirectional EVSE to the grid, even in areas where such a request would normally get rejected. In exchange, they allow their resource to be managed by the DSO during certain periods. This would offer a benefit to the customer, enabling the ability to deploy the V2G technology despite local grid constraints. However, limiting the export power could lead to unfavorable loss of income. To counteract this, a solution could be to provide event-based compensation for the customer. However, this would require further research to assess the fairness and viability of this compensation.

From the DSOs' perspective, some areas in Sweden have a high photovoltaic integration and production, meaning that injecting more power to the grid via V2G could cause additional problems. However, the energy system is evolving, and in the near future technologies such as V2G will need to be incorporated in a way that supports the grid rather than challenges it. Thus, the V2G capacity limit event could reduce the impact of these problems, resulting in a means for greater adoption of V2G. Furthermore, OpenADR could help fulfill grid code requirements in Sweden. For resources that generate power, grid code requirements include the remote reduction and ceasing of active power based on DSO instructions, as well as the

remote parameter exchange for monitoring and control. Both of these requirements can be fulfilled within the V2G capacity limit event. Thus, OpenADR can fulfill important grid code requirements for the V2G capacity use case, and it could possibly simplify the deployment of V2G via the proposed business model "V2G conditional grid agreements".

The VTN server used in this project was the OpenADR 3.0 VTN Reference Implementation, provided by the OpenADR Alliance. This server was essential for development and testing, providing the required tools for enabling the communication between the clients. However, because this server is not connected to a real DSO, a BL client had to be developed in addition to the VEN client to simulate operations otherwise performed by the DSOs. Only the VEN client would need to be implemented by the customer in a production deployment, resulting in a reduced system complexity. Although efforts were made to gain access to E.ON's and Ellevio's VTN implementations in the interviews, their systems are already launched and in production, thus unavailable for external testing. As a result, this project needed to simulate the whole system, including both the BL and VEN clients. This is worth mentioning, since real-world customers would face a less complex integration than was required in this project.

During the testing of the V2G capacity limit event, a coding error was discovered. This was due to the VEN client interpreting a float value as an integer when converting the limitation value from kilowatts to amperes for communication with the EVSE, which only supports amperes through its RESTful API. This coding error was discovered during the second test event, leading to inaccurate results in the first two test events. However, after correcting the error, the third test event gave the expected results, deviating only due to the 1 A offset. Despite the inaccurate results in the first two test events, the key discovery remained clear. The EVSE successfully responded to the events and returned to default operation upon event expiration. Thus, the results still confirm the important finding that the system functioned correctly. It is also worth mentioning that with additional testing time, the 1 A offset could have been accounted for in the code, resulting in a more precise result. However, since the key finding was clear, the results were deemed satisfactory and sufficient.

One important aspect that was not accurately measured in this project was the system response time. Measuring the interval between the creation of an event and the EVSE reaching the specified event value would have provided valuable insights into system performance. While this could have been achieved by implementing a timer to track this, several factors made it impractical. These factors were time constraints, lack of accurate EVSE UI timestamps, and the complexity of synchronizing timestamps across multiple components. However, approximation of the response time was made in the V2G capacity limit test using DewesoftX, where the EVSE successfully limited the export rate roughly 10 seconds after receiving the event. Important to note is that Ellevio's requirements are that the VEN must acknowledge events within 5 minutes. Thus, given that the observed response time was within seconds, not minutes, this was not considered an issue. Still, accurate response time measurement would provide valuable insights, and could be an improvement for future work.

5.1 Future Studies

There are several future studies and works that can further improve this project. Essentially, if the system is to transition from a local testing environment to a real-world deployment, authorization needs to be implemented. A widely adopted solution is the OAuth2 client credentials flow, which provides the system with a client ID and secret that can be traded for short-lived tokens. These tokens can then be utilized to securely authenticate the client with the VTN server, authorizing it access to perform certain operations and receive events. By implementing authorization, the system would enhance its security, and enable safe operations outside a local testing environment.

Another key feature required for real-world deployment is the report functionality. This functionality is implemented within the VEN client, and is used to generate responses to specific events. The complexity of the report can vary, and can be both minimal and advanced implementations. A minimal implementation could serve as a acknowledgment tool, where the VEN client confirms it has received an event and indicates whether the resource has responded to it. A more advanced implementation could also include real-time data, providing the BL client with deeper insights into the resource's behavior. Regardless of complexity, implementing the report functionality is mandatory for the system to work outside a local environment.

Since this project tested with one single EVSE, a valuable next step would be to extend the system to support multiple EVSE, allowing for resource aggregation. This would increase the manageable capacity for the DSO and provide insights into real-world aggregator scenarios. This enhancement would require some restructuring in the code, since it would remodel the current VEN client into a cloud application, managing multiple EVSE under one interface. However, it could prove useful to investigate the system's performance in various scenarios.

Additionally, another valuable next step could be to develop the system to support integration with other communication protocols, such as OCPP and EEBus. Enabling interoperability with these protocols would provide greater flexibility to the system, facilitating means for the system to support different communication pathways, not only direct communication, and enabling a wider range of use cases. This would also present the opportunity to further investigate how OpenADR can cooperate with other communication standards within the EV related domain.

Finally, an essential future work would be to initiate a test pilot in collaboration with an actual DSO. Such a pilot would initially require the system to implement the previously mentioned authorization and reporting functionalities, to ensure reliable and secure communication. However, with a refined system, a test pilot would be possible. A particularly interesting test pilot would be to explore the V2G capacity limit use case outside a local testing environment. Utilizing this use case in collaboration with a DSO could generate important findings and potentially simplify the future deployment of the V2G technology in Sweden.

6 Conclusions

This project has investigated various communication protocols which facilitate the direct interaction between the DSO and the EVSE, or indirectly through third parties, such as aggregators, for DR purposes. An analysis and comparison of these protocols concluded that OpenADR was the most promising communication protocol for adoption in Sweden. This conclusion is based on the protocol's interoperability, its widespread adoption, both globally and in Sweden, and since it can be implemented by several different stakeholders. Additionally, the Swedish special interest organization Energiföretagen has published an industry recommendation, in which they evaluated which communication protocol was the most suitable to implement as an industry standard in Sweden - from which they also concluded that it was OpenADR.

Furthermore, this project has conducted several interviews with key actors in the Swedish energy sector. These discussions provided several key insights, and revealed that Swedish DSOs believe that the most effective communication pathway to the EVSE is via an aggregator, not through direct communication. This is due to benefits such as greater available capacity, enhanced security, and higher levels of technical and operational professionalism. Additionally, aggregators often manage a diverse set of resources, some of which are more or less suitable for various DR programs, enabling a more flexible and efficient management.

Building on the conclusion that OpenADR is the most suitable protocol for adoption in Sweden, a system utilizing OpenADR was developed as a core part of this project. This system demonstrated that OpenADR can be used to facilitate direct communication between DSOs and EVSE for multiple use cases. To enable the different use cases, the system only required minor payload changes, which highlights the flexibility, modularity and scalability of the OpenADR protocol.

The use cases implemented in this project were the curtailment event and the V2G capacity limit event. Currently, Swedish DSOs have implemented conditional grid agreements using OpenADR to reduce consumption, similar to the curtailment use case in this project. However, the V2G export limit use case proposes a new application. Conditional grid agreements offers a mutually beneficial solution for both customers and DSOs, where customers benefit from faster grid connection processes, while DSOs gain access to a larger pool of flexible resources for grid management. As proposed in the discussion, introducing a business model named "V2G conditional grid agreements" could provide similar benefits, potentially simplifying the deployment of the V2G technology in Sweden.

Finally, there are regulatory grid code requirements that must be fulfilled when installing resources generating power. These requirements include the ability to remotely reduce and cease active power based on DSO instructions, as well as the ability to remotely exchange parameters for monitoring and control. The V2G capacity limit event developed in this project can fulfill both of these requirements, concluding that OpenADR can fulfill grid code requirements relevant for V2G deployment.

References

[1] Energimyndigheten, "Sveriges energi- och klimatmål," 2025. Accessed on 18 August 2025. https://www.energimyndigheten.se/om-oss/om-energimyndigheten/energimyndighetens-uppdrag/sveriges-energi–och-klimatmal/.

[2] Energiföretagen, "Elektrifiering," 2023. Accessed on 10 June 2025. https://www.energiforetagen.se/energifakta/elsystemet/elektrifiering/.

[3] Energimyndigheten, "Myndighetsgemensam uppföljning av samhällets elektrifiering," Huvudrapport, Energimyndigheten, 2024.

[4] Sweco, "Elnätsrapporten 2023 - Investeringsbehovet i det svenska kraftsystemet till 2045," 2023. Accessed on 10 June 2025. https://www.ellevio.se/globalassets/content/nyheter-pressrum/elnatsrapporten-2023.pdf.

[5] E. Lindholm, M. Malmaeus, E. Löfblad, and J. Westerberg, "Ledtider för att bygga ut det svenska elsystemet," NEPP Resultatblad 6/2025, IVL Svenska Miljöinstitutet, Profu, Energiforsk, 2025.

[6] M. Khalid, "Smart grids and renewable energy systems: Perspectives and grid integration challenges," *Energy Strategy Reviews*, vol. 51, 2024.

[7] J. J. Moreno Escobar, O. Morales Matamoros, R. Tejeida Padilla, I. Lina Reyes, and H. Quintana Espinosa, "A Comprehensive Review on Smart Grids: Challenges and Opportunities," *Sensors*, vol. 21, no. 21, 2021.

[8] C. Lamnatou, D. Chemisana, and C. Cristofari, "Smart grids and smart technologies in relation to photovoltaics, storage systems, buildings and the environment," *Renewable Energy*, vol. 185, pp. 1376–1391, 2022.

[9] T. Kataray, B. Nitesh, B. Yarram, S. Sinha, E. Cuce, S. Shaik, P. Vigneshwaran, and A. Roy, "Integration of smart grid with renewable energy sources: Opportunities and challenges – A comprehensive review," *Sustainable Energy Technologies and Assessments*, vol. 58, 2023.

[10] IEEE Standards Association [IEEE SA], "What are Standards? Why are They Important?," 2021. Accessed on 10 June 2025. https://standards.ieee.org/beyond-standards/what-are-standards-why-are-they-important/.

[11] AMPECO, "Demand Response," n.d. Accessed on 4 March 2025. https://www.ampeco.com/ev-charging-glossary/demand-response/.

[12] N. G. Paterakis, O. Erdinç, and J. P. Catalão, "An overview of Demand Response: Key-elements and international experience," *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 871–891, 2017.

[13] A. Elkasrawy and B. Venkatesh, "Demand Response Cooperative and Demand Charge," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4167–4175, 2020.

[14] G. G. Zanvettor, M. Fochesato, M. Casini, J. Lygeros, and A. Vicino, "A stochastic approach for EV charging stations in demand response programs," *Applied Energy*, vol. 373, 2024.

[15] International Energy Agency [IEA], "Demand Response," 2023. Accessed on 4 March 2025.https://www.iea.org/energy-system/energy-efficiency-and-demand/demand-response.

[16] J. Meiers and G. Frey, "A Case Study of the Use of Smart EV Charging for Peak Shaving in Local Area Grids," *Energies*, vol. 17, no. 1, 2024.

[17] J. W. Lee, M. H. S. M. Haram, G. Ramasamy, S. P. Thiagarajah, E. E. Ngu, and Y. H. Lee, "Technical feasibility and economics of repurposed electric vehicles batteries for power peak shaving," *Journal of Energy Storage*, vol. 40, 2021.

[18] A. Wallberg, C. Flygare, R. Waters, and V. Castellucci, "Peak Shaving for Electric Vehicle Charging Infrastructure—A Case Study in a Parking Garage in Uppsala, Sweden," *World Electric Vehicle Journal*, vol. 13, no. 8, 2022.

[19] K. H. Chua, H. Lih Bong, Y. S. Lim, J. Wong, and L. Wang, "The State-of-the-Arts of Peak Shaving Technologies: A Review," in *2020 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*, pp. 162–166, 2020.

[20] S. S. Ravi and M. Aziz, "Utilization of Electric Vehicles for Vehicle-to-Grid Services: Progress and Perspectives," *Energies*, vol. 15, no. 2, 2022.

[21] M. Uddin, M. F. Romlie, M. F. Abdullah, S. Abd Halim, A. H. Abu Bakar, and T. Chia Kwang, "A review on peak load shaving strategies," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 3323–3332, 2018.

[22] X. Li, Z. Wang, L. Zhang, Z. Huang, F. Guo, A. Sivakumar, and D. U. Sauer, "Electric vehicle charging flexibility assessment for load shifting based on real-world charging pattern identification," *eTransportation*, vol. 23, 2025.

[23] E. Tsybina, C. Winstead, B. Ollis, M. Olama, and T. Kuruganti, "Demand response for frequency regulation: Research continuity and knowledge gaps," *Renewable and Sustainable Energy Reviews*, vol. 207, 2025.

[24] A. Mohamed, R. Rigo-Mariani, V. Debusschere, and L. Pin, "Stacked revenues for energy storage participating in energy and reserve markets with an optimal frequency regulation modeling," *Applied Energy*, vol. 350, 2023.

[25] S. Cai and R. Matsuhashi, "Model Predictive Control for EV Aggregators Participating in System Frequency Regulation Market," *IEEE Access*, vol. 9, pp. 80763–80771, 2021.

[26] A. Mohsenzadeh, C. Pang, and L. Yang, "Transmission congestion management considering EV parking lots and demand response programmes," *International Journal of Ambient Energy*, vol. 42, no. 15, pp. 1727–1731, 2020.

[27] M. A. Fotouhi Ghazvini, G. Lipari, M. Pau, F. Ponci, A. Monti, J. Soares, R. Castro, and Z. Vale, "Congestion management in active distribution networks through demand response implementation," *Sustainable Energy, Grids and Networks*, vol. 17, 2019.

[28] N. Brinkel, T. AlSkaif, and W. van Sark, "Grid congestion mitigation in the era of shared electric vehicles," *Journal of Energy Storage*, vol. 48, 2022.

[29] A. Stawska, N. Romero, M. de Weerdt, and R. Verzijlbergh, "Demand response: For congestion management or for grid balancing?," *Energy Policy*, vol. 148, 2021.

[30] A. W. Thompson and Y. Perez, "Vehicle-to-Everything (V2X) energy services, value streams, and regulatory policy implications," *Energy Policy*, vol. 137, 2020.

[31] J. Leijon, J. Santos Döhler, J. Hjalmarsson, D. Brandell, V. Castellucci, and C. Boström, "An Analysis of Vehicle-to-Grid in Sweden Using MATLAB/Simulink," *World Electric Vehicle Journal*, vol. 15, no. 4, 2024.

[32] H. Yu, S. Niu, Y. Shang, Z. Shao, Y. Jia, and L. Jian, "Electric vehicles integration and vehicle-to-grid operation in active distribution grids: A comprehensive review on power architectures, grid connection standards and typical applications," *Renewable and Sustainable Energy Reviews*, vol. 168, 2022.

[33] C. Hecht, J. Figgener, and D. U. Sauer, *Protocols and Interfaces for EV Charging*, pp. 77–89. Cham: Springer Nature Switzerland, 2024.

[34] Svenska Kraftnät, "Sveriges elnät," 2024. Accessed on 19 February 2025. https://www.svk.se/om-kraftsystemet/oversikt-av-kraftsystemet/sveriges-elnat/.

[35] H. Bobadilla Robles, A. Emanuelsson, A. Aden, and C. J. Wallnerström, "Monitoring voltage quality in Sweden," in *27th International Conference on Electricity Distribution (CIRED 2023)*, vol. 2023, pp. 1189–1193, 2023.

[36] P. Johansson, M. Vendel, and C. Nuur, "Integrating distributed energy resources in electricity distribution systems: An explorative study of challenges facing DSOs in Sweden," *Utilities Policy*, vol. 67, 2020.

[37] M. Söderberg and M. Vesterberg, "Scale properties and efficient network structures in the Swedish electricity distribution market," *Journal of Regulatory Economics*, 2025.

[38] gridX, "E-Mobility Service Provider," 2024. Accessed on 19 February 2025. https://www.gridx.ai/knowledge/e-mobility-service-provider.

[39] N. Kauppi, "What Is the Difference Between e-Mobility Service Provider (eMSP) and Charge Point Operator (CPO)," 2023. Accessed on 19 February 2025. https://chargedrive.com/articles/what-is-the-difference-between-e-mobility-service-provider-emsp-and-charge-point-operator-cpo.

[40] Evcstar, "Top Charging Networks in Sweden," n.d. Accessed on 20 February 2025. https://evcstar.com/ev-charging-business/top-charging-networks-in-the-sweden/.

[41] "Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU," *OJ*, vol. L 158, p. 125, 2019-06-14.

[42] S. Kerscher and P. Arboleya, "The key role of aggregators in the energy transition under the latest European regulatory framework," *International Journal of Electrical Power & Energy Systems*, vol. 134, 2022.

[43] M. Kubli and P. Canzi, "Business strategies for flexibility aggregators to steer clear of being "too small to bid"," *Renewable and Sustainable Energy Reviews*, vol. 143, 2021.

[44] Energimarknadsinspektionen [EI], "Årlig sammanställning av hinder för efterfrågeflexibilitet — sammanfattning av workshop 20 oktober 2021," 2021. Accessed on 19 February 2025. ttps://ei.se/om-oss/nyheter/2021/2021-11-03-arlig-sammanstallning-av-hinder-for-efterfrageflexibilitet—sammanfattning-av-workshop-20-oktober-2021.

[45] CheckWatt, "Batterirapporten 2024," 2024. Accessed on 20 February 2025. https://checkwatt.se/pdf/BatteriRapporten2024.pdf.

[46] gridX, "Charge Point Operator," 2024. Accessed on 19 February 2025. https://www.gridx.ai/knowledge/charge-point-operator.

[47] Å. L. Sørensen, K. B. Lindberg, I. Sartori, and I. Andresen, "Analysis of residential EV energy flexibility potential based on real-world charging reports and smart meter data," *Energy and Buildings*, vol. 241, 2021.

[48] K. Sevdari, L. Calearo, P. B. Andersen, and M. Marinelli, "Ancillary services and electric vehicles: An overview from charging clusters and chargers technology perspectives," *Renewable and Sustainable Energy Reviews*, vol. 167, 2022.

[49] Statista, "Leading electric vehicle charging point operators in Sweden as of January 2024, by market share of the total charging point volume," 2025. Accessed on 20 February 2025. https://www.statista.com/statistics/1553096/sweden-leading-ev-charger-operators-by-volume-share/.

[50] B. Oumaima, E. F. Hassan, S. El Jeilani, H. Halima, and K. Abdelaziz, "Integration of Communication Between EV and EVSE Based on ISO 15118 with an OBD-II System," in *Automatic Control and Emerging Technologies* (H. El Fadil and W. Zhang, eds.), (Singapore), pp. 581–588, Springer Nature Singapore, 2024.

[51] A. Paraskevas, D. Aletras, A. Chrysopoulos, A. Marinopoulos, and D. I. Doukas, "Optimal Management for EV Charging Stations: A Win–Win Strategy for Different Stakeholders Using Constrained Deep Q-Learning," *Energies*, vol. 15, no. 7, 2022.

[52] S. Lu, B. Han, F. Xue, L. Jiang, and K. Qian, "Multi-Objective Optimization of EV Charging and Discharging for Different Stakeholders," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 6, pp. 2301–2308, 2023.

[53] K. Nelson, J. Mohammadi, Y. Chen, E. Blasch, A. Aved, D. Ferris, E. A. Cruz, and P. Morrone, "Electric Vehicle Aggregation Review: Benefits and Vulnerabilities of Managing a Growing Fleet," in *2024 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, 2024.

[54] AMPECO, "OpenADR," n.d. Accessed on 29 January 2025. https://www.ampeco.com/us/ev-charging-glossary/openadr/.

[55] OpenADR Alliance, "Overview," n.d. Accessed on 29 January 2025. https://www.openadr.org/overview.

[56] N. Uribe-Pérez, A. Gonzalez-Garrido, A. Gallarreta, D. Justel, M. González-Pérez, J. González-Ramos, A. Arrizabalaga, F. J. Asensio, and P. Bidaguren, "Communications and Data Science for the Success of Vehicle-to-Grid Technologies: Current State and Future Trends," *Electronics*, vol. 13, no. 10, 2024.

[57] P. Klapwijk and L. Driessen, "EV Related Protocol Study," 2017. Accessed on 30 January 2025. https://www.researchgate.net/publication/317265159_EV_Related_Protocol_Study.

[58] A. Augello, P. Gallo, E. R. Sanseverino, G. Sciumè, and M. Tornatore, "A Coexistence Analysis of Blockchain, SCADA Systems, and OpenADR for Energy Services Provision," *IEEE Access*, vol. 10, pp. 99088–99101, 2022.

[59] Z. Sultana, C. H. Basha, and M. M. Irfan, "Communication Protocols for Electric Vehicles: A Comprehensive Analysis," in *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, pp. 127–133, 2024.

[60] Łukasz Kulczyński, "A Developer Perspective Coping with a variety of standards - OpenADR, OCPP, etc.," in *2nd OpenADR++ Users Conference Europe*, 2024.

[61] Internet Engineering Task Force [IETF], "Extensible Messaging and Presence Protocol (XMPP): Core," 2011. Accessed on 30 January 2025. https://www.ietf.org/rfc/rfc6120.txt.

[62] OpenADR Alliance, "OpenADR - New Frontiers in EV Charging," 2024. Accessed on 30 January 2025. https://www.openadr.org/assets/OpenADR

[63] OpenADR Alliance, "OpenADR Alliance Annual Report," 2024. Accessed on 3 February 2025. https://openadr.memberclicks.net/assets/OpenADR_Annual_Report_2024.pdf.

[64] AMPECO, "OCPP (Open Charge Point Protocol)," n.d. Accessed on 3 February 2025. https://www.ampeco.com/us/ev-charging-glossary/ocpp-open-charge-point-protocol/.

[65] A. Hoekstra, R. Bienert, A. Wargers, H. Singh, and P. Voskuilen, "Using OpenADR with OCPP," 2016. Accessed on 3 February 2025. https://www.openadr.org/assets/using%20openadr%20with%20ocpp.pdf.

[66] S. Kirchner and H. Ruiz, "OCPP Interoperability: Democratized Future of Charging," in *36th International Electric Vehicle Symposium and Exhibition (EVS36)*, 2023.

[67] S. Rinaldi, M. Pasetti, A. Flammini, P. Ferrari, E. Sisinni, A. Depari, and G. Maternini, "Analysis of Communication Requirements for Enabling the Coordinate Charge of Sustainable Multimodal Electric Vehicle Supply Equipment," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1–6, 2021.

[68] OpenADR Alliance, "Purpose of Vehicle-to-Grid (V2G) Integration - and how it relates to OpenADR," n.d. Accessed on 4 February 2025. https://www.openadr.org/openadr-v2g-and-der.

[69] IEEE 2030.5-2023, "IEEE Standard for Smart Energy Profile Application Protocol," Standard, Institute of Electrical and Electronics Engineers, 2024.

[70] J. Mater, S. Kang, and R. Simpson, "IEC 61850 and IEEE 2030.5: A Comparison of 2 Key Standards for DER Integration: An Update," White Paper, QualityLogic, 2019.

[71] J. Kim, K. Park, B. Ahn, J. Choi, Y. Noh, D. Won, and T. Kim, "Real-Time Hardware-in-the-Loop Distributed Energy Resources System Testbed using IEEE 2030.5 Standard," in *2021 IEEE PES Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 1–5, 2021.

[72] D. Passos, C. de Sousa, R. C. Gomes, D. F. Assis, F. G. O. Passos, and C. Albuquerque, "A Tutorial and Security Overview on the IEEE 2030.5-2018 Standard," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2025.

[73] Greenlots, "Open vs. Closed Charging Stations: Advantages and Disadvantages," Open Standards–Based Networks White Papers, Greenlots, 2018.

[74] DNV and Open Charge Alliance, "OCPP & IEC 61850: a winning team," White Paper 3107, Open Charge Alliance, 2023.

[75] N. Galkin, C.-W. Yang, L. Nordström, and V. Vyatkin, "Prototyping Multi-Protocol Communication to enable semantic interoperability for Demand response Services," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 15–20, 2021.

[76] Open Charge Alliance (OCA), "OCPP 2.1 is now available!," 2025. Accessed on 5 February 2025. https://openchargealliance.org/ocpp-2-1-is-now-available/.

[77] Open Charge Alliance (OCA), "Certification OCPP 1.6," n.d. Accessed on 6 February 2025. https://openchargealliance.org/certificationocpp/certification-ocpp-1-6/.

[78] Open Charge Alliance (OCA), "Certification OCPP 2.0.1," n.d. Accessed on 6 February 2025. https://openchargealliance.org/certificationocpp/certification-ocpp-2-0-1/.

[79] J. Vujasinović, Z. Čiča, and G. Savić, "Possibility of Applying the EEBUS to the IoT Based Renewable Energy Sources Powered Station for Electric Vehicles Charging," in *2024 32nd Telecommunications Forum (TELFOR)*, pp. 1–4, IEEE, 2024.

[80] gridX, "EEBus," 2024. Accessed on 6 February 2025. https://www.gridx.ai/knowledge/eebus-universal-communication-protocol-of-the-energy-world.

[81] EEBUS, "About EEBUS," n.d. Accessed on 6 February 2025. https://www.eebus.org/about-us/.

[82] X. Wang, Y. Liu, and Y. SUN, "EEbus: Interoperability Technology Based on Family Energy Management System," *DEStech Transactions on Engineering and Technology Research*, 2018.

[83] J. Zuber, R. Bienert, R. Böhm, R.-I. Prümm, and A. Schwackenberg, "Combining OpenADR and EEBUS for Energy Control," White Paper, EEBus Initiative e.V, 2022.

[84] AMPECO, "ISO 15118," n.d. Accessed on 11 February 2025. https://www.ampeco.com/us/ev-charging-glossary/iso-15118/.

[85] M. Mültin, "ISO 15118 as the Enabler of Vehicle-to-Grid Applications," in *2018 International Conference of Electrical and Electronic Technologies for Automotive*, pp. 1–6, 2018.

[86] S. Madhusudhanan and P. Sivraj, "Development of Communication Simulator for Electric Vehicle Charging following ISO 15118," in *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, pp. 1–6, 2022.

[87] ISO 15118-20:2022, "Road vehicles — Vehicle to grid communication interface - Part 20: 2nd generation network layer and application layer requirements," Standard, International Organization for Standardization, 2022.

[88] J. Sautraeau and A. Walle, "Update protocol definition," Project deliverable D2.3, SCALE, 2024.

[89] J. Meersmans, "Analysis of hard- and software requirements," Project deliverable D1.5, SCALE, 2023.

[90] G. L. Maia, C. C. L. Santos, P. R. M. Nunes, J. F. C. Castro, D. C. Marques, L. H. A. D. Medeiros, L. R. Limongi, M. E. C. Brito, N. K. L. Dantas, A. V. M. L. Filho, A. L. Fernandes, J. Chai, and C. Zhang, "Ev smart-charging strategy for power management in distribution grid with high penetration of distributed generation," *Energies*, vol. 17, no. 21, 2024.

[91] N. Das, A. Haque, H. Zaman, S. Morsalin, and S. Islam, "Exploring the Potential Application of IEC 61850 to Enable Energy Interconnectivity in Smart Grid Systems," *IEEE Access*, vol. 12, pp. 56910–56923, 2024.

[92] P. Rai, A. Mishra, and A. Lal, "Smart Grid and IEC 61850," in *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–6, 2021.

[93] T. S. Ustun, S. M. S. Hussain, M. H. Syed, and P. Dambrauskas, "IEC-61850-Based Communication for Integrated EV Management in Power Systems with Renewable Penetration," *Energies*, vol. 14, no. 9, 2021.

[94] E. Ebeid, S. Rotger-Griful, S. A. Mikkelsen, and R. H. Jacobsen, "A methodology to evaluate demand response communication protocols for the Smart Grid," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 2012–2017, 2015.

[95] Energiföretagen, "Industry recommendation - Conditional Grid Connections," 2023. Accessed on 20 March 2025. https://www.energiforetagen.se/4a9818/globalassets/dokument/elnat/industry-recommendation-conditional-grid-connections-ver-1.0.pdf.

[96] Energiföretagen, "Supplement to industry recommendation - Conditional grid connections," 2025. Accessed on 15 May 2025. https://www.energiforetagen.se/498a45/globalassets/dokument/elnat/supplement-to-industry-recommendation-conditional-grid-connections-ver-1.0.pdf.

[97] Energiföretagen, "Welcome to Swedenergy - Energiföretagen Sverige," 2023. Accessed on 26 March 2025. https://www.energiforetagen.se/in-english/.

[98] Anna Nordling, "Sveriges framtida elnät - En delrapport," Interim Report, Kungl. Ingenjörsvetenskapsakademien (IVA), 2023.

[99] E.ON, "Nätutvecklingsplan 2025–2034," 2024. Accessed on 14 April 2025. https://www.eon.se/content/dam/eon-se/swe-documents/swe-eon-natutvecklingsplan-2025-2034.pdf.

[100] E.ON, "Det här är E.ON," 2025. Accessed on 13 May 2025. https://www.eon.se/om-e-on/om-foeretaget.

[101] Ellevio, "Tillsammans elektrifierar vi Sverige - Ellevio Årsberättelse 2024," 2024. Accessed on 14 April 2025. https://www.ellevio.se/globalassets/content/finansiell-information/2024/ellevio_ab24_swe_for-webb.pdf.

[102] Vattenfall, "Breaking society free from fossil fuels - Annual and Substainability Report 2024," 2024. Accessed on 14 April 2025. https://group.vattenfall.com/globalassets/com/sustainability/vattenfall-annual-and-sustainability-report-2024.pdf.

[103] OpenADR Alliance, *OpenADR 3.0 - OpenADR 3.0.1 User Guide (non-normative)*, 2024. Revision Number: 3.0.1.

[104] OpenADR Alliance, *OpenADR 3.0 - OpenADR 3.0.1 Definitions*, 2024. Revision Number: 3.0.1.