Technical University of Denmark

Anders Ørskov Christensen, S052548
Jonas Falck Frederiksen, S052608

Distributed Decentralised EV

Architecture and Requirements Specification

Master Thesis, October 2011

DTU Informatics
Department of Informatics and Mathematical Modelling

Anders Ørskov Christensen, S052548
Jonas Falck Frederiksen, S052608

Distributed Decentralised EV

Architecture and Requirements Specification

Master Thesis, October 2011

Distributed Decentralised EV,Architecture and Requirements Specification

This report was prepared by
Anders Ørskov Christensen, S052548
Jonas Falck Frederiksen, S052608

Supervisors
Bjarne Poulsen, DTU-IMM
Anders Bro Pedersen, DTU-CET

Department of Informatics and Mathematitical Modelling
Centre for Software Engineering (CSE)
Technical University of Denmark
Elektrovej building 322
DK-2800 Kgs. Lyngby
Denmark

www.imm.dtu.dk
Tel: (+45) 45 25 35 00
Fax: (+45) 45 88 61 11
E-mail: @imm.dtu.dk

Release date:	Date published
Category:	1 (public)
Edition:	First
Comments:	This report is part of the requirements to achieve the Master of Science in Engineering (M.Sc.Eng.) at the Technical University of Denmark. This report represents 30 ECTS points.
Rights:	©DTU, 2011

Preface

This thesis was conducted for the Institute of Informatics and Mathematical Modelling (IMM), at the Technical University of Denmark as a part of the final exam for acquiring the M.Sc. degree in Computer Science and Engineering.

The thesis investigates how the centralised control of charging schedules, in an electric vehicle population, can become decentralised and to what extent. Existing centralised solutions are analysed and the thesis also contains proof-of-concept through an implemented prototype.

The two authors have contributed equally to the entire thesis and are in that respect equally responsible for all sections. Finally, we would like to thank Bjarne Poulsen, Associate Professor at IMM, and PhD student Anders Bro Pedersen for supervising this thesis.

Kongens Lyngby, October 2011

Jonas Falck Frederiksen and Anders Ørskov Christensen

Summary

International events continue to make an impact on the price of crude oil, which causes the price of having a traditional car to rise. At the same time there has been an increasing focus on how to optimise the use of renewable energy resources in every day life. Electric vehicles (EVs) have been developing at a slow pace until a few years back. Now, due to the aforementioned events, more research groups and industry stakeholders are getting involved. Governmental initiatives have also been taken to help speed up the process on maturing EV technology and infrastructure. However, there is still a long way to go and the EV market is characterised by a lack of standards and unity of where to focus the development.

It is becoming evident that centralised control of a large EV population will not be entirely possible. This thesis targets to help the process of decentralising the control by examining to what extent the control can decentralised. Two major problems related to decentralised has been investigated. Furthermore, work has been focused on moving the charging schedule responsibility to the EV, while having a modular and secure architecture.

To ensure the architecture is designed properly a prototype has been implemented and tested in two case studies as proof-of-concept. The results indicates that the overall design constitutes a suitable architecture for a decentralised EV. However, much work still needs to be carried out both in terms of implementing more functionality, but also in terms of more research.

Dansk Resumé

Internationale begivenheder fortsætter med af få prisen på olie til at stige, hvormed prisen for at bruge en traditionel bil også stiger. Samtidig er der et enormt fokus på at udnytte vedvarende energi. Elbiler har ikke vundet meget terræn, men i de seneste år er der sket meget. Flere forskningsgrupper og industrielle interessenter viser interesse for at udvikle teknologi, der kan bruges i en elbil. Statslige initiaver er taget for at modne elbil-relateret teknologi og dertil hørende infrastruktur. Der mangler dog stadig standardisering og der er endnu lang vej for at en storstilet udrulning kan finde sted.

Central kontrol af mange elbiler har vist sig at have begrænsninger. Dette speciale ønsker at bidrage til at kontrollen kan blive decentraliseret ud i elbilerne. I dette tilfælde ved at designe en løsning, der kan generere et ladeskema, men samtidig har en modulær og sikker arkitektur.

En prototype er blevet udviklet og testet i to case studies, som et proof-of-concept. Resultaterne er lovende og arkitekturen kan bruges som grundlag for fremtidig arbejde med at implementere og teste yderligere funktionalitet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter will take a closer look at the motivation behind this thesis and the field of distributed energy resources and electric vehicles. Additionally, the background for the thesis is discussed and what related work and investigations have already been made. Last in this chapter the vision and description of the project are formulated, along with the problem statements to be investigated as well as the structure of the document.

1.1 Motivation & Background

Every day the demand for energy in the modern society increases. The tendency in well developed countries is that the demand keeps rising, but perhaps more importantly as less developed countries grow at a rapid pace the demand for energy will rise accordingly. Especially India and China have the potential to raise the global energy demand.

Using today's traditional energy sources, such as fossil fuels there are several potential problems waiting to be solved. Fossil fuels are a limited resource and as the global society consumes more than is being created by slow natural processes. Many countries are dependent on a constant delivery of energy from fossil fuels, hence many countries rely on the countries where the energy deposits are located. As the demand for energy rises the price follows and another important factor, is that the flow from the oil producing countries must be kept constant or controlled to keep the price stable. However, a recurring problem in history has been huge fluctuations in prices, most notably was perhaps the embargo of USA and Western Europe in 1973 by OPEC causing a huge impact on the economies in the most dependent countries. The price on crude oil has continued to fluctuate and recent revolutions and revolution attempts in the Middle East and in North African countries have once again caused a higher price on oil, due to fear of instability in the affected countries.

Another popular energy resource is nuclear power, which gained success following the high oil prices in the 70's and 80's. However, the general public support has suffered from the fear of a nuclear meltdown, and the Tjernobyl incident in 1986 was an example that caused a general opposition towards nuclear power. The most recent incident, which already has had a tremendous impact on the local area, but also indirectly on the energy strategy of countries worldwide, was the meltdown of the Japanese Fukushima nuclear power plant in March 2011, due to an earthquake and the following devastating tsunami. After this catastrophe, the German government decided to close seven old nuclear power plants. Recently they decided to keep them closed and phase out all of its nuclear power plants by 2022 [12].

The environmental consideration is also a factor and the role that the consumption of fossil fuels plays, in the global warming debate, has been vigorously discussed and been a dominating political issue for years. The only viable solution seems to be renewable energy to solve the problem with limited resources on Earth, pollution and energy supply dependency, as it can be deployed most places. Many countries therefore invest a lot of resources in this area of research, hoping to avoid future energy crises. A growing number of countries are investing in energy supply solutions, like wind and water turbines, to get more renewable energy resources. Another research topic, is how to optimize the current power grid, so it is better utilised and can support more functions or resources. One specific research topic is how the power grid can support distributed energy resources (DER) and integrate new types, as they are developed. A relatively new type of DER, is the electric vehicle (EV) and if the number of EVs continue to grow they will form a huge combined energy resource, which can support the power grid. However, barriers exist that hinder an EV to communicate directly with the power grid. Recent research projects have been focusing on overcoming these barriers by joining a large number of EVs in one common solution - a virtual power plant (VPP) with centralised control of the EV population and the purchase of power from the distributors. Though, when relying on renewable energy sources, one of the challenges is how the entire power grid responds to periodic fluctuations in power production and consumption. Using Denmark as an example, the power from wind turbines can not easily be adjusted to fit peak demand, as this would be unprofitable with a huge production of surplus power. The fundamental problem is that it is still not economically feasible to store energy on a large scale. EVs may provide part of the solution to meet the needed supply during peak periods. The potential power storage that will become available as EVs become common is tremendous, but also depends on the EVs' ability to feed power back to the grid (V2G). The VPP acts as an aggregator between the EVs and the grid, and it might be used the other way around. In the research project EDISON, the focus has been on, among other things, the electric vehicle's storage technology and suitable architectures. The background and motivation for this thesis are to investigate how this control and architecture can be decentralised so that calculations are made locally in each EV and thereby leveraging the large scale VPPs, while maintaining the flexibility, reliability and security.

1.2 Vision

The vision of this master thesis is to investigate if it is possible for an EV to be fully decentralised and work with the smart grid of the power network. An analysis of the functional and non-functional requirements of such an EV will be carried out and the design must fulfill the requirements. Finally, two case studies will function as proof-of-concept of the implemented prototype.

1.3 Project Description

This thesis will contain a survey of other projects currently in development around the world and what areas of the EV the researchers are concerned with. The outcome will be a picture of what is the state-of-the-art within the area of EVs, which will be succeeded by an analysis of the VPP developed at DTU and the EDISONVPP from the EDISON project. where the focus will be on which functional features can be transferred and used in a decentralised EV. Following the analysis of the VPP a closer look must be taken at the EV vs. VPP and the ability of the EV to be able to operate in a decentralised way, as an independent DER in the smart grid. This leads to a proposal of a suitable architecture for this kind of solution and a closer look at what challenges are created, when the control is decentralised. These surveys and analyses will result in the functional and non-functional requirements of the EV and will conclude with a list of use cases to give a complete overview of system functions in EV design, in this thesis and of which actors can interact with the system.

Based on the functional requirements identified in the analysis, the design of the EV architecture will be composed. One of the main topics, which has to be taken into account, when designing the architecture, is the ability of the

EV to receive the market price and to receive it in a secure way. This leads to a survey on how the EV can communicate with the smart grid, and which kind of technology is needed to allow the EV to receive the price signal in a secure way.

To make a proof-of-concept, a prototype of the designed architecture for the EV is implemented. The idea behind the prototype is to prove the basic functionalities of the solution. Therefore, not all of the functional requirements identified in the analysis, will be implemented, but will instead be prepared for future adoption. Besides the prototype implementation, two case studies will be conducted to test that the solution meets the determined requirements.

The following list summarises an overview of the problem statements for the thesis:

1. Survey which other projects are being developed, and what is state-of-the-art
2. Analyse the VPP concept and the developed EDISON VPP solution
3. Analyse the possibility for an EV to become a fully decentralised DER
4. Identify VPP features, which can be used in a decentralised EV
5. Identify the functional and non-functional requirements of an EV
6. Based on the identified requirements, design the architecture of an EV
7. Analyse needed security measures to protect the integrity and authenticity of the received price signal
8. Implement a prototype of an EV solution
9. Create two case studies to prove the successfulness of the solution according to the objectives of the thesis.

1.4 Report Structure

The structure of the project report is derived from the project description in 1.3 as it is clear that a number of tasks must be carried out in order to fulfill the goals of the vision in 1.2.

Chapter 1 - Introduction
This chapter contains the introduction of the goals and objectives of this thesis. The chapter begins with a fundamental picture of the motivation and

background, which precedes the prerequisites and domain specific knowledge for this particular project. With the knowledge gained within the area of the smart grid and EVs, the vision for this thesis is made. The project description helps to act as a scope with associated problem statements.

Chapter 2 - Analysis & Requirements
In this chapter a closer look is taken at other projects, which are concerned within the research area of EVs. The survey is followed by an analysis of the VPP from the EDISON project. The analysis must clarify what elements that can be used from the VPP in the EV. It will then be possible to analyse the possibility of the EV to be a completely decentralised DER, or if it still needs to have some kind of dependency to the VPP. All the surveys and analyses must, at the end, lead to an identification of the functional and non-functional requirements of the EV.

Chapter 3 - Architecture Design
From the identified functional requirements, this chapter covers the design of a suitable architecture for a decentralised EV. This involves an analysis of what security measures are needed to ensure the integrity of the price signal received by the EV.

Chapter 4 - Prototype Development
After the EV solution is designed, the chapter covers the technical concepts on how the proof-of-concept prototype of a decentralised EV has been implemented.

Chapter 5 - Case Studies
This chapter presents two case studies conducted to prove that the solution lives up to the objectives stated in section 1.2. The first case study tests the basics of the solution; for instance that the architecture of the solution is suitable, that it can receive a price and by using its self generated charging schedule it can start charging the battery. The second case study concerns how an actual implementation would be carried out, while preserving all the requirements and the security of the solution.

Chapter 6 - Conclusion & Future Work
Finally, after the developed solution is tested against two case studies, the chapter present an overall conclusion on the goals for this thesis. Furthermore, tasks and ideas for future work and development will be discussed.

1.5 Development Process

When choosing the appropriate development model, it is important that it fits the project to be developed, as well as the project group. Especially for software development, where a new software product is built from scratch, the end result will inevitably differ from what the customer had in mind in the first place. Therefore, iterations can help minimise the gap created between the product, as the customer imagined it, and the actual outcome. It can also help to minimise problems with new technologies as the problems are discovered and dealt with when they arise, and not as it all comes together in the end. An agile and iterative model is therefore needed.

Unified Process

Unified Process is a customisable framework well suited for software development and thoroughly documented in the form of Rational Unified Process (RUP) introduced by IBM. The most lightweight and simple version of RUP is the OpenUP. It is targeted at small teams and relatively short term projects. One of the benefits of using OpenUP, is that it is well suited for handling small contributions, also known as micro-increments. In this way, every few hours, or by the end of each day, developers can share their progress and this encourages productivity and teamwork. It preserves the very basics of RUP with iterative development, use cases and the architecture-centric approach. It is the chosen approach for this project as agile and iterative development is needed [3].

Chapter 2

Analysis & Requirements

Research groups around the world are looking into different solutions within the area of electric vehicles. Thus, this chapter will be looking closely at what kind of solutions that have been developed in connection with electric vehicles and what is state-of-the-art within the field. This will be followed up by an analysis of the Virtual Power Plant and how this is able to adopt distributed energy resources to the power grid.

The analysis of the VPP concept is followed by an analysis of the VPP developed in the Danish and international research project EDISON. The analysis continues with a short discussion of arguments for and against decentralising functionality of the VPP. The arguments against decentralising are then sought to be solved. The results lead to a section about what is necessary for an EV to become a decentralised DER, and which kind of architectures are suitable.

The next section concerns the identification of the needed elements in a solution, where the EVs are decentralised. It concludes with a clarification of which requirements are necessary for those elements to function and with a model of the domain to be designed.

Finally, the findings of the analysis are summed up in the conclusion of the chapter.

2.1 Research & Development - State-of-the-art

In this section a survey will be made of what is the latest news within the field of electric vehicles. Both in terms of players on the market, and tendencies in business strategies, but also within the most recent research projects.

2.1.1 PricewaterhouseCoopers study

In a recently published study performed by the consultancy firm, PricewaterhouseCoopers (PwC)[13], they asked more than 200 executives with very different backgrounds to identify major concerns and strategies to get a future outlook on EVs. The first major concern they present is a lack of investment in modernising the support for the new infrastructure needed to support EVs. The second problem is the lack of power capacity in peak periods. Furthermore, about one third of the interviewees felt that the financial burden should be shared among the government, local municipalities and private investors. The incentive for companies to continue to perform research and pilot projects, as well as producing the cars, is their financial gain.

As long as the infrastructure is not in place, this incentive will remain small. However, infrastructure projects are being escalated, but require support from governments and local municipalities. These involved parties have reasons to support the companies and should provide incentives, so that they make investments attractive. Also, the study showed that the companies need to provide an alternative to the customer that is greener, but first and foremost cheaper than the existing alternatives. It is likely that the increasing instability in oil producing countries and the rise in demand will help make this happen. Finally, about two thirds recognise hybrid cars as the most appealing overall solution to the customer. The current limitations of EVs are probably the cause of the last statement as EVs have limited range and the infrastructure in general is still very limited.

In the following section, we will take a closer look at some current players on the market for EVs and some of the collaborations made. We will also be looking into the companies who have built a business model around providing the charging facilities, which have the potential to become a huge business in itself. Finally, a brand new fuel technology, invented by MIT, will be presented.

2.1.2 Collaborations

As IT in general is becoming more integrated in everyday life, it is only a natural development that the integration also happens within the car industry. Whereas, the traditional car manufacturers are experts within the traditional way of producing and selling cars, new technology possibilities open up for new ventures with other companies.

Google and Ford

Recently Ford and Google announced a collaboration to integrate Google Analytics and computing components in Ford's hybrid cars. Based on a driver's profile and his/her stored statistical data about usage the of the car, Google and Ford want to predict and optimize performance of the car on the given route, or even make suggestions to the driver, about how to optimise fuel consumption etc. In the first release it will not, however, be too big a change, as the changes will be transparent to the driver. However, the concept opens up for a number of interesting scenarios. The car will always be online and connected to the cloud and therefore able to react to the latest events or road accidents, so that the driver is redirected. Another scenario where this technology comes in handy, is if EVs get special priority in traffic. For instance if some lanes are open to electric cars only during rush hour as a counterpart to the carpooling/ridesharing systems known from large cities around the world. A hybrid car with the right means of prediction and detection could also save the electric energy, so that those reserved lanes could be used, or the car be set to use battery to avoid pollution within the city limits [17].

Mitsubishi and PSA Peugeot Citroën

Another and perhaps more traditional venture was announced in September 2009 between Mitsubishi and PSA Peugeot Citroën. The benefits of working together are that the companies share the risks of entering into the new market, and can thus save costs on the research and time to market, as they draw on experiences from one another. Peugeot, Mitsubishi and Citroën have launched their 100% electrical car models iOn, iMiev and C-Zero repectively, also known as the triplets, as they are almost identical. The chance the companies take with these launches clearly illustrates the positive expectancies towards the market for EVs.

EVs have been launched before, but never in this magnitude. Besides the aforementioned collaborations, Toyota and Daimler AG have been working together on the Tesla project and Renault and Nissan with infrastructure provider, Better Place. The rise in collaborations indicate that the companies are moving in the same direction as the conclusion drawn by PricewaterhouseCoopers. Not only are they collaborating within their own industry, but also across different areas such as the collaboration between Google and Ford, to discover new and enhance existing possibilities with EVs.

Moving beyond their collaboration in car making Peugeot, Mitsubishi and Citroën also collaborate with the Danish infrastructure provider ChoosEV. ChoosEV has a business model built around leasing a car with charging service subscriptions or just the subscription. The charging services include quick charging, which is the possibility to charge the car up to 80% in 20 minutes, but this is only available at special charging stations of which there is only one at the moment. Other services include a home charging station and access to the normal charging facilities provided by service providers in public or at work. Another collaboration between an infrastructure provider and car manufacturers, is between American Better Place and Renault and Nissan. Better Place offers traditional charging services and home charging, but the company also offer battery swapping, which can replace the entire battery within five minutes and in that regard is more similar to the time spent, when fueling a traditional combustion engine car at a petrol station. However, this battery exchange has been questioned by for instance Daimler AG's director of future mobility, stating that the battery swap solution is simply not feasible as different car manufacturers will have different batteries and different placement of the batteries in the car [18]. Which solution will prevail depends largely on the ongoing tests, and of the future design of EVs, and the agreement of a standardisation of quick and standard charging stations. Common for all the mentioned charging methods is that they are based on centralised charging, where the charging schedule is calculated centrally by the provider.

Strategic Platform for Innovation and Research - iPower

iPower is a project supported by the Danish Agency for Science Technology and Innovation through their Strategic Platform for Innovation and Research (SPIR). SPIR is a collaboration platform targeting to improve the research interaction between the business community and universities, both on a national and an international level. The target is to improve the competences of the fully-trained graduates and to ensure that they have the exact qualifications needed by the companies to succeed in the globalised community [7]. As a knowledge-based society it is imperative that a platform exists that forms a strong foundation between companies and universities. This is important if Denmark is to uphold its level of welfare, which was the reason that SPIR was launched in the first place.

In the first phase there were two research areas that could receive support, and it was within the food and energy industry, where iPower was chosen as the energy project. iPower aims to contribute to the future intelligent and flexible power grid, where a large portion of the energy comes from renewable energy sources and therefore has fluctuating power generation. To address this, it is necessary to make power consuming units use the power when it is produced in excess, and not when the demand rises. Requirements of such a setup includes intelligent control of all the units that also must be flexible enough so they can time their consumption of power. iPower expects to contribute to growth within the industry that will produce these intelligent decentralised power consuming, producing and controlling tools[19]. iPower is coordinated by Risø National Laboratory for Sustainable Energy at the Technical University of Denmark (Risø DTU) and has 32 partners including universities from the USA, Sweden and Ireland and large companies in Denmark, such as DONG Energy and IBM Denmark. In regard to EVs, this project is highly relevant as it addresses the limitation of the current grid and the control of DERs. However, the project is only just starting this year and finishes in 2016.

2.1.3 MIT and Semi-solid Flow Cell

A research project that might help leveraging the power grid limitations, has been published by a research team from MIT. Basically, it is concerns splitting the charging and the discharging of the battery into two different physical structures. When a car enters a charging station the discharged liquid of the battery is pumped out and replaced with liquid that has been fully charged. According to the article, this storage method is relatively inexpensive and makes it suitable for upscaling. The idea is that the charging station is connected to the grid and when there is an energy surplus available, it can be used to charge the discharged liquid held at the charging station. The potential of this discovery is huge, however, it will probably take years before the technology comes of age and the price becomes cost-competitive. In regards to EVs it is interesting, when talking about charging speeds, as this can reduce the quick charge to a time much similar to a regular petrol fueling scenario. Preserving the possibility of normal charging is also important in order to allow traditional slow charging at home. Moving a bit further, a house could have its own charging facilities with its own storage of semi-solid flow fuel, which could be charged when the electricity is cheapest. This latter scenario moves the charging responsibility to the home storage and therefore this could infer that the storage would act as a central unit. It would have a charging schedule and user preferences not just for the house as a whole, but also for individual DERs such as an EV [5].

2.2 Virtual Power Plant

All over the world DERs are becoming more used. This increasing number of DERs induce a number of requirements to the grid, less environmental pollution, different kinds of energy resources and improved energy efficiency. Additionally, the energy market has been liberalised so that the structure of the market, has made it possible for a more competitive market. With these two tendencies the number of DERs connecting to the grid rise and this brings some challenges:

Market participation The DERs are in general prohibited from entering the market because of their size.

Intermittent nature At the moment, many of the DERs are weather dependent like a photovoltaic system and wind turbines, and because of the variable contribution they provide the power grid, economic penalties are connected with the unexpected imbalances.

Stand alone Due to the ownership of the DERs, many work alone. Because of the lack of communication between the DERs they are limited to satisfy the local need, instead of providing support to the entire grid.

These challenges have induced several projects between governments, organisations and manufacturers, to research and development within the area of DERs and VPPs. The projects have resulted in a solution, where the DERs are aggregated under one VPP so that the DERs act as one visible, controllable and functional power plant, in relation to the grid.

2.2.1 Virtual Power Plant Structures

The solution of the architecture for the VPP can be divided into three different categories:

Centralised Controlled VPP (CCVPP) The CCVPP is designed as shown in figure 2.1, where the CCVPP requires the complete knowledge of all DERs connected to it. This gives the CCVPP the opportunity to meet the requirements from the grid and perform with the highest potential of operation. The disadvantage is the scalability and compatibility of the solution, because the operation is often case specific [11].

Decentralised Controlled VPP (DCVPP) As illustrated in figure 2.2 the design of the DCVPP is where the VPPs are structured in a hierarchical order, so that the local controllers coordinate a limited number of DERs and send information to the top VPP.

Fully Decentralised Control VPP (FDCVPP) The FDCVPP is designed in a way where the VPP has changed function so that it acts as a service provider, which distributes information to the DERs. The

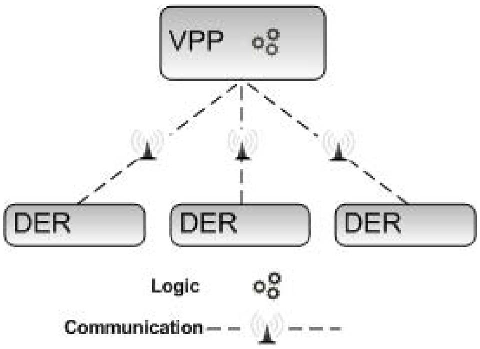


Figure 2.1: The design of a CCVPP

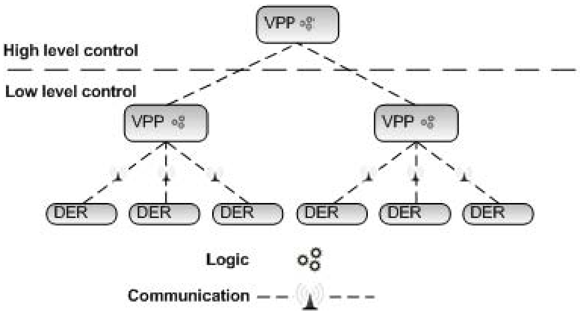


Figure 2.2: The design of a DCVPP

DERs work independently and manage the data calculations on their own, as illustrated in figure 2.3.

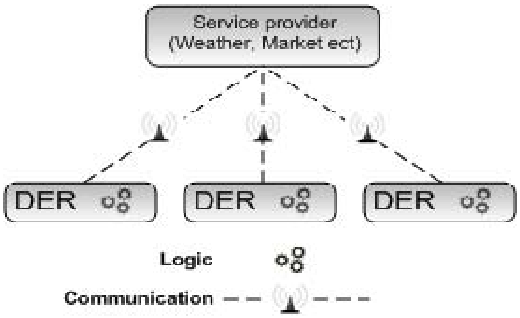


Figure 2.3: The design of a FDCVPP

From these three designs of a VPP the design of the FDCVPP gives a more dynamical plug and play design for DER units.

2.2.2 Functional Requirements of a Virtual Power Plant

Instead of having a design for each of the three VPP structures described in section 2.2.1 to support the different DER technologies, a more general concept of a VPP has been developed called Generic Virtual Power Plant (GVPP)[16]. The purpose of the GVPP is to create a usable entity, which can act as a player on the electricity market. To do that requirements for a functional-based design are developed. Figure 2.4 shows the design of a function-based GVPP, where the DER units are connected to the GVPP by an integration interface. The services in the green boxes, named GVPP infrastructure, constitute the core system. Additionally, the orange blocks are functional modules, categorised according to their purpose. The functional modules can be used in an arbitrary combination.

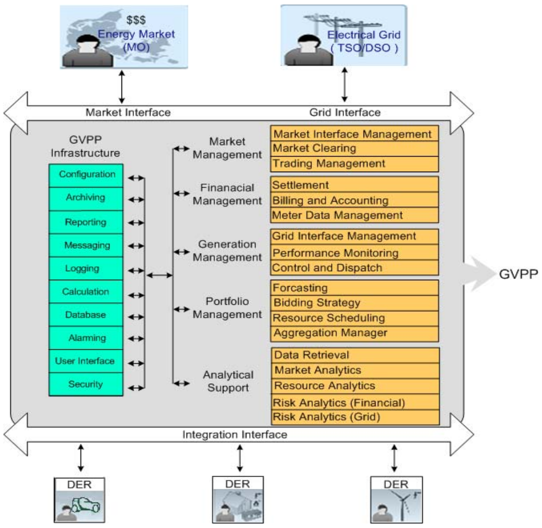


Figure 2.4: Function-based design of a GVPP

2.3 EDISON VPP

The goal of the EDISON project is to implement a VPP in a realistic environment and in that way obtain valuable experiences with the practical implementation and impact on society and the grid [6]. It is based on the idea that a properly managed fleet of EVs can be an asset to the grid, as well as to the owners of an EV. One of the research areas is within vehicle-to-grid, which is needed to efficiently ease the load on the power plants during peak demand and optimise the use of renewable energy resources - especially wind power.

The platform needed to support this feature is the main focus of the EDI-

SON project and it must consider the constraints in the grid, as well as the preferences of the EV owners. In practice this is done on the Danish island of Bornholm, which has been chosen because of its closed environment and with its considerable amount of power from wind turbines. The EDISON VPP (EVPP) is both market based and centralised in the way it controls the EVs. It supports fast charging, while considering the local electrical infrastructure, and controlled charging where the EVs are connected for a longer period of time and can be considered as possible contributors to the grid. The charging is controlled by the EVPP, which schedules the charging to avoid overloading the grid.

2.3.1 EDISON Architectures

Two architectures have been considered in the EDISON project, which are called integrated and standalone architectures. The idea behind the integrated architecture is that the EVPP should be an integrated part of an already existing player on the electric market e.g. the power generating companies. As shown in figure 2.5, the EVPP is integrated into the architecture of an already existing company, as a service that provides the ability to provide the energy schedules and act on the ancillary, balancing and services market to divide capacity of energy. Thus, the EVPP needs the knowledge of all associated EVs to calculate the amount of needed energy.

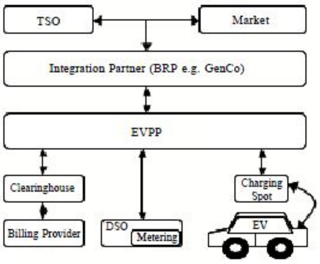


Figure 2.5: Environment diagram of an integrated EVPP

The other architecture is the standalone solution in which the EVPP is integrated into the electric market as a direct market player, acting as a Balanced Responsible Party (BRP). The idea of having the EVPP working as a BRP is that it is responsible for buying electricity with which to charge the EV, as well of supporting the stability of grid by selling ancillary services on the ancillary service market. This is shown in figure 2.6. The position of the EVPP, as a actor in both the regular market and the ancillary service market, entails that the EVPP needs the knowledge of the associated EVs and besides a more intelligent system to enable it to decide on buying and selling energy.

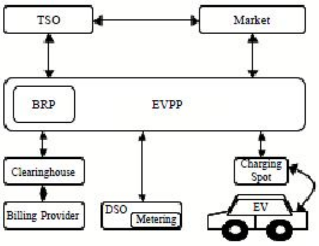


Figure 2.6: Environment diagram of an standalone EVPP

The two types of architectures for the EVPP have the same three groups of modules. These groups are the following - *control group for a single EV*, *data storage and member management*, and *aggregation and partner interface*. As shown in figure 2.7 and 2.8 the number of modules differ in the *aggregation and partner interface*. This is because of the needed interface for the standalone EVPP to interact with the additional participants, which are handled by the BRP in the integrated EVPP.

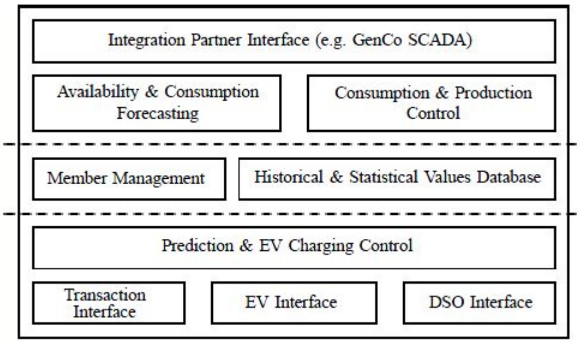


Figure 2.7: Module diagram of an integrated EVPP

As mentioned, the main difference of the two architectures is the module group called *aggregation and partner interface*. Here the integrated EVPP consists of three modules. The first is *Availability & Consumption Forecasting* module, which generates an overall EVPP energy consumption and availability based on the predictions from the EVs. The second is the *Production & Consumption Control* module, which is managing the distribution of energy in relation to the charging schedules and prediction of the single EVs. The third is the *Integration Partner Interface*, which is used by the two other modules - *Availability & Consumption Forecasting* and *Production & Consumption Control*, when they perform actions against the market.

The standalone EVPP on the other hand consists of six modules in module group *aggregation and partner interface*. As the integrated EVPP it has the same *Production & Consumption Control* module. Additionally, it has the

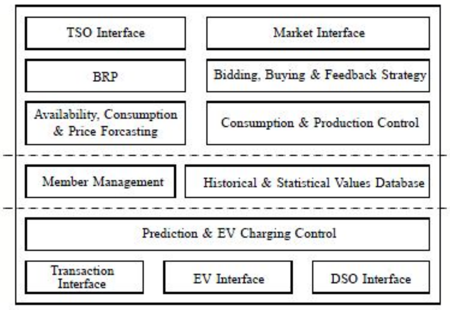


Figure 2.8: Module diagram of an standalone EVPP

Availability, Consumption & Price Forecasting module, which is similar to the *Availability & Consumption Forecasting* in the integrated architecture, though it takes care of the price prediction based on historical data that is used to buy electricity as well. The third module is the *Bidding, Buying & Feedback Strategy*, which computes the use of energy, based on the forecasts provided by the *Availability, Consumption & Price Forecasting*, to achieve the maximized profit. The fourth module is the *Market Interface* performs, based on the computation of energy use, the bidding and buying on the electricity market. Last is the *Balanced Responsible Party* and *TSO Interface* module, where the *Balanced Responsible Party* is responsible for creating and submitting charging schedules to the TSO, through the *TSO Interface*. Additionally, the *TSO Interface* is also used when receiving commands from the TSO.

The *Data Storage and Member Management* group is the same for the two types of architectures. This group of modules consists of two modules; the first is the *Historical & Statistical Values Database*, which supports the other modules with functionality to access stored data. The second module is *Member Management*, which gives the possibility to modify the default settings and requirements of the EV operator.

The last group is the *Control Group for a Single EV* that consists of four modules. The first is the EV interface, which is using transaction based communicated to get secure and reliable communication needed to schedule and plan the charging schedules. The DSO interface is the second and is used to make sure that the local grid is not overloaded. The third is the transaction interface, which ensures the consumer is paying for the battery charge. Finally, the last module is the *Prediction and EV Charging Control module*, which based on statistical data predicts when a specific EV will connect, its charging time and state of charge.

The optimal charging schedule is based on price and grid constraints. If the EV connects as predicted, the original schedule is used, otherwise it is recalculated at connection time.

Why Decentralise?

The EDISON architecture is already well developed so why consider the decentralised setup? With the centralised control of the EDISON VPP it will suffer from large scale EV populations. The computational constraint of updating millions of charging schedules and sending them out is simply too much and therefore the decentralised option seems to be the solution.

Why Not?

Centralising means control. If the responsibility for calculating the charging schedules is decentralised there is, for instance, no guarantee that the EVs will not start charging at the same time or a large group charging within a local part of the grid. Also, as the market is today, the barriers of entering it, simply cannot be overcome by a single individual EV, leaving out the benefit of having a EV fleet as a buffer during peak hours. The distribution service operators (DSO) might not want to do business directly with the customers as they would want to have precise estimates, supply and save the trouble of the bureaucracy with millions of customers.

Before investigating if the centralised architecture can be redesigned to support a more decentralised setup, there is a need to look into the reasons against decentralising. The following two subsections will deal with this.

2.4 Decentralising the VPP

Another VPP solution is proposed by Chalkiadakis et al. [9]. The idea is that if the DERs cooperate to gain capacity, reliability and controllability they can overcome the traditional barriers to enter the energy market. An individual EV does not have much to offer in regards to the aforementioned characteristics, because they are small compared to for instance power plants and also because they also work in a decentralised way. Chalkiadakis et al. propose a cooperative solution as they name a Cooperative Virtual Power Plant (CVPP). It implements several mechanisms to ensure that the characteristics of a traditional power plant in the energy market, can be imitated by a cooperation of many DERs. As opposed to the first mentioned VPPs, where the DERs are controlled by a central entity, the key to making this work is a cooperative. The grid should be interested in these cooperatives, because they offer much needed flexibility. When there is peak demand the grid, can incite the CVPPs to increase their production. The solution in the article is built around the cooperative game theory and employs mechanisms to ensure that a subset of DERs do not have the incentive to form their own CVPP and to ensure the accurate payments based on the degree of contribution.

Grid and CVPP

The grid operators have, according to Chalkiadakis et al., two overall requirements that must be met. A CVPP must supply reliable estimates of production and in return the grid will reward CVPPs that have proven to be reliable. Secondly, the grid has no interest in working directly with too many parties and the number of CVPPs should be limited. The incentive for the CVPP is revenue and the article presents a formula that honours the two requirements, by maximizing the revenue, when the predicted production is close to the actual. The CVPP members' revenue grows as the number of members rise. However, the revenue follows a logarithmic growth, with respect to number of EVs, and will eventually flatten out, which leads to the conclusion that there is a benefit in being in a large CVPP, but there is no benefit in having only one CVPP for all EVs.

CVPP and DER

The reliability requirement of the CVPP to its DER members is the same as the one between the grid and the CVPP. The DER must provide accurate estimates and production rates. The payment function is therefore much similar without the logarithmic function, though. A reliable DER receives full payment for the power it has produced if it was accurately estimated. The figure below shows how to determine the accuracy of the estimation.

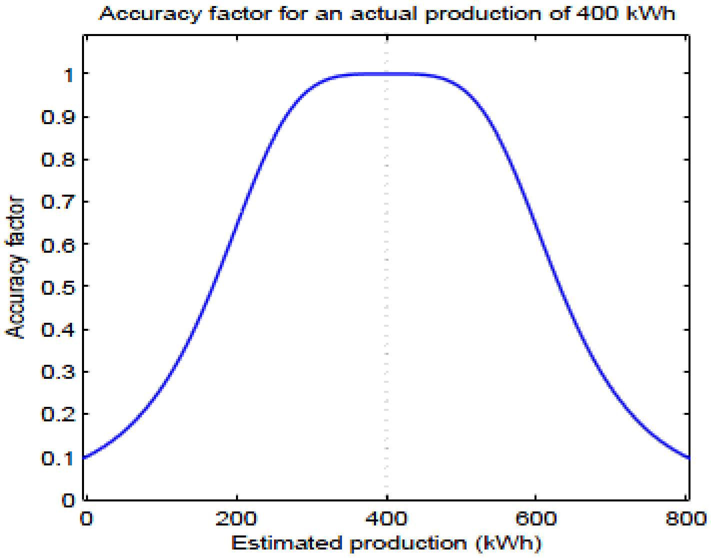
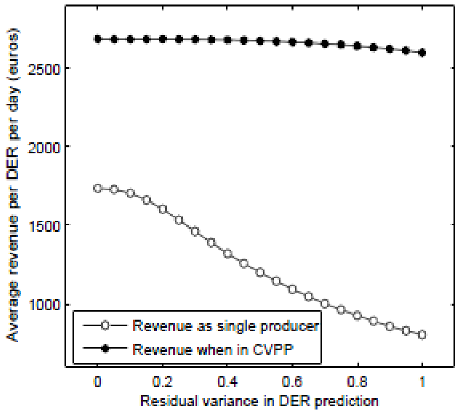


Figure 2.9: Bell curve of the accuracy factor

Managing DER members

The article also implements means of evaluating performance of the DER members so that members that constantly provide inaccurate production rates can be penalised or even excluded. The formula to compute the score takes two types of errors in account. Systematic errors are errors over which the DER has no influence, for instance imprecise weather forecast might lead to less or more production from wind turbines. Residual errors are due to errors on a particular DER for instance due to wear and tear. Consequently DERs with erroneous predictions, due to residual errors, can be expelled if their marginal contribution does not make up for the errors.

The article tests the theory on real datasets from the experimental Sotavento wind farm[1] in two cases. The symmetrical case, where all DERs are equally good or bad at predicting their production and the more interestinging asymmetric case, where the DERs are divided into two groups. The first group is made up of good predictors, with low residual deviation (0.05). The second group consists of poor predictors, with high residual deviation (0.6). The results from the simulation in the symmetrical case is shown in figure 2.10. It is evident that the incentive to join a CVPP is very high no matter the general precision rate. If the CVPP consists of 24 DERs with poor prediction abilities, they are still able to provide the grid with a fairly precise result.



(a)

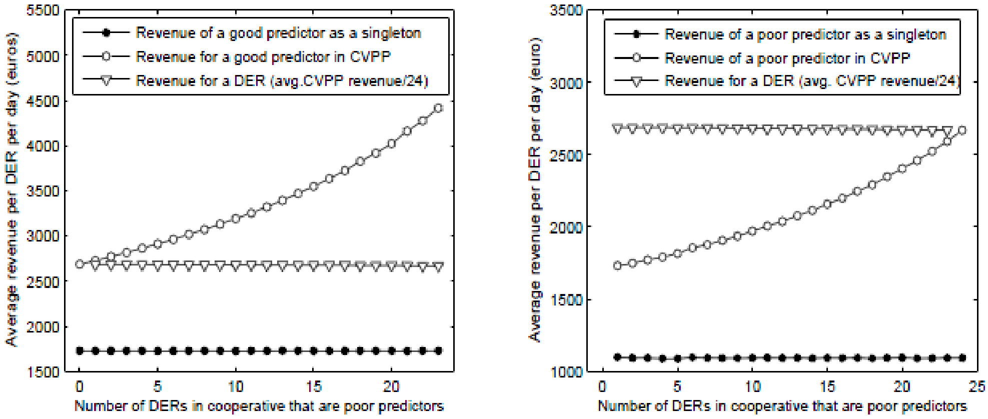
Figure 2.10: Differences between being a single producer and being in a CVPP in the symmetric case

In the asymmetric case shown in 2.11 (b) and (c), it is obvious that the DERs have high economic incentive to join the CVPP. In (b) it can be seen that if there is only a few good predictors they are very well rewarded when the CVPP pays out its revenues. A poor member will however still benefit from

---

[1] `http://www.sotaventogalicia.com/index.php`

the membership. The incentive to provide accurate estimates is paramount for the system as a whole, and it is clearly implemented in this algorithm.



(b) (c)

Figure 2.11: Differences between being a single producer and being in a CVPP - (b) shows the the revenue gain of good predictors in the asymmetric case. (c) shows the revenue gain of poor predictors in the asymmetric case

The results are promising and highly relevant when thinking of EVs as DERs.

2.4.1 Decentralised Charging Control

If all EVs operate in a decentralised way with regards to the creation of charging schedules based on historic prices, they might in average decide to charge at the same time, for instance during the night, because this is when the electricity is cheapest. A large number of EVs that decide to do so can cause serious problems for the grid, due to overloading and it might cause the price level to soar. Ma et al. presents a solution to this problem is suggested by using the Nash Certainty Equivalence Principle [20]. In short, this method exploits that a single entity in a large population can solve its own optimal control problem with knowledge of only the average value in the entire population, instead of having knowledge of all other entities in the population. In an ideal world, the EVs charging schedules converge to a Nash equilibrium as will be shown later.

In the example of a population of EVs, every EV needs to optimise their charging schedule with regards to the electricity price. To do so every EV needs knowledge of the average expected load on the grid. In the model there are two numbers that constitute the total demand. The first is the total demand by everything besides the EVs (non-EV demand) and the other is the average EV demand of the population. This allows every EV to calculate their own charging control and the strenuous computational and communication requirements that would rest on a single central entity is avoided. Ma et al. propose an algorithm that implements an iterative procedure that exploits that the EVs know the average charging strategy of the entire EV population. In practice, some central utility distributes information about this to all. The procedure is defined by the following four steps:

Step 1 A utility distributes the prediction of non-EV demand for some interval to all EVs.
Step 2 The EVs update their optimal charging scenario, taking into account the commonly known aggregate EV demand. The scenarios are returned to the utility.
Step 3 The utility updates the aggregated EV demand and redistributes it to all the EVs.
Step 4 Repeat step two and three until there are no changes.

A homogeneous population of EVs, same battery size and charging rate, will behave as shown in figure 2.12. The first iteration is shown as the blue fluctuation and it is clearly not achieving the wanted valley-filling property. The result of the second iteration, where all the EVs see the impact of the first iteration, results in the green fluctuations. It is still not performing as intended and it would continue to switch between the green and blue curves. Using this procedure alone is not enough to make the EVs converge.

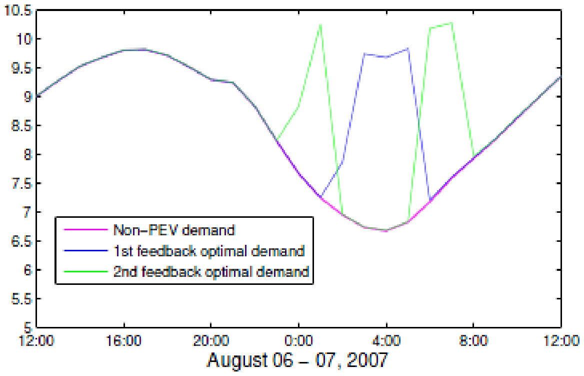


Figure 2.12: The charging strategy of a homogeneous EV group without a tracking-cost parameter

Their solution is to penalise the EVs when they differ from the average, this the authors name the tracking-cost parameter. With this in place, a homogeneous group of EVs is guaranteed to converge to the Nash equilibrium and therefore have no deviation and no penalty in the end. This is shown in 2.13.

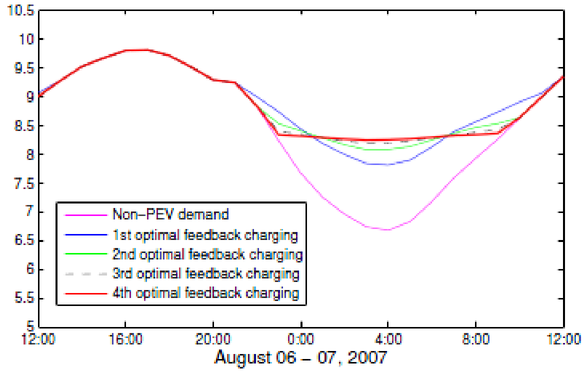


Figure 2.13: The charging strategy of a homogeneous EV group with a tracking-cost parameter of 0.007. The strategy converges to the Nash Equilibrium in the fourth iteration - red line.

A similar approach is also taken with two different groups of EVs. It turns out that they have nearly the same schedules and they come near the Nash equilibrium as seen in figure 2.14. The difference in the schedules is therefore small, and the tracking-cost parameter is also small, but not zero as in the other case. However, it is an insignificant price compared to the savings achieved.

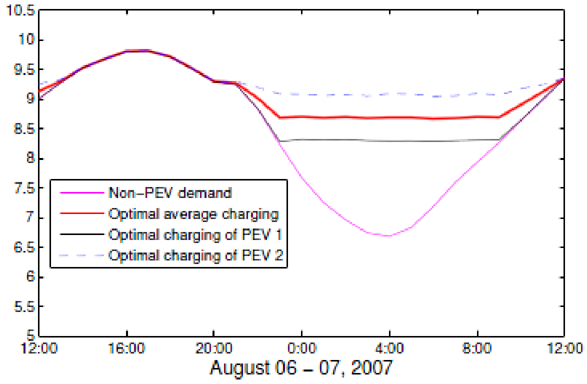


Figure 2.14: The average charging strategy of two EV groups with a tracking-cost parameter of 0.007. The strategy converges to the Nash Equilibrium in the fourth iteration - red line

Some of the challenges of this method are that the non-EV demand cannot be predicted exactly and major differences can cause the EVs to charge in a less than optimal way. This is because the method is used to forecast the demand and not react to real-time events, which a more robust solution should consider. Local grid limitations are not considered as the utility has no awareness about the location of the EVss. The charging might in general be distributed evenly over a period of time, but chances are, that within a small geographical area a relatively large portion of the EVs will charge at the same time. This will undoubtedly cause problems for the grid. Finally, the solution does not consider EVs as resources to the grid, which would change the whole charging scenario.

Ma et al. present a robust approach is taken by the same authors to be more tolerant to small or even large fluctuations in the charging estimations [21]. This is done by revising the former algorithm and using model predictive control (MPC). They achieve promising results, where the EV charging schedule changes when unforeseen incidents happen in the grid. However, they still have many limitations and use a relatively simple EV setup. This revised version of the decentralised algorithm does not account for V2G technology or for how many iterations are needed for each recalculation, when reacting to real time changes. V2G is important to fully utilize the full potential of EVs, as resources on the supply side and the recalculation matter is important, to avoid communication breakdown, due to limited bandwidth or local computational power. The simulation is also using 10.000.000 EVs as the basis for the calculation to demonstrate the theory, but it is natural to consider how smaller sets of EVs will affect the overall performance of the Nash Certainty Equivalence Principle.

The work done in regards to both the decentralised charging strategies and the CVPP suggests that it is possible and reasonable to continue to investigate the options of decentralising some of the tasks of a VPP. As the power market is today, a CVPP of some sort will most likely be preferred, even if the control can be completely decentralised. Fluctuations in predictions can be controlled and the players on the market can interact with fewer, larger unions of EVs instead of all individually. The CVPP proves that some of the barriers can be broken and allow the EVs to supply power back to the grid through the CVPP. The decentralised charging algorithm shows that it is possible with a minimum of central control, to ensure that the EVs, do not act completely without consideration of other EVs, but actually obtain a good agreement of when to charge during the night. It is done without extensive communication and without the need of communication between all EVs. The work is in no way perfect or ready for the market, but it shows result that are promising and can solve some advanced tasks, in a mild environment. These tendencies are important for the continual work with EVs as decentralised entities.

2.5 EV - A Fully Decentralised DER

Much work has already been carried out in the EDISON project, which has been introduced in section 2.3. When developing a fully decentralised EV like proposed in figure 2.2, where the EV manages all estimations and receives information from a service provider, a number of functionalities from the already developed GVPP can be deduced and utilised.

The main components in the overall architecture such as the TSO, Market, Clearinghouse, Billing Provider and DSO, as shown in figure 2.5 and 2.6, are still needed, since they are responsible for running the grid. The EVPP component on the other hand is the one, which gets a completely different responsibility. Looking at the previous section 2.3.1, the EVPP collected information from all connected DERs including EVs and based on that information, such as the price and the load on the grid, it estimates the consumption of energy.

The purpose of a fully decentralised EV is then to apply the main functionalities of the EVPP to the EV. The composed architecture of the solution is that the EVPP is replaced with an aggregator, whose task it is to collect information from the different entities and make them available for interested parties in the smart grid. The EV can then connect to the grid in two different scenarios.

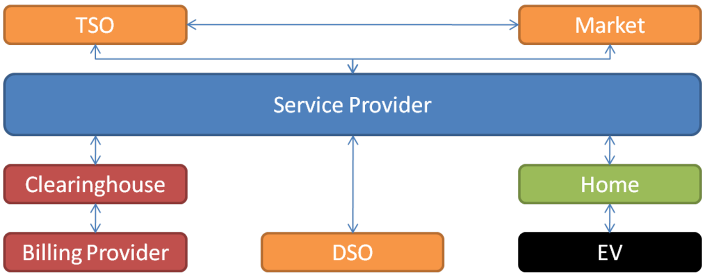


Figure 2.15: Environment diagram of the EV located at home (v2h)

In the first scenario, illustrated in figure 2.15, the EV is located at home. Here the EV charges and gets information from the service provider through a smart meter in the house. It can also contribute to the power consumption of the house, so that the house discharges the battery of the cars, when the electricity price is high in the market(vehicle-to-home, V2H). Similarly, the battery is recharged at times where the price is low. On the other hand, when the EV is located at a charging station, like the one at the office, the EV charges and gets information from the service provider as a guest at a charging station and it might still contribute to balancing the grid. This is the second scenario, which is illustrated in figure 2.16.

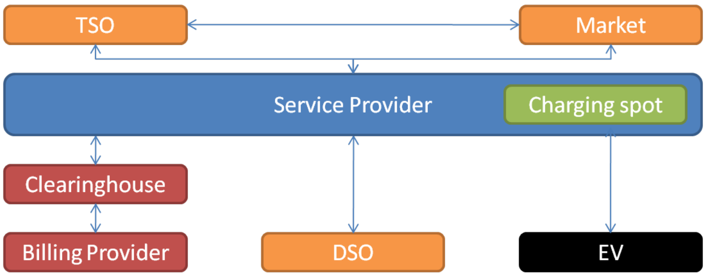


Figure 2.16: Environment diagram of the EV located at a charging spot

2.5.1 Module Structure of the EV

In extension to the composed module diagrams for the EVPP, a couple of the modules can be reused in a module diagram for the EV. Mainly it is the modules like *Historical & Statistical Values Database* and *Member Management* from the *Storage and Member Management* group. Additionally, the module for prediction and charging control can in some sort be reused. The work done by Andreas Aabrandt et al. in [1], according to the prediction and charging control, is focused on the knowledge the EVPP has about all the connected EVs. The work does therefore need adjustments and modifications for it to be applied to the EV.

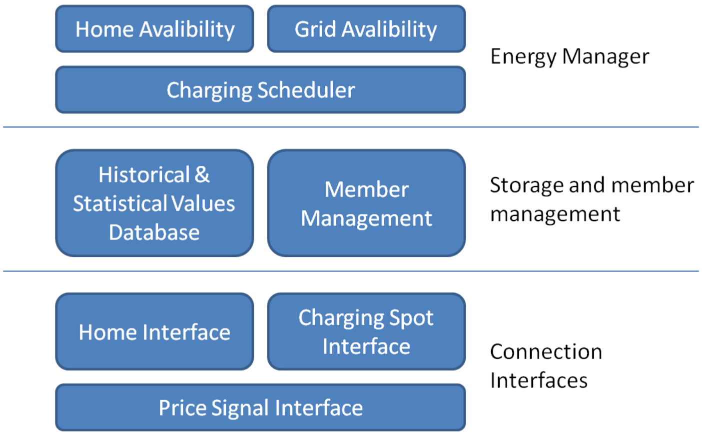


Figure 2.17: Module diagram of an EV

Storage and Member Management

This group contains two modules, which are the same as the modules in the data storage and member management group in the EDISON VPP. The first is the *Historical & Statistical Values Database*, which supports the other modules with functionality to access stored data. The second module is *Member Management*, which gives the possibility to modify the default settings and requirements of the EV operator. Additionally, it manages settings such as the surrounding connections like GPS, contract information of the operator and information on who is the owner of the EV.

Energy Manager

This group consists of three modules. The first module, *Consumption Prediction*, estimates the distribution of the EV consumption. The second module, *Price Forecast*, estimates the price based on the historical data. Lastly, the third module, *Charging Scheduler* generates a charging schedule for the EV based on the estimations made by the two modules *Consumption Prediction* and *Price Forecasting*.

Communication Interfaces

The third group, Communication, consists of three modules. The module *Home Connection Interface* is used for the EV to connect to the home charging station if EV is going to be used as a energy resource at home. The second module, *Charging Station Interface*, is used to authenticate the EV, so that it can be allowed to charge when it is connected to the charging station. Lastly, the third module is the *Price Signal Interface*, which is used to receive the price signal.

2.6 Requirements

As the result of the analysis conducted in the previous section 2.5 one can deduce the requirements of the EV to become fully decentralised. The functional requirements will be identified in the following section 2.6.1 with use cases, and the non-functional requirements will be identified and stated in section 2.6.2. Besides the functional and non-functional requirements of the EV, there are requirements to the surrounding environment, as well. These requirements will be identified in the end of this section.

2.6.1 Functional

Based on the analysis five actors of the system hav been identified:

Operator - The operator is the person, who has the possibility to configure the preferences of the EV such as the minimum level of the battery. The operator must also be able to specify a future trip or see the charging schedule. Additionally, the operator should have the possibility to activate the charging process as an instant charge if needed.

Charging Station - The charging station will act with the system during the process of the connecting the EV to a charging station.

Information Controller - The information controller acts with the system as a trigger that starts the process of extracting the market price from a service provider.

Battery Monitor - The battery monitor acts as trigger on the battery level, which starts that storing process of the battery level.

Charging Schedule Monitor - The charging schedule monitors acts with the system as a trigger that starts the process of performing a charge schedule action.

These five actors perform the different use cases that are stated in the use case diagram in figure 2.18. Each single use case is described further in details in the following tables.

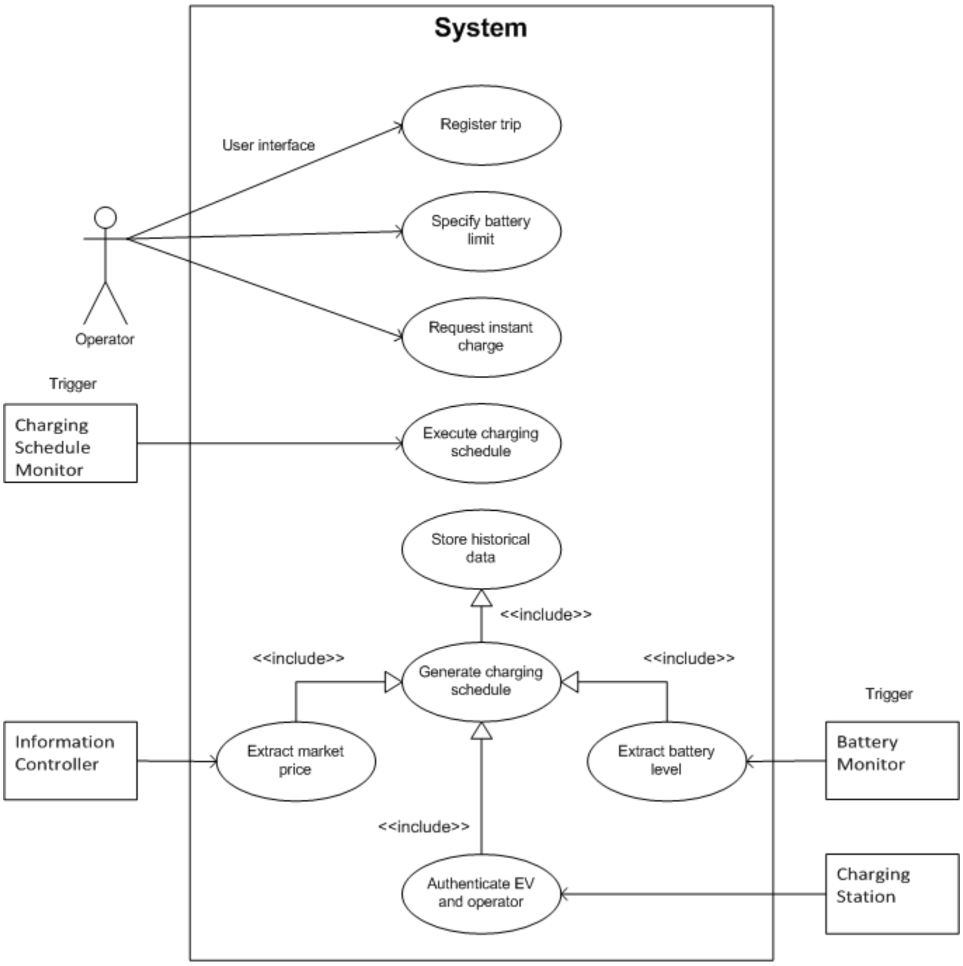


Figure 2.18: Use case diagram over the functional requirement

Use case name	Register trip
Description	A future trip for the operator is registered
Actors	Operator
Pre-conditions	The EV is connected to a GUI
Scenario	1. The operator is going on a trip 2. The operator specifies the destinations and date of the trip 3. The data of the trip is stored in the system
Alternative scenario	
Post conditions	The data of the trip is registered in the system

Table 2.1: Detailed use case description of registration of future trips

Use case name	Charge battery limit
Description	The operator wants a minimum limit for the battery
Actors	Operator
Pre-conditions	The EV is connected to a GUI
Scenario	1. The operator specifies the minimum limit 2. The system stores the configuration
Alternative scenario	
Post conditions	The EV got a battery limit configured

Table 2.2: Detailed use case description of configure the battery limit of the EV

Use case name	Request instant charge
Description	The operator wants the EV to charge instantly
Actors	Operator
Pre-conditions	1. The EV is connected to a charge station and authenticated 2. The EV is connected to a GUI
Scenario	1. The operator requests an instant charge 2. The system checks that the EV is connected to a charging station 3. The charging schedule is canceled 4. The system starts the charging of the EV
Alternative scenario	2. The EV is not connected to a charging station 2.a. The charging request is rejected
Post conditions	The EV starts charging immediately

Table 2.3: Detailed use case description of the perform instant charge

Use case name	Execute charging schedule
Description	The system executes the charging schedule
Actors	Charging Schedule Monitor
Pre-conditions	1. A charging schedule is generated 2. The EV is connected to a charging station
Scenario	1. The system extract the next schedule action 2. The system checks that the EV is connected to a charging station 3. The system execute the schedule action 4. The system waits until next schedule action needs to be executed
Alternative scenario	2. The EV is not connected to a charging station 2.a Waits to execute the action until connected 2.b The schedule action is outdated
Post conditions	The charging schedule is executed

Table 2.4: Detailed use case description of execution of the charging schedule

Use case name	Store historical data
Description	Historical data is stored in the storage
Actors	System
Pre-conditions	1. Historical data is generated
Scenario	1. The system receives historical data 2. The system stores the historical data in the storage
Alternative scenario	
Post conditions	The historical data tag is added to the storage

Table 2.5: Detailed use case description of storing historical data

Use case name	Generate charging schedule
Description	Generates a charging schedule and sends it to charge car component
Actors	System
Pre-conditions	1. There is sufficient statistical data in the database to generate the schedule
Scenario	1. The charging scheduler is triggered when new data is received 2. The system passes the data to the storage 3. The charging scheduler algorithm recalculates the charging schedule 4. The new generated charging schedule is prepared to be executed
Alternative scenario	
Post conditions	The EV has a newly generated charging schedule

Table 2.6: Detailed use case description of the generate charging charging schedule

Use case name	Extract market price
Description	The system extracts market prices from service provider
Actors	Information Controller
Pre-conditions	1. Connection established to service provider
Scenario	1. The system extracts market price from service provider 2. The system passes the price to the charging scheduler 3. The system waits five minutes until next extraction
Alternative scenario	
Post conditions	The market price is extracted from the service provider and passed on to the charging scheduler and storage

Table 2.7: Detailed use case description of extracting market prices

Use case name	Extract battery level
Description	The battery level is monitored and stored as historical data
Actors	Battery Monitor
Pre-conditions	1. A monitoring censor exists established on the battery
Scenario	1. The system extract the battery level 2. The system passes the battery level to the charging scheduler 3. The system waits five minutes until next extraction
Alternative scenario	
Post conditions	The battery level is extracted and passed on the to charging scheduler and storage

Table 2.8: Detailed use case description of extracting battery level

Use case name	Authenticate car and operator
Description	Before being able to charge the EV at any given charging station the car and operator must be authenticated.
Actors	Charging Station
Pre-conditions	1. The car is parked at a charging station and turned off
Scenario	1. Operator authenticates toward charging station 2. The EV validates the user credentials 3. The EV authenticates towards charging station 4. Charging spot verifies the EV 5. The EV receives authentication answer 6. The time of connection is passed to the charging scheduler
Alternative scenario	2. Operator is not valid 2.a. Retry 2 2.b. Connection denied 4. EV is not verified 4.a Connection denied
Post conditions	The EV and operator is successfully authenticated

Table 2.9: Detailed use case description of the authenticate car and operator

2.6.2 Non-functional

Besides the functional requirements identified in 2.6.1, there are also non-functional requirements for the EV. These requirements are as follows:

Performance

The minimum demand is that a new charging schedule is generated before the next price signal is extracted.

Usability

EV operator interfaces should be intuitive and easy to use. Especially important in regards to the EV display during a drive. Therefore, a minimum of user interaction is a must.

Security

The price signal must be secured so that the EV can verify the integrity of the signal. The price signal does not need to be secured against eavesdroppers as the price is not considered confidential.

The communication between charging station and EV must be secured.

The EV operator must be authenticated when connecting to the charging station to hinder misuse.

Availability

The EV should extract a price signal every five minutes. It is not critical for the EV, as lacking updates can be extracted at a later stage, for instance when the service is restored or through a charging station. A new schedule would then be generated.

Modularity

The system should be modular. Due to much development within grid research the solution must be able to be extended.

Adaptability

The system should be adaptable. It is likely that there will be changes in the environment surrounding the EV and it must therefore be able to adopt with a minimum of change to existing parts.

Testability

To have the system as adaptable and modular as possible, there must be an option of testing individual components.

2.6.3 Requirement for Surrounding Environment

This section contains a list of the requirements that are imposed to the surrounding environment of the EV so that the EV can act on its own.

Price Signal
A service that offers the price signal to be extracted is needed.

Managing the Price Signal
The control of the price signal can turn out to be critical in controlling the EV population both in regards to the aforementioned Nash equilibrium, but also to avoid local grid constraints. A differentiation in the price signal in context with the location of an EV could help controlling the EV population. Also, EVs will be unaware of local constraints and only react according to the extracted price signal. It is therefore a requirement that the price signal includes such information or that these problems are sought solved in other ways. It is assumed to be out-of-scope for this thesis.

Authentication at Connection Time
An external service that authenticates the EV and the operator when connecting to the charging station.

2.7 Domain Model

The domain model relates physical or abstract objects in the system domain. The model consists of two overall models. One showing the EV and its internal domain and the other showing the environment in which the EVs operate. The model gives an overview of the whole system and its environment, which is important to have visualised and clarified before the implementation, in order to streamline the rest of the design and development process. In the following a description will be made of the separate objects to explain their connection in the overall picture:

ACS Unit - The *ACS Unit* object is the core aggregator of the EV, which runs the system that manages the components of the EV.

Price Controller - The task of the *Price Controller* is to retrieve the information that the Service Provider makes available.

Price History - The *Price History* is for the history of the price tags that are received from the Service Provider.

Information Controller - The task of the *Information Controller* is to manage the different configuration information from the operator of the EV and the history data of trips travelled.

Trip History - The *Trip History* represents a historical tag of a trip the EV has travelled.

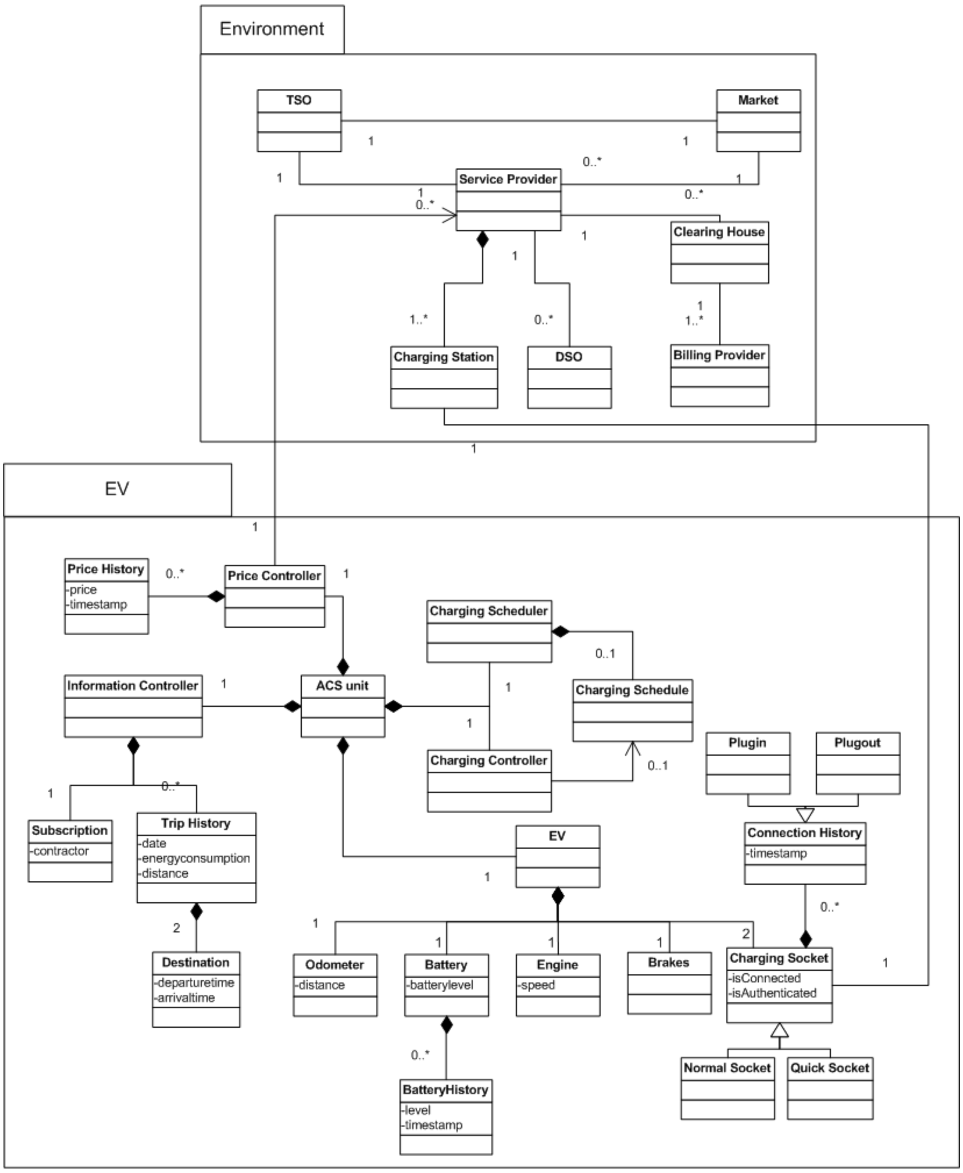


Figure 2.19: Class diagram of the domain model

Destination - The *Destination* object represents a location, where the EV has travelled from and to, with the departure and arrival times.

Subscription - The *Subscription* is the representation of the EV subscription signed at a supplier.

Charging Scheduler - The task of the *Charging Scheduler* is to generate a new charging schedule when historical and statistical data have changed.

Charging Schedule - The *Charging Schedule* is a representation of the generated charging schedule, which the charging scheduler produces.

Charging Controller - The *Charging Controller* is the representation of the component that manages the charging of the EV by executing the actions from the Charge Schedule or by request from the operator.

EV - The *EV* is a representation of the physical EV, that consists of an engine, a battery, brakes, an odometer and two charging sockets.

Engine - The *Engine* is a representation of the physical engine of an EV and the link to it.

Battery - The *Battery* is a representation of the physical battery of an EV and the link to it.

Battery History - The *Battery History* objects are stored to get a historical overview of the battery energy balance.

Brakes - The *Brakes* is a representation of the physical brakes of an EV and the link to them.

Odometer - The *Odometer* is a representation of the physical odometer of an EV and the link to it.

Charging Socket - The *Charging Socket* is the representation of the physical charging socket in the EV. The EV is supplied with two types of sockets, one for normal charging (*Normal Socket*) and one for quick charging (*Quick Socket*).

Connection History - The *Connection History* keeps a historical overview of the situations where the EV is plugged in or out at a chargin station. The *Connection History* can therefore be of the types *Plugin* or *Plugout*.

The objects in the surrounding environment are the following. Though, this report will not cover these object in details:

TSO - The *TSO* represents the Transmission System Operator, who has the task to keep the grid stable and maintain and operate the high-voltage grid.

Market - The *Market* represents the energy market where the electrical prices are negotiated.

Clearing House - The task of the *Clearing House* is to bill the EV operators for the charged energy.

Billing Provider - The *Billing Provider* is a monetary institution, which can withdraw the money from the consumers.

DSO - The *DSO* represents the Distribution System Operator who has the task of maintaining the low-voltage grid and supply the consumers.

Service Provider - The task of the *Service Provider* is to collect the information which is necessary for the EV and make it available by sending it to the info managers of the EV. This should include information like the price signal and the load of the grid.

Charging Station - The *Charging Station* is responsible for the connection with the socket of the EV mentioned above.

2.8 Conclusion

This chapter has covered problem statements 1, 2, 3, 4 and 5 as defined in 1.3. The ability to collaborate was identified as a key attribute among stakeholders in the EV industry. Analyses of different collaborations, such as the one between Mitsubishi and PSA Peugeot Citroën, led to general knowledge about state of the art within the EV industry and the grid domain.

The next step was to turn to the existing solutions building on the VPP concept and analyse different VPP solution such as the EDISON VPP. Central control was a problem in relation to generating and maintaining charging schedules of a large EV population. The subject of having decentralised EVs was a relatively blank page within the industry and an analysis was carried out to address the major problems of decentralising the control.

Based on the analysis there was basis to continue the work with a decentralised EV. However, the findings in the chapter indicated that a 100 % decentralised EV was not realistic as the market is today:

- Means of overcoming barriers of entering the grid still require some central control
- To control local grid constraints and to ensure that the general population do not start charging at the same time, some amount of central control is required.

Promising research results were presented to prove that theoretically this can be solved. How it can be implemented is omtted for other research projects and are out-of-scope for this thesis. From the analysis it was also evident that some modules of the EVPP are reusable and can be transferred to the EV.

By identifying actors and how they interact with the system a use case diagram was designed with use case descriptions as a consolidation of the wanted functionality. Furthermore, non-functional requirements were identified of which the most important were security and modularity.

The idea of implementing a EV prototype, which autonomously generates a charging schedule, is a rather novel approach and the proof of concept is therefore extra critical to support the analysis. The next chapter will take a look at how the architecture of an autonomous charging scheduler (ACS) in an EV can be designed based on the domain model.

CHAPTER 3

Architecture Design

Besides the analysis and requirement specification, one of the objectives for this thesis is to come up with a proposal for a suitable EV architecture. In this chapter a thorough walk-through of such a possible architecture will be conducted. The first decision is what overall approach to take, when starting the design and this involves a description of SOA and related security aspects. The following sections describe the architecture of the different services and the communication between them.

3.1 SOA - Service Oriented Architecture

"Service-oriented architecture (SOA) is an architectural style that modularises information systems into services"[4]. Earlier systems were tightly coupled because they were bound to the same language-specific objects and thus the same platform[2]. SOA aims at letting the internal communication in each service be language-specific, while having the external communication follow certain universal standards. This allows the services to work across platforms and programming languages. Web services use the commonly supported XML protocols as standard for the external communication.sz

Olsen et al. describe four essential guidelines to follow when designing web services [2].

Loosely Coupled - When making a call to a web service the caller should not be required to have any domain or platform specific knowledge. This infers a separation of the logic and the external interface of the web service. How to connect to the web service or where it is running should be transparent to the user. These guidelines ensure maximum flexibility, but makes it hard for the user to know how performance will be. Therefore, performance should be considered, when designing the solution.

Autonomous - The idea is to split the overall design into autonomous services, which can be added, removed or changed as needed. If one service fails the rest should not fail. Error handling in the web service itself must ensure that it does not crash and always replies to any calls made to it. If a webshop wants to change its payment service it can be done by changing the service alone, without having to change other parts of the existing architecture. If a service does not work, callers of the service must be informed to avoid doubt about the transactions.

Shares Formal Contract - To ensure that platform independence is in place contracts must be made of what will be exchanged through the XML descriptions. A service has contracts to handle the data types, the services made available by the web service and contracts of how the messages are sent. It is evident that changes to these contracts should be avoided especially in large scale implementations with legacy requirements, as might be the case in the EV scenario with many different types of EVs.

Transport and Compatibility Policies - Information about choice of transport protocol and security information are exchanged using policies and are related to the quality of the service. These policies can be used to improve performance or security, and they provide a policy, which can be changed if some new protocol is to be used. If web service needs to upgrade its security mechanisms, then this can be handled by adding a new policy and removing the old. Clients are then forced to use the new policy to perform calls.

The loose coupling and autonomous aspects are essential to the project developed in this report, as the final solution should be modular, so that it allows the services to be developed independently and allows them to be replaced by other modules at a later stage. SOA will be used internally in the EV to ensure above properties.

The idea of having web services that can be reached without having to worry about firewalls and allow the design to identify and breakdown the architecture into relevant services, is perfect in a project like this, which involves a lot of different partners, who in different ways interact with the system. The web service approach is also very useful as this project uses external providers (price signal, user interaction and authentication) and these can be integrated as services.

Decker et al. propose a solution model where DERs connect to GVPPs through a so-called match maker [11]. The match maker ensures that the connection links are established and afterwards the DER and GVPP can communicate directly. The match maker provides loose coupling between DERs and GVPPs. A DER can change GVPP if another has better offerings. This approach of using SOA not only internally in the EV system, but also with the surrounding entities will be adopted and used in the rest of the design.

3.1.1 SOA - Security

One of the important aspects when choosing SOA is regarding security. Since SOA has become so widely adopted, security measures have already been taken and they are well documented and proven. This is important as security in a setup with decentralised EVs is a considerable concern. In the following a list of the most common threats, identified by National Institute of Standards and Technology[14], will be examined:

- Replay attack. An attacker intercepts a package, copies it and resends it. The attacker does need to have knowledge of the content of the package to do so. In the advanced man-in-the-middle attack the attacker is able to intercept and alterate a message between the receiver and sender before replaying it. These flaws can be hindered if the message is encrypted, digitally signed and gets a nonce.
- Eavesdropping. An attacker is able to read data on the network from or to the web service. This is especially a problem if the message header contains credential data. It can be avoided with the proper encryption mechanisms to provide confidentiality.
- Confidentiality attack. The confidentiality must be protected, but the importance is different depending on the situation. The price signal is probably not critical to disclosure, however, information about when, who and where the operators charge, their IDs and whereabouts should not be disclosed. Information related to payment is critical. A disclosure of confidential information can happen in case that insufficient authorisation and authentication measures are used.
- Message tampering. If an adversary can change parts or the entire message there is no message integrity. This can be avoided by using encryption (privacy) and digital signatures (integrity).
- Authenticity of sender. An adversary who tricks the receiver to believe that the message is in fact from another person. Avoided by means of digital signatures.

- Web Service Description Language configuration data disclosure. If the web service supports WSDL in a dynamic generation format or as downloadable files, it can provide valuable configuration information disclosure for an attacker. Implementation details could be disclosed if insufficient exception handling is implemented[1].

These are the threats against the system as whole. The goal of this thesis is protect the price signal. The remaining system should be secure as well, however, these details are out-of-scope. In section 3.4 a description is made of how to secure to price signal.

3.2 Solution Architecture

As discussed earlier in the chapter the use of SOA as an architecture design, gives the advantage of loose coupling between the different components in the ACS solution. Based on the concepts stated in the domain model in section 2.7 and the identified requirements in section 2.6, the work performed within this thesis has resulted in a picture of the architecture for an ACS unit in a fully decentralised EV.

The illustration in figure 3.1 shows the internal components that define the ACS unit. Even though the EV must work fully decentralised, the ACS unit still requires some data from the environment such as a price from the *Service Provider* and plug-in/plug-out signal from the *Charge Station*, for the ACS unit to be functional. Figure 3.1 therefore shows, besides the internal components, the external components for the ACS unit, as well.

The proposed architecture contains seven key services, which operate the main functionalities of the ACS. These services are the following:

Information Controller whose task is to extract the price from the external information provider.

Persistent Controller whose task is to manage and control access to the persistent storage.

Charging Scheduler whose task is to estimate and calculate the charge schedule for the EV.

Charging Controller whose task is to execute the charging of the EV, performed from the charging schedule or a request from the user.

Charging Station Controller whose task is to manage communication with a charging station when connected to one.

---

[1] http://msdn.microsoft.com/en-us/library/ff648643.aspx

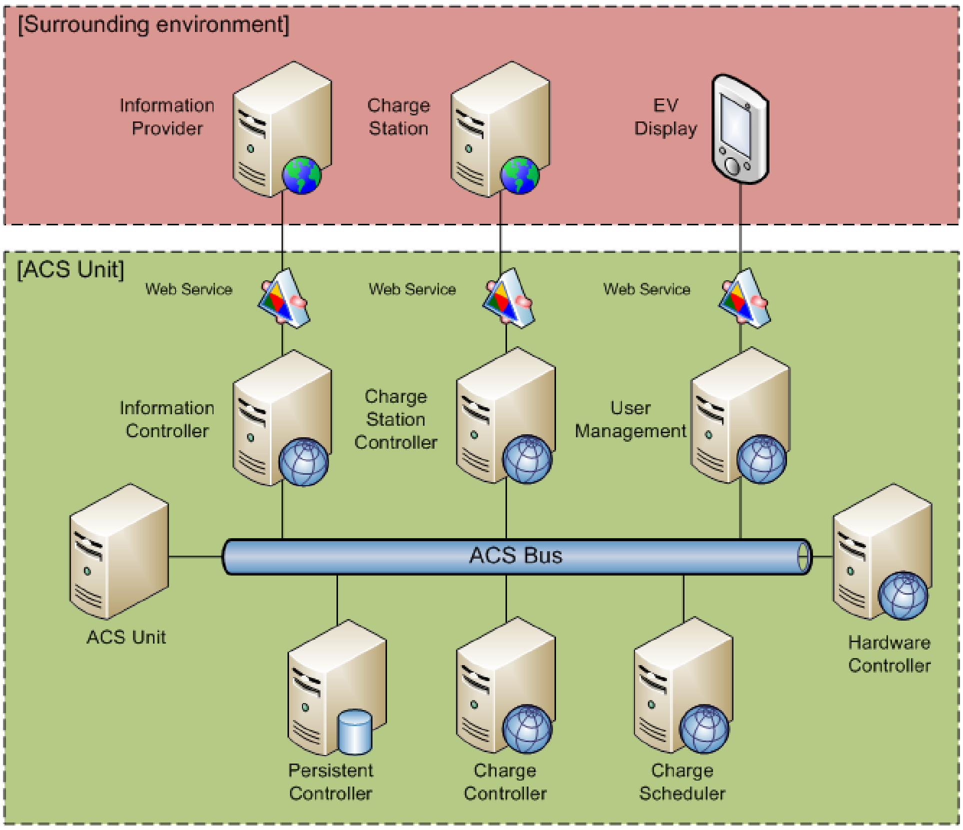


Figure 3.1: The architecture of the services in the ACS

User Management whose task is to communicate with the user interfaces, that show the status and information of the EV.

Hardware Controller whose task is to act as the interface between the ACS unit and the hardware.

Besides these seven key services, the ACS unit consists of one other internal service as you can see in figure 3.1, which is the following:

ACS Unit whose has the task of initialising and administrating all the services of the EV.

The communication between the internal services occurs across an internal communication channel, which is figure 3.1 is illustrated as the ACS Bus. On the other hand, for the internal services to receive the required data from the surrounding environment, the internal services such as - *Information Controller*, *Charging Station Controller*, *User Management*, offer or require global interfaces, respectively as web services.

These offered or required global interfaces are then used by the external services which are the following

Information Provider whose task is to offer the price signal every fifth minute.

Charge Station whose task is to authorise the operator and EV, when connecting to a charging station.

EV Display whose task is to offer the states and information of the EV for the operator plus give the operator the possibility to configure settings of the ACS unit.

The architectural design of these external services are out-of-scope for this thesis and are not described more in details. However, section 3.4 looks at already conducted work and which architectural designs could be integrated with the system. The following sections will look into further details of the functionality that are required or offered by the interface of the internal service.

3.3 Component Architecture

Even though the architectural design of the ACS solution discussed in section 3.2 supports loose coupling between the involved components, they still have some dependencies between them.

The diagram shown in figure 3.2 illustrates the design of the components and their dependencies as they are structured in the ACS Unit. As one can see in the component diagram, each component is relatively mapped to their respective service from the solution architecture in figure 3.1. This is because each service operates independently. However, each service component is a subcomponent of the ACS Unit component, which acts as a host that makes them available.

Thus, descriptions and details of the functionality of each of the seven services are covered in the following sections, plus the interfaces that they use or offer.

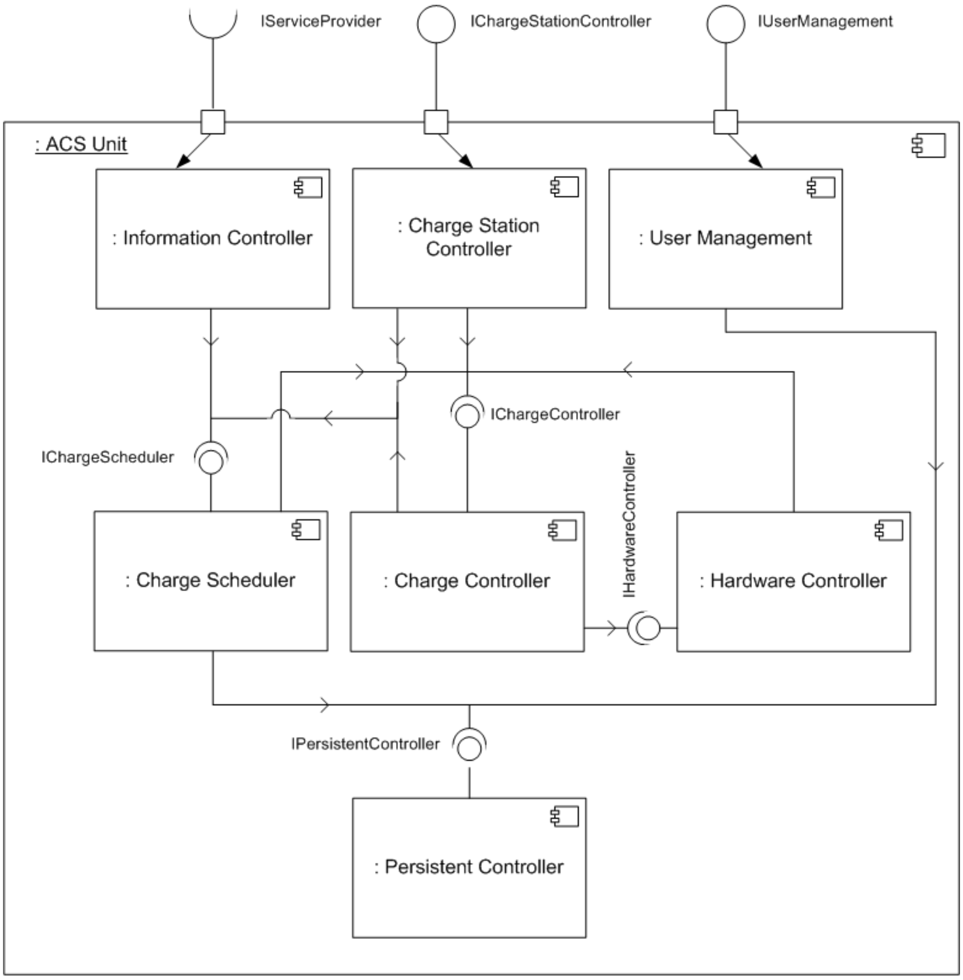


Figure 3.2: The architecture of the component in the EV

3.3.1 Information Controller

The role of the Information Controller service is to extract the price signal from the Service Provdier. This functionality of extracting the offered market price covers use case requirement identified in 2.6.1 called *Extract market price* in figure 2.18.

The information controller requires an interface, shown in figure 3.3 of an external service that provides the information controller with functionality, which returns a price. Besides the dependencies of the external Service Provider service, the Information Controller depends on the two internal services, Charging Scheduler which it passes the market price on to using the IChargeScheduler interface 3.5.

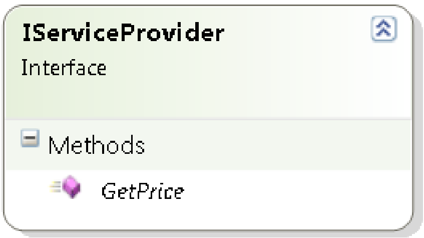


Figure 3.3: The IServiceProvider interface

3.3.2 Persistent Controller

The task of the Persistent Controller service is to act as an access point to the storage of the ACS unit. The service offers functionality that makes it possible for the other services to store or extract historical and configurational data of the ACS unit. These functionalities cover the use case identified in 2.6.1 named *Store historical data* in figure 2.18. To comply with these functionalities the service offers an interface with the functionalities shown in figure 3.4.

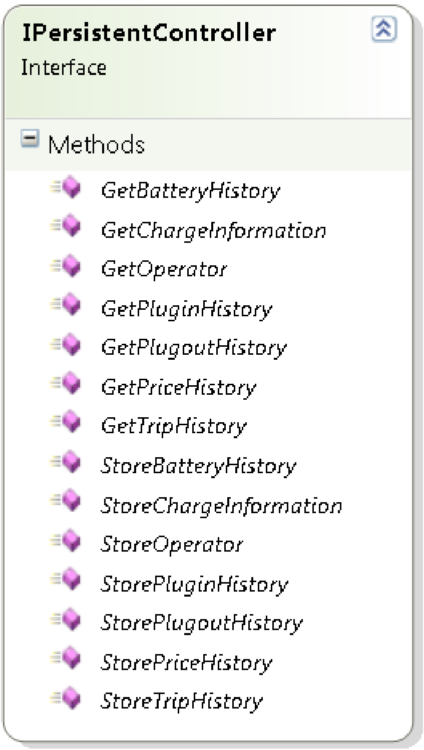


Figure 3.4: The IPersistentController interface

3.3.3 Charging Scheduler

In 2.6.1 the use case, *Generate charging schedule*, was identified and was shown in figure 2.18. The functional requirements of this use case is covered by the functionality of the service Charging Scheduler. The Charging Scheduler service has the task to calculate and generate the charging schedule of the EV.

For the Charging Scheduler service to meet these requirements it requires historical data from the environment. The IChargeScheduler interface, figure 3.5 is then offered by the environment service to provide the charging scheduler with data. When the charging scheduler receives new data it passes the data on to the storage using the IPersistentController interface, figure 3.4 followed by a recalculation of the charging schedule. Therefore, the Charging Scheduler is dependent on the IPersistentController interface.

Additionally, the Charging Scheduler depends on the IChargeController interface figure 3.7, because when the Charging Scheduler has generated a new charging schedule it passes the charging schedule on to the Charging Controller, that then executes it.

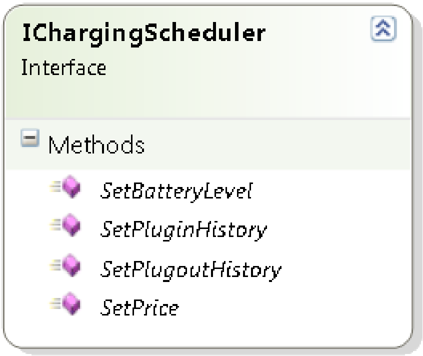


Figure 3.5: The IChargeScheduler interface

3.3.4 Charging Station Controller

The role of the Charging Station Controller is to administrate communication with a charging station when the EV is connected to one. When the EV connects to a charging station, it requests authentication of the operator and EV. The charging station then processes the entered information and returns an authorisation result. This functionality covers the identified requirement in the use case named *Authenticate EV and Operator* in figure 2.18.

The Charging Station Controller service therefore offers an external interface shown in figure 3.6, which provides functionality when requesting EV information, authorisation or receives disconnection information.

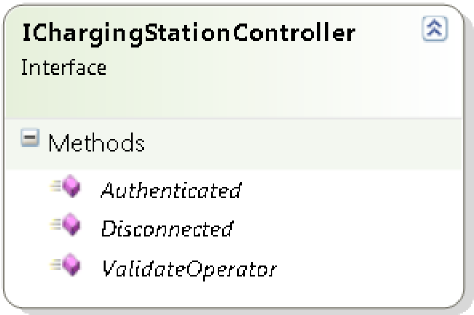


Figure 3.6: The IChargeStationController interface

When an authorisation or a disconnection to a charging station is performed the Charging Station Controller notifies the Charging Scheduler with the historical data of when the connection occurred. This requires the functionality offered by the IChargeScheduler interface. In addition, the Charge Controller service is notified, as well through the IChargeController interface.

3.3.5 Charging Controller

The task of the Charging Controller service is to administrate the charging of the EV. This is done e.g. by executing a received charging schedule or by a received instant charging request from the operator. This functionality covers the identified use cases in 2.6.1 called *Execute charging schedule* and a part of the *Instant Charge*.

The charging controller therefore offers an interface that supports the required functionalities, which are illustrated in figure 3.7. Besides the input on when to start or stop charging, the charging controller requires that it is told that the EV is connected and authorised to a charging station. This information is as mentioned in the previous section 3.3.4, received through the Charging Station Controller service.

When executing a schedule or instant charging request the charging controller service will communicate with the Hardware Controller service to which it will request a start or stop of charge, and are therefore dependent on the interface shown in figure 3.9.

The functionality described in the use case named *Extract battery level* in figure 2.18 is covered by the Charging Controller, as well. Besides managing

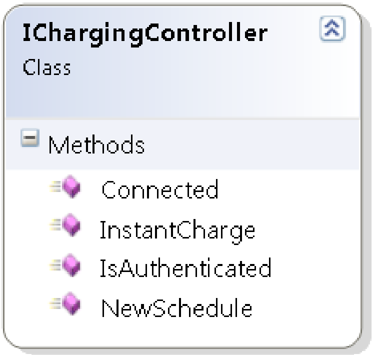


Figure 3.7: The IChargeController interface

the charging process of the EV the Charging Controller performs logging of the battery level. To meet these requirements, it must perform a request on the battery level from the IHardwareController interface every five minutes.

3.3.6 User Management

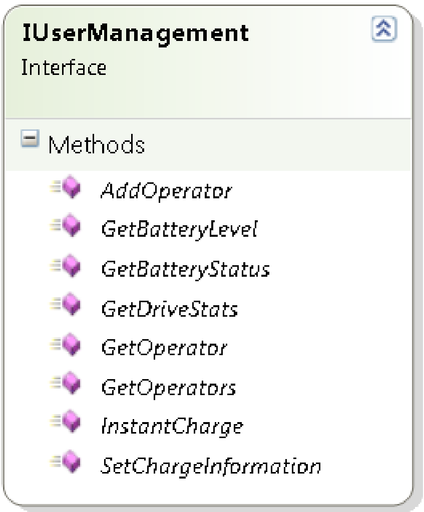
The User Management service offers an external interface, which is shown in figure 3.8 for user interfaces to communicate with the ACS unit. The interface provides the functionalities which give the operator the possibility to view state and status of the EV. Additionally, the operator has the options to configure the ACS unit e.g. specify a minimum level for the battery or add destinations for a future trip. These functionalities are the ones, which cover the requirements that are identified in the use cases in 2.6.1 called *Register trip*, *Specify battery limit* and *Request instant charging*.

Besides the offering of the IUserManagement interface, the User Management service is dependent on the IPersistentController interface. This is because when an operator either makes a request for information or specifies a configuration the User Management service communicates with the Persistent Controller service that extracts or stores the information in the storage.

3.3.7 Hardware Controller

The role of the Hardware Controller service is to provide a connection point for communication with the hardware. The hardware controller therefore offers an interface that provides functionality to begin charging, stop charging or extract the energy level of the battery, as shown in figure 3.9.

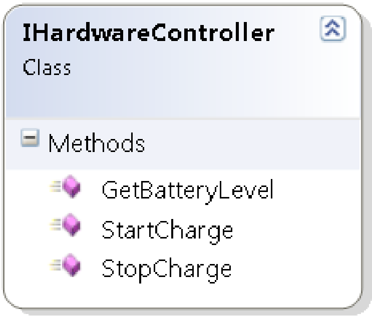
The Hardware Controller is however dependent on the IChargeController



IUserManagement
Interface

Methods
AddOperator
GetBatteryLevel
GetBatteryStatus
GetDriveStats
GetOperator
GetOperators
InstantCharge
SetChargeInformation

Figure 3.8: The IUserManagement interface



IHardwareController
Class

Methods
GetBatteryLevel
StartCharge
StopCharge

Figure 3.9: The IHardware interface

interface figure 3.7, because when the EV connects to a Charging Station the Hardware Controller passes a connection signal to the Charging Controller service so that the Charging Controller can compare the signal with the authentication.

3.4 Required External Services

The architectural design of the external services is mentioned in section 3.2 and are out-of-scope. This section will however look into the three services that are required for the ACS unit to be fully functional.

The walk-through will look at some architectural concepts, that have been developed in other projects. The reason for looking these projects is that their architectural design could be integrated with the ACS unit without big adjustments.

3.4.1 User Interface

The EV operator needs to interact with the EV in order to access and set charging preferences. The operator would also need to see basic information such as battery level, charging schedule, expected range and time to full charge. Ketan Singla thoroughly investigates how to design such a user interface [15]. He designs a web interface and a web page designed for mobile phones. The user interface is designed with regards to usability to allow best possible interaction with the system for the operator. The ACS solution should be designed so that it can be extended with Ketan's user interface.

3.4.2 Authentication

Wind et al. present a system where the EVs are allowed to roam between different VPPs [10]. It includes an authentication mechanism using an RFID tag, where the VPPs can exchange information about user membership. The exchange is suggested secured using encryption, where the key exchange is done using a ”switchboard” as a public key repository. The repository acts as central register so all VPPs do not need to have public keys exchanged with all other VPPs. Furthermore, by adding the transport layer security (TLS) a symmetric key exchange can be established to allow rapid encryption / decryption. The system allows an EV to connect to any charging station, authenticate and charge if the the authentication was successful. The design of the ACS solution should allow for future interaction with the authentication solution.

3.4.3 Service Provider

The ACS system needs a price signal. This is expected to come from a service provider. The service provider's role is to be an intermediary between the DSOs and the EVs. The service provider will buy power from the grid and sell it to its associated EVs, which might be on a subscription - an analogue to the cellular network market. When an EV connects to a charging station the infrastructure behind the charging station should authenticate the EV, and allow access to the charging services using the infrastructure architecture described by Wind et al.

The price signal provided from the service provider must be secured to maintain authenticity of the service provider and integrity of price signal, when received at the EV. To do so the price signal needs to be signed by the service provider. This is done by hashing the price signal and then encrypting it using the private key of the service provider. This process is only needed to be performed once for every price signal. The original price signal is send along with the signature to the EV.

The EV must know the public key of the service provider. The EV can then decrypt the signature using the public key of the service provider. Finally, the price signal is hashed using the same function as the service provider and the result is checked with the decrypted signature. The check is successful if the two hashes are equal. In this way both price signal integrity and sender authenticity are verified. By adding a time stamp replay attacks can be hindered.

It is assumed that appropriate hashing algorithms and encryptions schemes are chosen to avoid standard problems such as collisions and weak encryption in a final implementation.

3.5 Conclusion

In this chapter the problem statements six and seven described in 1.3 have been covered.

The requirement specification had a clear focus on the modularity of the system to be designed. SOA was identified as an architecture design that would lead to the most modular and flexible solution. Other projects related to the implementation of a VPP were found, analysed and used as inspiration for the design of seven key services and to confirm the choice of architecture design. Dependencies between services were identified and described.

In section 2.6.3 requirements to the surrounding were identified and in the design suggestions were made of how to handle these:

- The user interface which will allow the operator to access EV data and set preferences were found in a report focused on the usability of such an interface. The report forms a reference for future work to improve integration with the user interface.
- A roaming system with authentication of the EV and operator was presented and chosen as authentication provider to be implemented in a future expansion.
- The role of a service provider is a key role in a complete setup, where

> all parts have been developed. It willl be responsible of controlling its associated EV population in terms of the grid and also in terms of user profiles for authentication and billing procedures. The role is somewhat like a traditional mobile service provider.

The next chapter covers the development of the prototype.

Chapter 4

Prototype Development

Based on the proposed architecture design, which is conducted in section 3, a prototype of the ACS unit is developed as a proof of concept. First the chapter will cover the decisions made according to the choices in the development technology section. Additionally, the chapter will cover the overview of the solution and the developed services and components.

4.1 Development Technology

As mentioned in section 3.2, a proposed design for the ACS unit is done by using the SOA approach. To support the architecture the Microsoft .Net framework, Windows Communication Foundation (WCF), will be used for the communication between the different services. Additionally, the Microsoft .Net framework, Windows Workflow Foundation will be used for the behaviour of services and Windows Presentation Foundation will be used as the graphical interface.

WCF is now part of the .NET Framework 4 and provide support for SOAP, WSDL, WS-Discovery, WSDL, WS-Policy. Furthermore, WCF handles security through support for WS-Security, WS-Trust and WS-SecureConversation, which all adds to the SOAP header to provide authentication, data integrity, data privacy and other. Furthermore, WS-ReliableMessaging provides an addition to the SOAP header that ensures reliable end-to-end communication even through several SOAP intermediaries.[1]

---

[1] http://msdn.microsoft.com/en-us/library/ee958158.aspx

4.2 Solution Design

To support the proposed architecture based on SOA as discussed in chapter 3. The designed prototype solution will implement each service and component in separate assemblies. This creates the modular structure with loose dependencies between the different components, where the communication occurs between interfaces as described in section 3.2.

Each of the internal services are developed based on their identified required behavior in 3.3, which is described with the use of activity diagrams. Additionally, the structure of the services are designed by using class diagrams to specify the relations and associations between classes.

As the objective of the thesis is to develop a prototype of a decentralised EV, the external services and the graphical displays are only developed to demonstrate the basic functionality of the ACS unit. The simple implementation of the external services are described in section 4.2.2 and the implementation of the graphical displays, that are used to monitor the services and behaviour of the ACS unit, are described in 4.2.3.

4.2.1 ACS Unit Services

As mentioned in section 3.2 the components of the ACS unit are the following: Information Controller, Persistent Controller, Charging Scheduler, Charging Controller, Charging Station Controller, Hardware Controller, User Management.

4.2.1.1 Information Controller

To support the task of managing the extraction of the price signal, as identified in 3.3.1 the activities of the Information Controller service are as illustrated in figure 4.1. These activities are performed repeatedly after the Information Controller is started. The needed structure of the Information Controller service, which is described in figure A.1 in the appendix, consists of a class that manages the initialisations and a process class that performs the core process with dependencies to the interfaces IServiceProvider and IChargeScheduler as designed in 3.3.1.

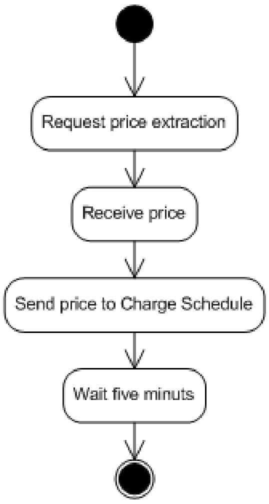


Figure 4.1: The activity diagram of the Information Controller

4.2.1.2 Persistent Controller

Based on the design and requirements of the Persistent Controller service described in section 3.3.2, the Persistent Controller is developed as an integration layer between ACS unit and the database. The Persistent Controller implements the activities as shown in figure 4.2, so whenever it is invoked, one of the branches are executed.

To support this behavior the Persistent Controller implements a data source architecture pattern named *Data Mapper*[8]. This pattern is chosen as it separates the dependencies between the database objects and the system objects. The structure of this pattern and the Persistent Controller is described with a class diagram in figure A.2 in the appendix. In 3.3.2 it is designed that the Persistent Controller offers the IPersistentController interface, which is illustrated in the class diagram by the inheritance of this interface to the PersistentController class.

4.2.1.3 Charging Scheduler

The function of the charging scheduler component is to generate a charging schedule from historical data, as described in section 3.2. These functional requirements have resulted in an activity diagram figure 4.3, which is an illustration of the activities that the Charging Scheduler service performs during the process of generation of a charging schedule.

The architecture design in section 3.3 has resulted in the following detailed design of the charging scheduler component as shown in figure A.3, in the

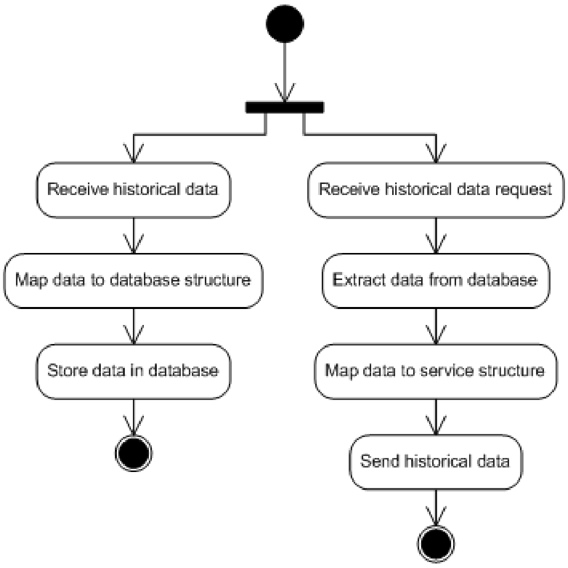


Figure 4.2: The activity diagram of the Persistent Controller

appendix. One can see in the figure that the class ChargeSchedulerService inherits the interface IChargeScheduler, which was described in section 3.3.3. The architecture design deduced that the charging scheduler component required two dependencies of the two interfaces IPersistentController and IChargeController. These dependencies are illustrated in figure A.3 as associations to the interfaces.

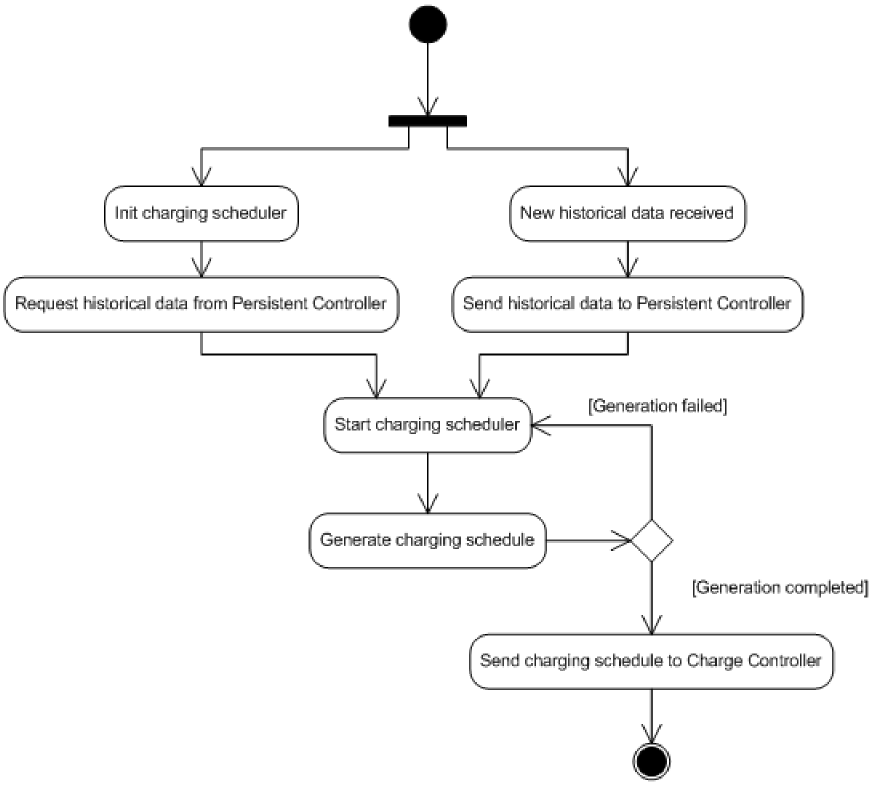


Figure 4.3: The activity diagram of the Charging Scheduler

4.2.1.4 Charging Controller

The Charging Controller is a primary in the ACS unit. The requirements of the Charging Controller, as identified in 3.3.5, are that the service must administrate the execution of the charging schedule and instant charge request. Additionally, the service must manage the monitoring of the battery level. To describe the behaviour of the Charging Controller a number of activity diagrams have been created.

Figure 4.4 shows an activity diagram that describes the activities performed, when a new charging schedule is received. During the execution of the process, when receiving a new charging schedule, the activity called *Start charging schedule* as one can see in figure 4.4 initialises the process described in figure 4.5.

This process performs the execution of the charging schedule, such as extracting and executing these schedule actions. Similar to the process, when receiving a new charging schedule, the *Start charging schedule* can be initialised from the authentication process. Since the EV is not connected to a charging station at the time of execution of the charging schedule service, the process must be started when connecting to a charging station.

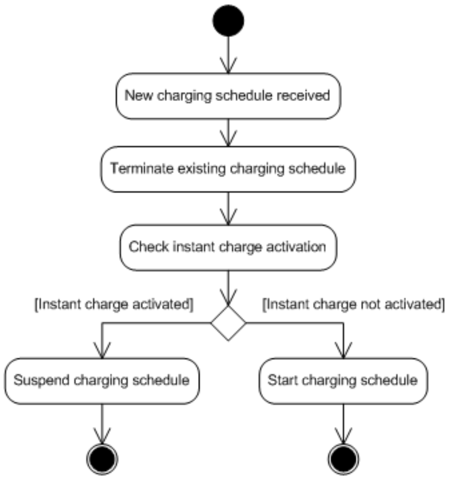


Figure 4.4: The activity diagram of the Charging Controller when a new charging schedule is received

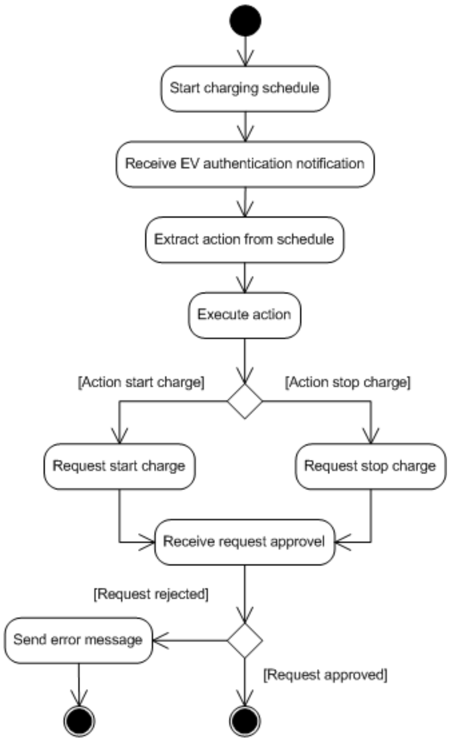


Figure 4.5: The activity diagram of the Charging Controller, when starting the execution of a charging schedule

The process when receiving an authentication or connection notification, is described in figure 4.6. As one can see in the diagram the Charging

Controller must have a notification both according to the connection to a charging station and one for authentication for the Charging Controller to start the execution of the charging schedule.

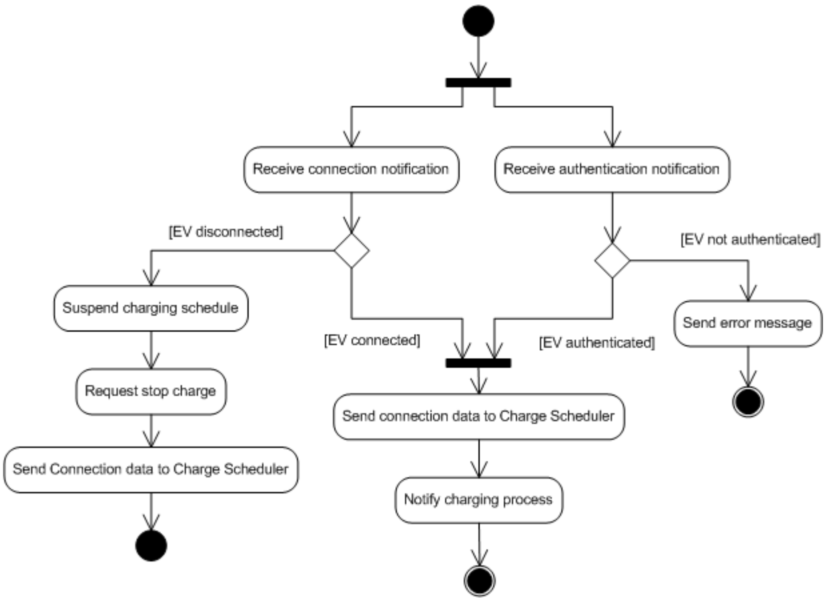


Figure 4.6: The activity diagram of the Charging Controller, when an authentication or connection notification is received

Besides the execution of a charging schedule the EV has the possibility to be requested to start charging. This is a process that is requested by the operator. However, when the Charging Controller receives an instant request it performs the activities described in diagram 4.7.

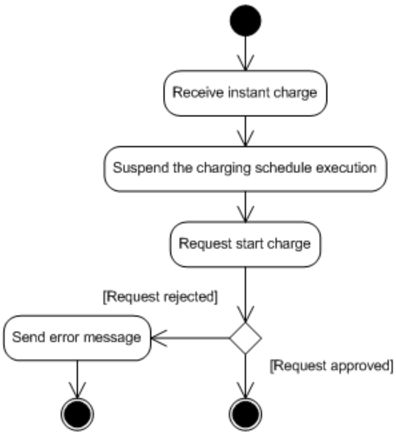


Figure 4.7: The activity diagram of the Charging Controller, when an instant charge is requested

The last task of the Charging Controller is to perform monitoring of the battery level. The process according to the monitoring is repeated every five minutes, where the activities illustrated in figure 4.8 are performed.

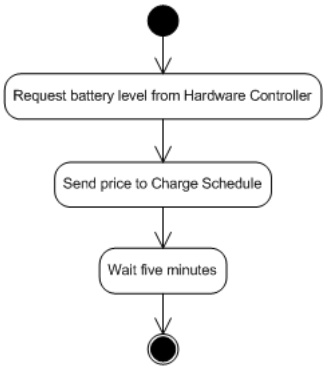


Figure 4.8: The activity diagram of the Charging Controller, when monitoring the battery

All these activities performed in the different processes are carried out across the classes of the Charging Controller. The structure of the Charging Controller is therefore described in a class diagram in figure A.4 in the appendix. This diagram shows the associations between the different classes of the Charging Controller. It also shows the inheritance of the interface that specifies the entry point for the Charging Controller. Additionally, the class diagram shows the associations and dependencies that the Charging Controller has to the interfaces; IChargeScheduler and IHardwareController.

4.2.1.5 Charging Station Controller

The task of the Charging Station Controller as, identified in 3.3.4, is to act as an entry point for the charging station. The functionality of the service is therefore quite simple, as illustrated in figure 4.9, as they primarily include forwarding of information.

4.2.1.6 Hardware Controller

The functional requirements of the Hardware Controller are to work as a virtual component of the hardware in the EV. The service offers functionality for the other services in the ACS unit to request. The activities performed when a request occurs are illustrated in the activity diagram in figure 4.10. A class diagram is also conducted, in figure A.5 in the appendix, which shows the associations and the inheritance of the service interface.

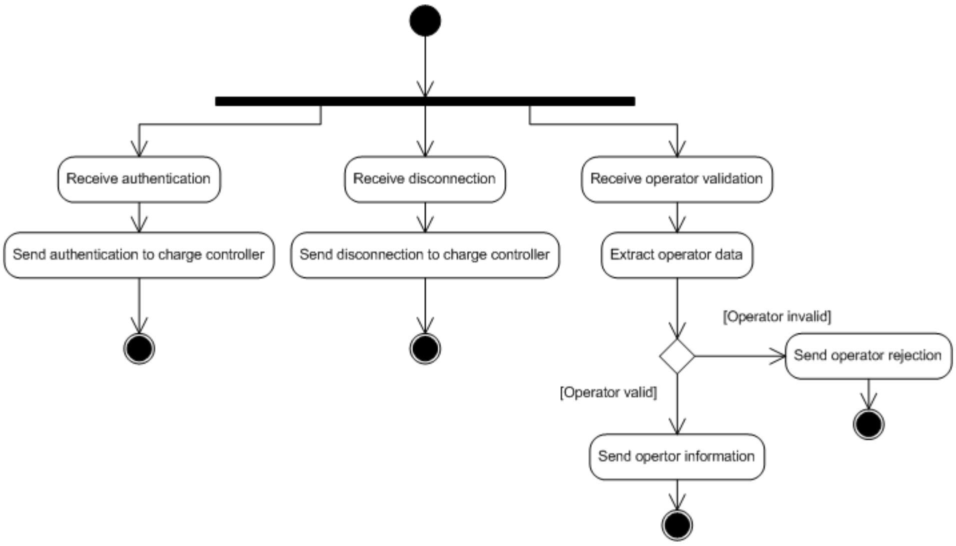


Figure 4.9: The activity diagram of the Charging Station Controller

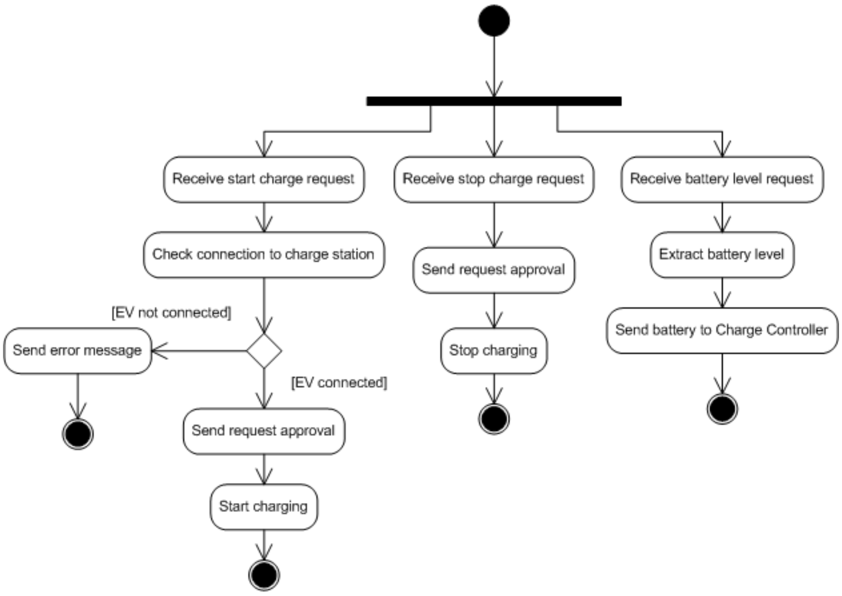


Figure 4.10: The activity diagram of the Hardware Controller

4.2.1.7 User Management

The required functionality of the User Management service mostly acts as an entry point for requests from an operator interface. As mentioned earlier the scope of this thesis is to develop a prototype of a decentralised EV to demonstrate the proof-of-concept. The implementation of the User Management service is therefore omitted.

4.2.2 Environment Services

Aside from the components that are functional within the ACS unit, the ACS unit still require some functionality and input from external services. These services are as described in section 3.2, the *Information Provider*, *Charge Station* and *EV display*. The actual functionality of these services are, however, out-of-scope of this thesis. Though, they are still required for simulation of the prototype and are therefore developed with the minimum of functionality.

4.2.2.1 Service Provider

The Service Provider service has the task, as mentioned in 3.4.3, of ensuring the price signal is available for the ACS unit every five minutes. The Service Provider are in this thesis thereby developed as a service that is made available for the ACS unit to extract a price tag with structure of:

- Expected price
- Actual price
- Timestamp

The price tags that are provided are authentic data extracted from a worksheet that is generated from the public prices of the electrical market.[2]

4.2.2.2 Charging Station

The function of the Charging Station service is to authenticate the EV and the operator when connecting an EV to a charging station. The Charging Station is in this thesis developed as a simple service that manages a connection signal from the EV, which results in authentication of the EV and operator. An actual development of the Charging Station service can be deduced from the proposed design of the conducted system in the thesis Roaming in a Virtual Power Plant Environment [10], as mentioned in 3.4.2.

4.2.3 Graphical Components

Like the other external services the graphical components in this solution are used for the demonstration of the prototype. In this solution a GUI has been developed, which consist of an overview of the different services

---

[2] This worksheet was provided by Peter Bach Andersen

and their log information, where one can follow the flow within the services. Furthermore, the GUI has also a virtualisation of the hardware entities in an EV, such as; ignition, brakes, throttle and socket connection. However, in longer terms the graphical interface as mentioned in 3.4.1 should be able to be integrated in the solution.

4.3 Conclusion

Based on the identified architecture design from chapter 3, this chapter covered the development design of an ACS prototype unit. The design followed the principles of SOA.

The choice of development technology was the Microsoft .Net framework, including Windows Communication Foundation for the communication between the services, since it supports standard security measures. Windows Workflow foundation was chosen to develop the service processes as it provides design friendly development tools. Windows Presentation Foundation was used in the implementation of a graphical interface. These decisions were made based on previous projects within the development of the EVPP, which has been using the .Net framework.

The chapter described the solution architecture, that was designed such that every service was developed in its own assembly to support the SOA architecture. It also made the solution modular and prepared for a later integration in a real EV. The description of the solution architecture resulted in a number of activity diagrams that described the behaviour within each service, from an input is received to it is processed. For the components, which were more complex in their structure, the activity diagrams were supported by class diagrams over the classes in a component.

Based on the conducted work performed in this chapter, problem statement eight, described in 1.3 has been covered.

Chapter 5

Case Studies

The motivation of this thesis is to support the development of decentralised EVs so that they can support the grid and help renewable energy resources to gain a bigger market supply. It requires decentralised charging of the EV population in an intelligent way and the prototype developed in this thesis aims to support the generation of a charging schedule in an EV.

The requirements have led to a number of use cases, which show how the normal use of a decentralised EV is in relation to its surrounding environment. This chapter aims at proving that the identified use cases are supported by the implemented prototype and thereby supports one important element in the concept of a decentralised DER. If the proof-of-concept is successful the final part of the vision will be fulfilled. The content of this chapter covers the problem statement 9 in 1.3 and consists of the two following case studies.

5.1 Case: Proof-of-concept of a Decentralised EV

The purpose of this case study is to test the developed ACS prototype. It includes test scenarios for the main functionalities such as:

- Storing historical data
- Generation of the charging schedule using historical data
- Charging the EV according to the charging schedule

Storing Historical Data

The test scenario *Storing historical data* is performed as a system test, where the flow of the ACS unit is followed when the execution of the ACS unit is started. The preconditions for the *Storing historical data* test scenario are that the services; Information Controller, Charging Scheduler, Persistent Controller are initialised and ready. Figure 5.1 shows the flow across the services, when storing a price in the system, where the green line in the figure indicates the successful test scenario.

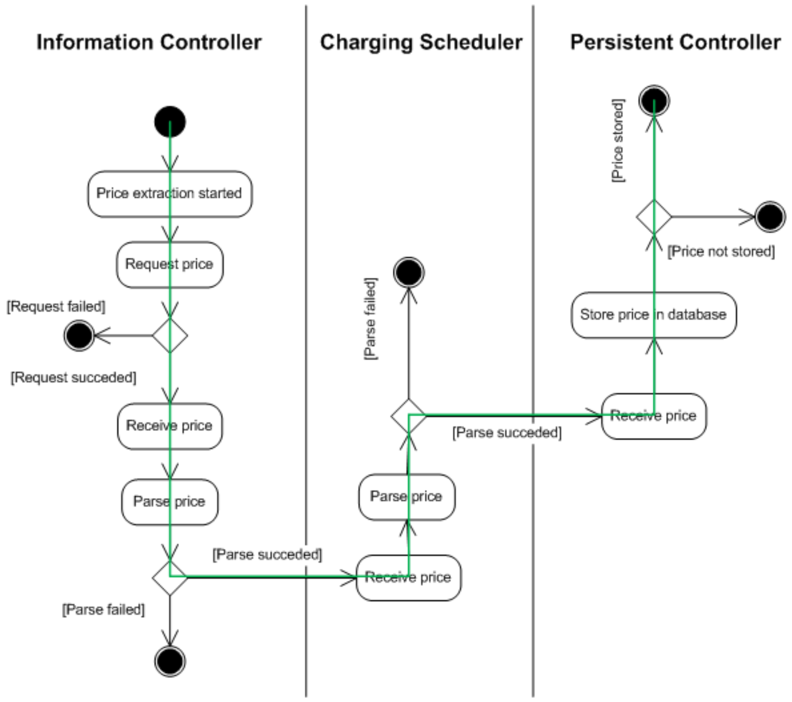


Figure 5.1: The test scenario of when storing historical data

After having executed the test scenario, the test showed that the implemented ACS unit performed successfully, since the solution returns the expected result, which was an extraction of a price signal and stored in the database.

Generation of the Charging Schedule using Historical Data

The test scenario named *Generation of the charging schedule using historical data* performs the test of the generation of a new charging schedule based on the new data plus historical data. A precondition for the test is that services of the ACS unit are started. The flow of the test scenario are described in figure 5.2, which shows the activities across the services Charging Scheduler and Charging Controller. The flow, that makes the test scenario *Generation of the charging schedule using historical data*, successful is shown in figure

5.2 as a green line.

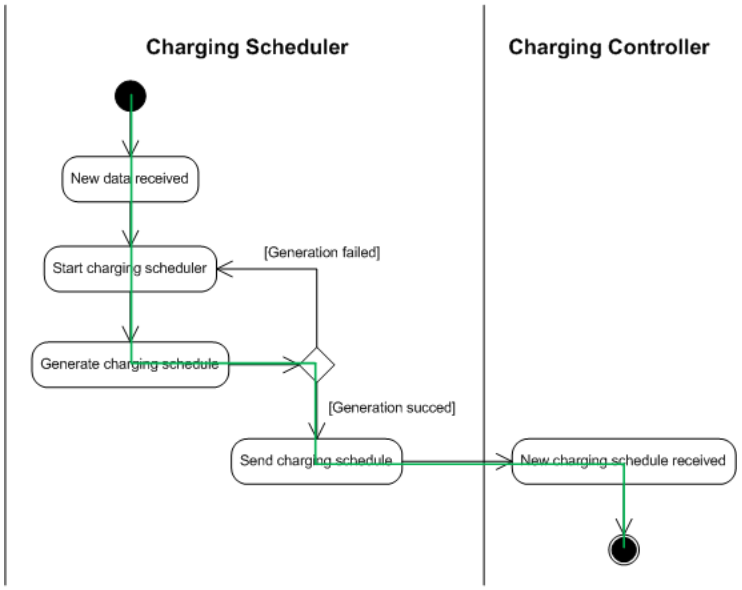


Figure 5.2: The test scenario when a new charging schedule is generated

The performed test was successful as a charging schedule was generated.

Charging EV according to the charging schedule

The *Charging EV due to the charging schedule* test scenario performs a test on the process, when the ACS needs to perform a charging action according to the charging schedule. The flow in figure 5.3 is a description of the path through the services involved, which are the Charging Controller, Charging Station Controller and Hardware Controller. The green line in the figure indicates the successful path of the process.

The flow was partially successful up until the execution of charging action, where it turned out that the EV did not carry the charging as expected.

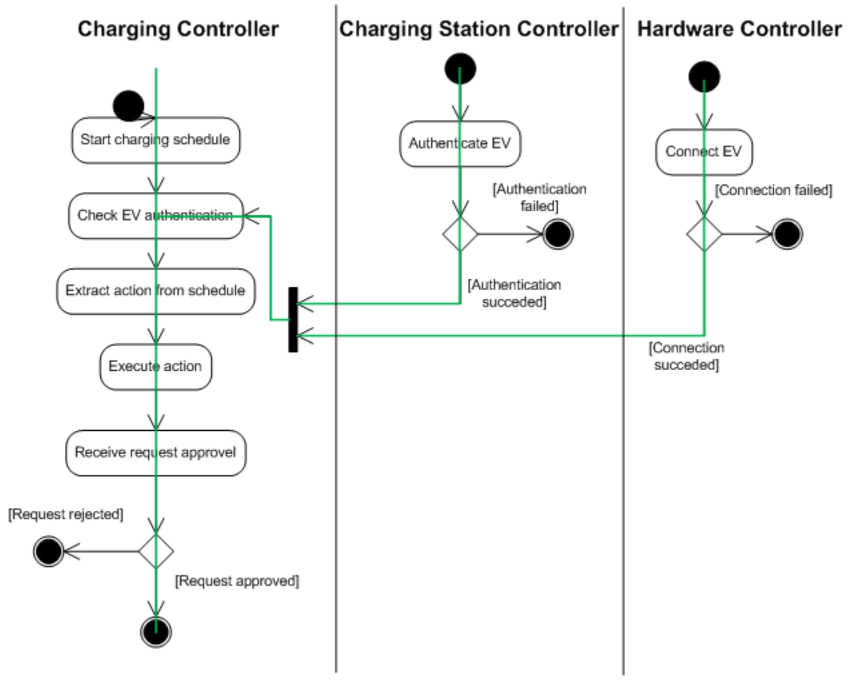


Figure 5.3: The test scenario when charging an EV due to the charging action from a charging schedule

5.2 Case: How the prototype can be implemented in a real EV

The purpose of this case study is to describe how the designed and implemented ACS solution is prepared for integration in a real EV. As described in section 3.2 the architecture is designed using the concepts of SOA. This design choice is followed up by the choice of implement the solution services in each assembly as described in 4.2.

The choice of using WCF for communication between the services induce that they should be prepared for implementation in an EV. The only thing that is needed is to specify the address of the required external service, the Service Provider.

To make it possible to start up the ACS unit after implementation in the EV, a small windows service or web application must be implemented such that this windows service initialise the ACS unit. This is needed since the start up application that starts the prototype in the simulation is a windows desktop application.

Furthermore, it is required of the EV, that it has installed an operation system that supports the Microsoft .Net framework with WCF and WF.

With the choice of a solution design, where every service is divided into each individual assembly, and the choice of the development technology it seems fairly painless to implement the solution in a real EV.

Chapter 6

Conclusion

This chapter concludes this thesis with a conclusion on the overall progress made throughout the chapters 1-5. The problem statements from the project description are revisited and examined to clarify what has been achieved and where in the report they have been addressed. Finally, the thesis ends with an outline of perspectives on future work.

6.1 Final Conclusion

As of today no solution exists, to the knowledge of the authors, that covers all aspects of a decentralised EV. Different fragments of an overall solution are being explored in a detailed, however, somewhat isolated manner. This thesis aimed at investigating the possibility of having a fully decentralised EV and gathering enough information to conduct a proof-of-concept of such solution.

In chapter 2 general trends within the EV industry were explored and it was clear that the industry has high expectations, however, still needs a broader collaboration including governmental institutions. Within the research area the idea of implementing a decentralised EV is still relatively novel. Traditionally, a VPP has been used to facilitate control in a centralised manner. This thesis sought to challenge this setup and look into the limitations of decentralising the EV. A few major challenges related to controlling large EV populations were extensively examined and solved on a theoretic level. The conclusion was that although it might be alluring to think of a fully decentralised EV, the reality is that the VPPs still have a large incentive to continue to collaborate in some form, but to a limited extent compared to the traditional VPP. An analysis was therefore conducted on how much control could be shifted from the EVPP to the EV. Based on this analysis the requirements of such an architecture were identified. At this stage in the thesis the first part of the vision was realised:

The vision of this master thesis is to investigate if it is possible for an EV to be fully decentralised and work with the smart grid of the power network. An analysis of the functional and non-functional requirements of such an EV will be carried out and the design must fulfill the requirements. Finally, two case studies will function as proof-of-concept of the implemented prototype.

The analysis and the requirement specification led to a fundamental design choice in chapter 3. SOA was chosen to ensure full modularity and granularity of the prototype. The interoperability of web services was an important design choice to allow the prototype to interact with services across platforms. The overall design was broken down into individual services, whose dependencies on each other, were described. Some external services identified in the analysis also needed to be incorporated on a conceptual level. References to external reports describing the implementation of an EV user interface and authentication towards the charging station solved this. Furthermore, the critical roles of the service provider and the price signal were discussed. A security proposal was made to protect the integrity and authenticity of the price signal.

A prototype was needed to prove the concept of a decentralised EV as presented in the analysis and to prove that the designed architecture fulfilled a subset of the functional and non-functional requirements. The prototype was tested in two case studies and it can be concluded that the overall architecture, with its required external services, satisfy the initial requirements. The solution is proof-of-concept of the decentralised EV and constitutes a solid foundation for future work. With this statement the final part of the vision has been fulfilled and the goals of the thesis have been met.

6.2 Conclusion on Problem Statements

In section 1.3 the problem statements were identified and have been the goals of the thesis. In the following, each of the statements will be examined to show how they have been realised:

1. **Survey which other projects are being developed and what is state-of-the-art.**

This statement has been covered in general in 2.1, where a PwC study formed the basis for the investigation, with its outlines of market trends, based on interviews with key industry and research resources, see section 2.1.1. The trend of collaborations were investigated and in 2.1.2 and interesting partners were examined. Finally, brand new research from MIT was presented in 2.1.3, presenting the most novel approach of charging the EV.

2. **Analyse the VPP concept and the developed EDISON VPP solution.**

   The VPP concept was successfully investigated along with the EDISON VPP in the section 2.2 and 2.3 respectively. The structures were examined and problems of having central control of the EV population were identified.

3. **Analyse the possibility for an EV to become a fully decentralised DER.**

   To solve the problems related to having the control of the EV population, an extra focus was put on decentralised research to find a theoretic solution. How to make EVs charge in a smart way during the night was presented in 2.4.1 and a model that would allow EVs to sell power to the grid via a CVPP was examined in 2.4. The results were not perfect, but showed promising perspectives, which led to the conclusion that further thesis work could be conducted, within the field of a decentralised EV.

4. **Identify VPP features, which can be used in a decentralised EV.**

   The features, which could be transferred to the design phase and be prepared for implementation, were found in section 2.5.

5. **Identify the functional and non-functional requirements of an EV.**

   In section 2.6.1 use cases and a use case diagram were presented, which were based on the conducted analyses. Based on that, the functional requirements could be extracted, and by examining the domain and use case descriptions, the non-functional requirements were also identified, see 2.6.2. Finally, the surrounding environment came into consideration in 2.6.3 and three requirements were identified.

6. **Based on the identified requirements, design the architecture of an EV.**

   Using the requirements from 2.6 and the conclusions of the analysis, from 2.8, the architecture design was conducted. This resulted in a design proposal using SOA, in 3.1, and continued with a definition of all services involved, 3.2. The requirements for the design of the surrounding environment were out-of-scope for this thesis. However, an analysis on how to integrate solutions from existing work was investigated in section 3.4.

7. **Analyse needed security measures to protect the integrity and authenticity of the received price signal.**

   Based on the price signal, a security analysis was carried out to identify threats in 3.1.1. At a later stage a general security proposal was made to protect the price signal in section 3.4. The choice of SOA and the use of the standard Microsoft .Net platform, with its associated development suites (WPF, WCF and WF), also provides standard procedures of securing web services.

8. **Implement a prototype of an EV solution.**

   The implementation of the prototype is described in chapter 4. The implemented solution aims at providing the functionality described in the use cases in 2.6. This was achieved by generating activity diagrams that described the process flow within the solution.

9. **Create two case studies to prove the successfulness of the solution according to the objectives of the thesis.**

   The prototype has been examined through a series of tests in the first case study to validate use case functionality. The conclusion of the first case study showed that most use case functionality had been implemented. The case studies are described in chapter 5. The second case study concerned an elaboration on how the solution could be implemented in a real EV. It was concluded that the solution under the right, but reasonable, circumstances could be implemented in an EV. The overall positive results of the two case studies prove that the work related to the prototype has been worthwhile, as it can be used as a foundation for future work and research, within the area of a decentralised EV.

6.3 Future Work

The project scope has conveniently limited the amount of work, however, it is important to focus on what can be done to enhance the solution of this thesis, but also what can be done in surrounding environment.

Collaboration

As identified by PwC, strong collaborations are needed to continue the development of EVs. Standardisation, in general, is important, but also collaboration across industries, universities and governments is needed to boost the infrastructure and provide incentives, so that when a person wants to buy a car, he/she is more likely to choose an EV. There is still a long way to go:

Price Signal

Today there is no standard price signal. However, the importance and impact of the final design influences, not only the EV operator's wallet, but can also prove to control the EV population and ensure that the grid constraints are not violated.

Battery and Charging Procedure

There is no battery standard nor is there a final charging procedure standard. If the infrastructure is to support a large EV population, then the fastest way to success would be to support one charging standard i.e. one type of charging socket.

Research

There is no doubt that much research is still needed across many different aspects of the decentralised EV. The offset after this thesis could be in this direction of more decentralised control and of more V2G technology:

Decentralised Control

The research presented in the thesis involves rather simple EV populations. More extensive research is needed to investigate how a real scenario, with many different types of EVs and preferences impacts, the Nash equilibrium. The current research does not solve the problem of unforeseen events. If a change occurs, such as a power failure, the control needs to update the population accordingly and preferably as close to real-time as possible.

CVPP or other V2G technology

More research of the CVPP concept is needed to design a better solution, which allows the EVs to provide energy to grid. It could also be that other ways of enabling V2G can be designed, so that they become completely independent from the VPP.

Implementation and test

The prototype developed in the thesis is a basic proof-of-concept. Much work lies ahead until it can be implemented in a real, full scale scenario. The following is a list what needs to be implemented:

- User interface implementation
- Authentication and roaming implementation

- Service Provider implementation
- Include support for V2G
- Implementation in a real EV
- Large scale tests

References

[1] A. B. Pedersen S. You B. Poulsen J. Oestergaard A. Aabrandt, P. B. Andersen. Prediction and optimization methods for electric vehicle charging schedules in the edison project. 2011.

[2] Baris Özdil Andreas Kargård Olsen. Prototype for a iec 61400-25 compliant generic server. Technical report, Technical University of Denmark, 2006.

[3] Ricardo Balduino. Introduction to openup (open unified process). http://www.eclipse.org/epf/general/OpenUP.pdf, August 2007. Accessed 03 October 2011.

[4] Paul C. Brown. *Implementing SOA - Total Architecture in Practice*. Addison Wesley, 2008.

[5] David L. Chandler. New battery design could give electric vehicles a jolt. http://web.mit.edu/newsoffice/2011/flow-batteries-0606.html, June 2011. Accessed 03 October 2011.

[6] C.Binding D.Gantenbein B.Jansen O.Sundström P.B.Andersen F.Marra B.Poulsen C.Traeholt. Electric vehicle fleet integration in the danish edison project - a virtual power plant on the island of bornholm. Technical report, IBM Research - Zurich & Technical University of Denmark, January 2010.

[7] Danish Agency for Science Technology and Innovation (fi.dk). Spir - strategic platform for innovation and research. http://www.fi.dk/nyheder/nyheder/2010/spir-strategic-platform-for-innovation-and-research/SPIR%2026.%20januar%202010.pdf, January 2010. Accessed 29 June 2011.

[8] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison Wesley, 2005.

[9] R. Kota A. Rogers N.R. Jennings G. Chalkiadakis, V. Robu. Cooperatives of distributed energy resources for efficient virtual power plants. *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011), Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 787-794.*, 2011.

[10] P. Wind J. Hansen. Roaming in a virtual power plant environment. Technical report, Technical University of Denmark, 2011.

[11] P.B.Andersen B.Poulsen M.Decker C.Traeholt J.Oestergaard. Evaluation of a generic virtual power plant framework using service oriented architecture. *2nd IEEE International Conference on Power and Energy (PECon 08), December 1-3, 2008, Johor Baharu, Malaysia*, 2008.

[12] Politiken.dk. Tyskland dropper atomkraft i 2022. `http://politiken.dk/udland/ECE1295163/tyskland-dropper-atomkraft-i-2022/`, May 2011. Accessed 30 May 2011.

[13] pwc.com. Collaboration among industry participants is critical to the success of electric vehicles, according to pwc survey. `http://www.pwc.com/us/en/press-releases/2011/collaboration-among-industry-participants.jhtml`, May 2011. Accessed 15 June 2011.

[14] Winograd Singhal and Scarfone. Guide to secure web services. `http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf`, August 2007. Accessed 20 September 2011.

[15] Ketan Singla. Interface - enabling user control over electrical vehicle charging. Technical report, Technical University of Denmark, April 2011.

[16] B.Poulsen S.You, C.Traeholt. Generic virtual power plants: Management of distributed energy resources under liberalized electricity market.

[17] TechNewsWorld. Google and ford to guzzle data so cars can sip gas. `http://www.technewsworld.com/story/72432.html?wlc=1308045794`, May 2011. Accessed 3 June 2011.

[18] WAtoday.com.au. Benz rejects battery swapping. `http://watoday.drive.com.au/green-motoring/benz-rejects-battery-swapping-20110112-19n78.html`, January 2011. Accessed 17 June 2011.

[19] www.risoe.dtu.dk. Research and industry heavyweights in partnership for the intelligent power grid of the future. `http://www.risoe.dtu.dk/en/News_archives/News/2010/1214_SPIR.aspx`, December 2010. Accessed 29 June 2011.

[20] I. Hiskens Z. Ma, D. Callaway. Decentralized charging control for large populations of plug-in electric vehicles: Application of the nash certainty equivalence principle. *2010 IEEE International Conference on Control Applications Part of 2010 IEEE Multi-Conference on Systems and Control Yokohama, Japan, September 8-10, 2010*, 2010.

[21] I. Hiskens Z. Ma, D. Callaway. A decentralized mpc strategy for charging large populations of plug-in electric vehicles. *Preprints of the 18th IFAC World Congress Milano (Italy) August 28 - September 2, 2011*, 2011.

Appendix A

Appendix

This appendix contains the different class diagrams that describes the technical design of the prototype, and represents the structure of the assemblies that are developed for the ACS solution.

A.1 Class diagram of Information Controller

The class diagram of the assembly Information Controller

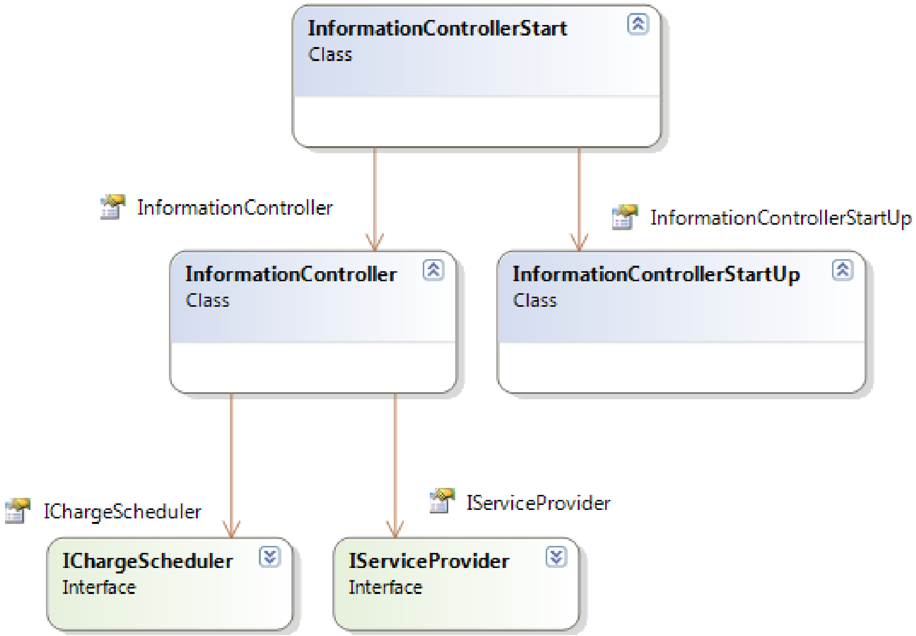


Figure A.1: Class diagram over the Information Controller component

A.2 Class diagram of Persistent Controller

The class diagram of the assembly Persistent Controller

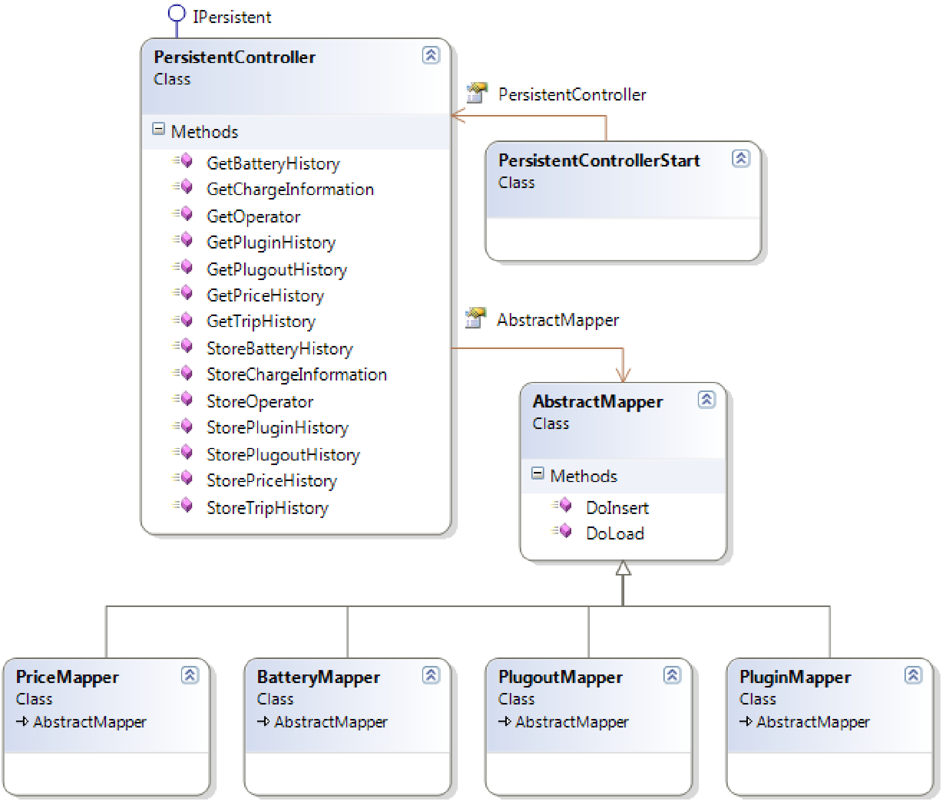


Figure A.2: Class diagram over the Persistent Controller component

A.3 Class diagram of Charge Scheduler

The class diagram of the assembly Charge Scheduler

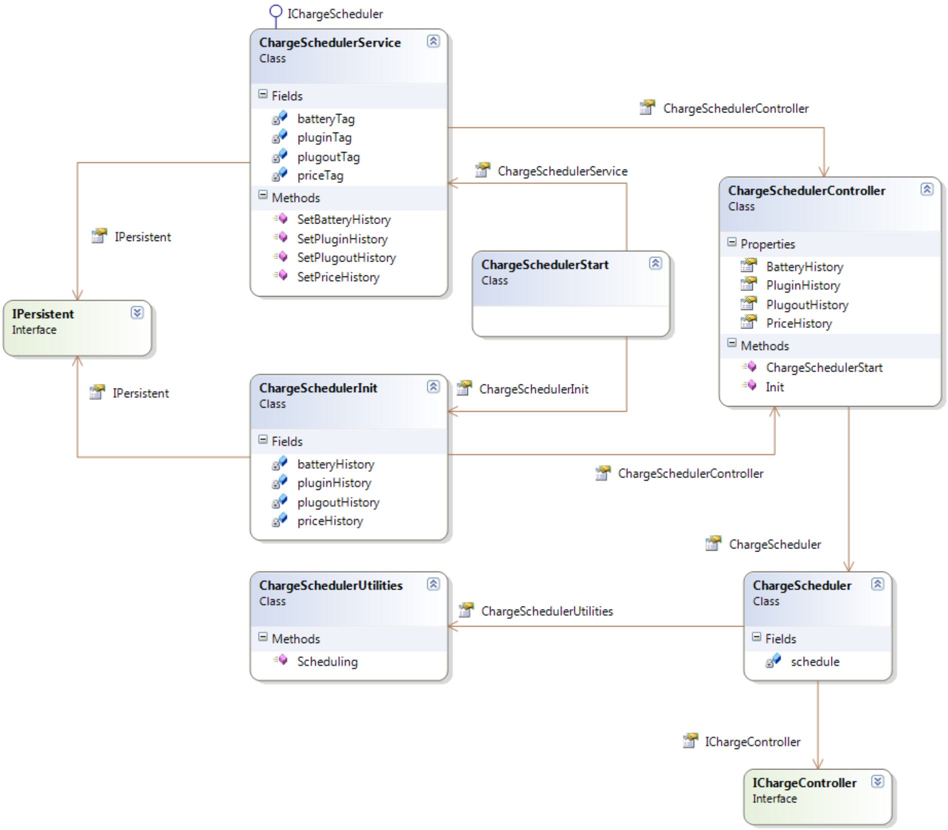


Figure A.3: Class diagram over the Charge Scheduler component

A.4 Class diagram of Charging Controller

The class diagram of the assembly Charging Controller

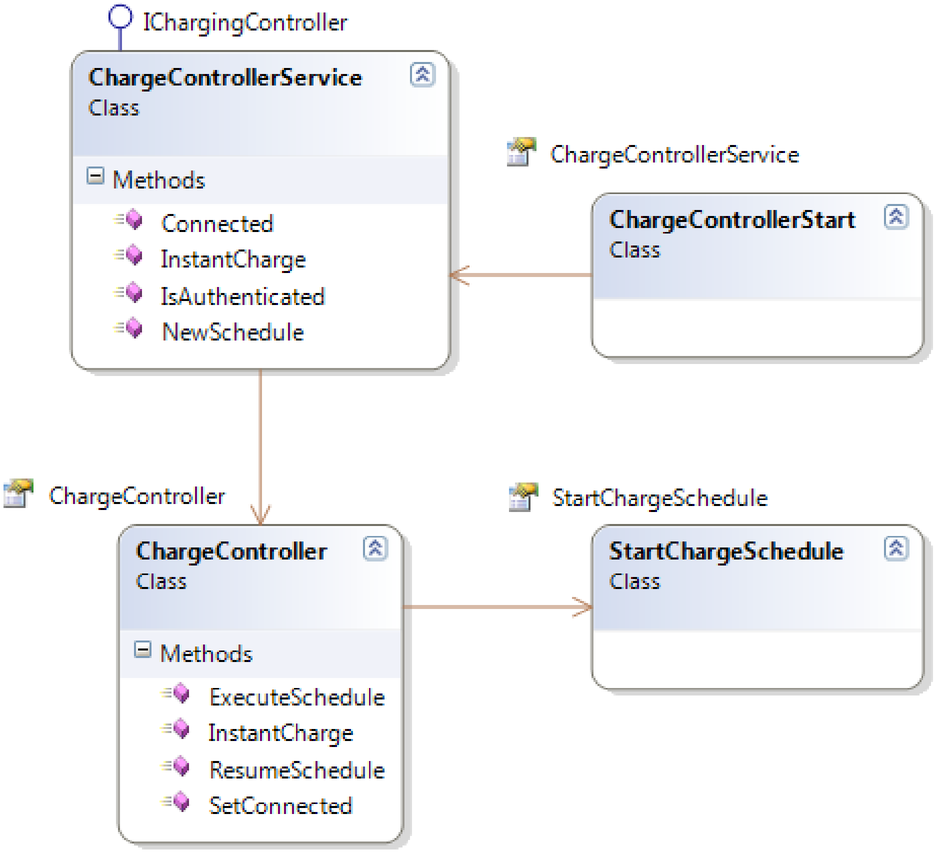


Figure A.4: Class diagram over the Charging Controller component

A.5 Class diagram of Hardware Controller

The class diagram of the assembly information controller

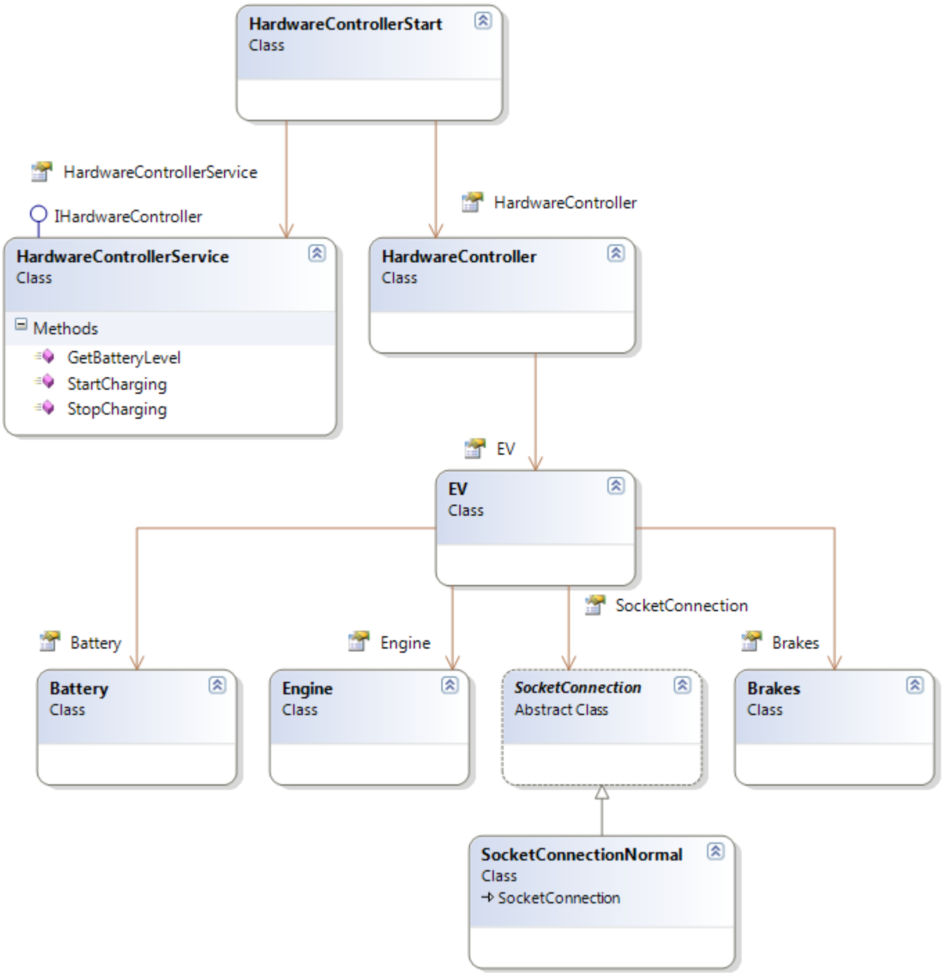


Figure A.5: Class diagram over the Hardware Controller component

www.imm.dtu.dk