Salihovic Jasmin, BSc

Design and Implementation of a mobile Smart Meter application

Master's Thesis

to achieve the university degree of

Diplom Ingeneur

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Scerbakov Nikolai

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Lindstaedt Stefanie

Graz, April 2019

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ____________________ ________________________

Date Signature

Abstract

Recently, the desire to optimize Europe's energy supply has become very prominent due to economic and ecological reasons. Therefore, a smart grid is being created and expanded throughout Europe, but smart meters are needed to ensure everything is functioning properly. In Austria, the smart meter rollout, the installation of smart meters in households, has been a major project in the past couple of years. With their help, it is possible to create a detailed analysis of the energy consumption and the amount of energy that is needed in a certain area, since data is being sent to the energy supplier in 15-minute intervals. However, for a successful installation and communication on-site, it is crucial to enable a link between the meter and the energy supplier. The focus of this thesis is the development of a module which should solve the problem mentioned.

Hitherto, the energy supplier's field engineers had to transfer all data from the meters manually to a software solution on a mobile device. The solution underlying the master thesis enhances the existing software solution and enables the communication with the meter reading device. With just one click it is now possible to obtain and/or send all relevant information regarding energy usage, which in turn means that all advantages of smart meters can be harnessed.

Kurzfassung

Aufgrund von ökologischen und ökonomischen Gründen ist der Grundgedanke der Optimierung der Energieversorgung in Europa groß. Deshalb wird an der Ausweitung des intelligenten Netzes (smart grid) gearbeitet. Für die reibungslose Funktion desselben, werden intelligente Stromzähler (smart meter) benötigt. In Österreich ist das Smart Meter Rollout, die Installierung von intelligenten Zählern in Haushalten, seit einigen Jahren im Gange. Mit deren Hilfe wäre unter anderem die genauere Analyse von Informationen bezüglich Verbrauch, dem Ort und der Menge an benötigter Energie möglich, da diese im 15-Minuten Intervall Daten an die Energieversorger senden können. Doch für eine erfolgreiche Installierung und Kommunikation vor Ort muss eine Verbindung zwischen dem Zähler und dem Energielieferanten gewährleistet werden. Der Fokus dieser Diplomarbeit liegt in der Entwicklung einer Lösung, die das oben genannte Problem lösen soll.

Außendienstmitarbeiter der Energieversorger mussten in der Praxis bis dato manuell die Zählerdaten in die bestehende Softwarelösung auf einem mobilen Gerät übertragen. Die der Diplomarbeit zugrundeliegende Lösung erweitert diese bestehende Softwarelösung und ermöglicht die Kommunikation mit dem Auslesekopf. Mit nur einem Klick erhält und/ oder sendet der Mitarbeiter nun alle relevanten Details zur Energieverwendung und den Eigenschaften des eingebauten Zählers, wodurch alle Vorteile eines intelligenten Zählers genutzt werden können.

Acknowledgements

I would like to thank God for giving me the chance to work on and finish this thesis. Special thanks goes to my thesis advisor Dr. Nikolai Scerbakov for providing help during the writing process. I must express my very profound gratitude to my parents, wife and daughter who have a special place in this whole story and who pushed me and gave me the needed energy. I am also grateful to my brother and parents in law and my friends and familiy for supporting me.

Thank you everyone!

Contents

List of Figures

1 Introduction

Electricity has been a key component in humans life since the end of the 19th century. With the AC ampere hour meter, introduced by Oliver B. Shallenberger[1], the era of electric metering began. Since then, the traditional electric meter has become prevalent in society. These conventional meters use an aluminium disc[2] to measure the electricity. Nowadays, they are digital, but they also have many limitations.

The limitations of the meters that are used nowadays will be discussed in the thesis. They are many reasons why to use something else instead of the used meters.

Due to the humanity's observation that traditional and conventional sources of electricity seem harmful to the environment, the leading countries of the world decided to tackle the reduction of carbon dioxide emissions, as well as other damaging substances.

[1] Dyer, 2001.

[2] Thiele, 2019.

This is the reason why the European Union has set an aim until the years 2030 to 2050 to provide electricity solely by using renewable energy. Therefore, the focus lies on solar panels (producing energy by using the sun's rays), hydro plants (producing energy by using water) and wind turbines (producing energy by using wind).[3]

The objective is to completely shut down nuclear power plants (producing energy by using the heat from nuclear reactions) and thermal power plants (producing energy by burning fossil fuels) and replace these by the above mentioned renewable energy sources. To fully optimize the consumption process and downscale the energy loss, it is necessary to be able to analyze how much electricity is being used at any given time in which specific households or factories. This is the only feasible way to accordingly allocate the energy. Therefore, new, so called smart meters have been brought into life. These are directly connected to the suppliers who have the possibility to access real time data whenever they want. The meters also send them information on the consumption in a 15-minute interval.

However, this is not an easy task. In order to materialize the project successfully, the infrastructure has to be adapted and optimized accordingly. Up until now, the only way information and electricity were able to flow in one direction: from the suppliers to the consumers. Consequently, the existing infrastructure in Europe must be changed to enable the flow in both directions: from supplier to consumer and vice versa. The goal is to create a

[3] Cederlöf, 2018.

unique smart grid in the European Union, which could be connected among all the member states.

Since the data concerning the electricity consumption is collected every fifteen minutes, the data collection is so immense that it can be called "big data". This term refers to data, which is too big, too complex, too fast-paced and too weakly structured to be evaluated using conventional methods of data processing. This matter will be discussed further in the sections below.

Introducing smart (95%) meters in Austria will have to be finished until 2022[4]. All of the energy suppliers in Austria have been working on the implementation of smart meters over the past few years. I was given the opportunity to work with one of the suppliers and help them with this process. Specifically, I worked on the smart meter rollout in certain parts of Austria. Due to specific restrictions which will be discussed in detail in this thesis, a solution had to be found with which it would be able to communicate with the smart meters. The hypothesis is that the smart meter can not communicate with the existing applications that are used by the energy suppliers. This is the reason why I started to work on a solution that will expand the actual environment with a module that is called Meter Communicator. This module should enable the communication with the smart meters and is also the main topic of this master's thesis.

[4] Digitalisierung und Wirtschaftsstandort, 2017.

1.1 Motivation

Like it is described in the chapter above all of the energy suppliers in Austria have been working on the implementation of smart meters. The suppliers and all companies that are currently working on the electric grid have some pressure to prepare the grid and do all other necessary steps for a successful rollout of smart meters in whole Austria. There is an act in the Austrian law that forces the suppliers to hurry up the rollout.[5] Due to many restrictions, at this point there will be listed just one of them, it is not possible to use the full functionality and features that are given with a smart meter. At this moment, it is impossible to communicate remotely with all installed smart meters. The reason or more precisely the restriction why it is like this nowadays is lack of the needed infrastructure for remote connection and communication with the newly implemented smart meters. During the installation of a smart meter in a household the grid is not able to ensure a remote communication.

To ensure that any of the many functionalities and features that a smart meter has are used, it was necessary to develop a solution that allows to communicate with the newly installed smart meter. This was my motivation for creating an application that will ensure this communication in the given environment.

[5] Digitalisierung und Wirtschaftsstandort, 2017.

In addition, in this thesis the structure and existing environment that is used nowadays will be described. Afterwards the focus of the thesis will be on goodies and features that are given with the smart meters compared to the standard electric meters that are used nowadays in most of the households. The environment where the implementation of this application was embedded will be also described in a part of this thesis. In addition, the thesis will examine the functionalities and requirements. It will be also information about the design and the technology stack. One of the last chapters will be used to describe the implementation of the solution. In the end the conclusion will be presented together with ideas for future work.

2 Existing technology overview

In this chapter the focus will lie on the existing technologies that are used nowadays. There will be some information about the existing electric grid and the plan to build a functional smart grid with all its components. The smart meter itself and benefits and restrictions will also be discussed. Last but not least it will contain some sections where the main focus is on the existing environment where the practical part of the thesis is implemented.

2.1 Existing electric grid

The electric grid is the biggest and most complex system in the world that has been built by humans so far. The system raised and developed pretty centralized because historically, the way how to create electric energy was the same for most civilizations/nations. The most common way is to build few big power plants and from there the grid to the cities and all consumers. To generate enough energy at one location, the countries build huge thermal energy plants, which use fossil fuels like coal, or nuclear power plants.[1] Both of these kinds of power plants are able to generate a huge amount of electric energy and that is the reason why the countries build them. Few of these power plants are big enough to supply big cities or whole regions with electric energy. The existing electrical grid is built from such power plants to the suppliers and that is the reason why it is centralized nowadays. The figure 2.1 shows the electric grid in Germany in year 2012 with their power plants that generate electric power above 1000MW.

The conclusion after looking at the graphic above is that the electric grid in Germany is built from the big power plants in the west of the country to the suppliers in central Germany and so on. That shows that the grid is pretty centralized.

Additionally, the existing grid enables only one-way flow of electricity and data, namely from the supplier to the consumer. This kind of energy consumption results in a constant pollution of the environment. Burning fossil fuels emits huge amounts of carbon dioxide, which is the main cause for the greenhouse effect and hence also influences global warming. In addition, various heavy metals and particulate matter are released, which can cause numerous health problems and illnesses for those people living in the surrounding areas. This is why alternatives to thermal energy plants have to be found. In comparison to thermal plants, nuclear power plants do not produce such high carbon dioxide emissions. However, since the disposal of nuclear waste is environmentally harmful, the risk of power

[1] Gonen, 2014.

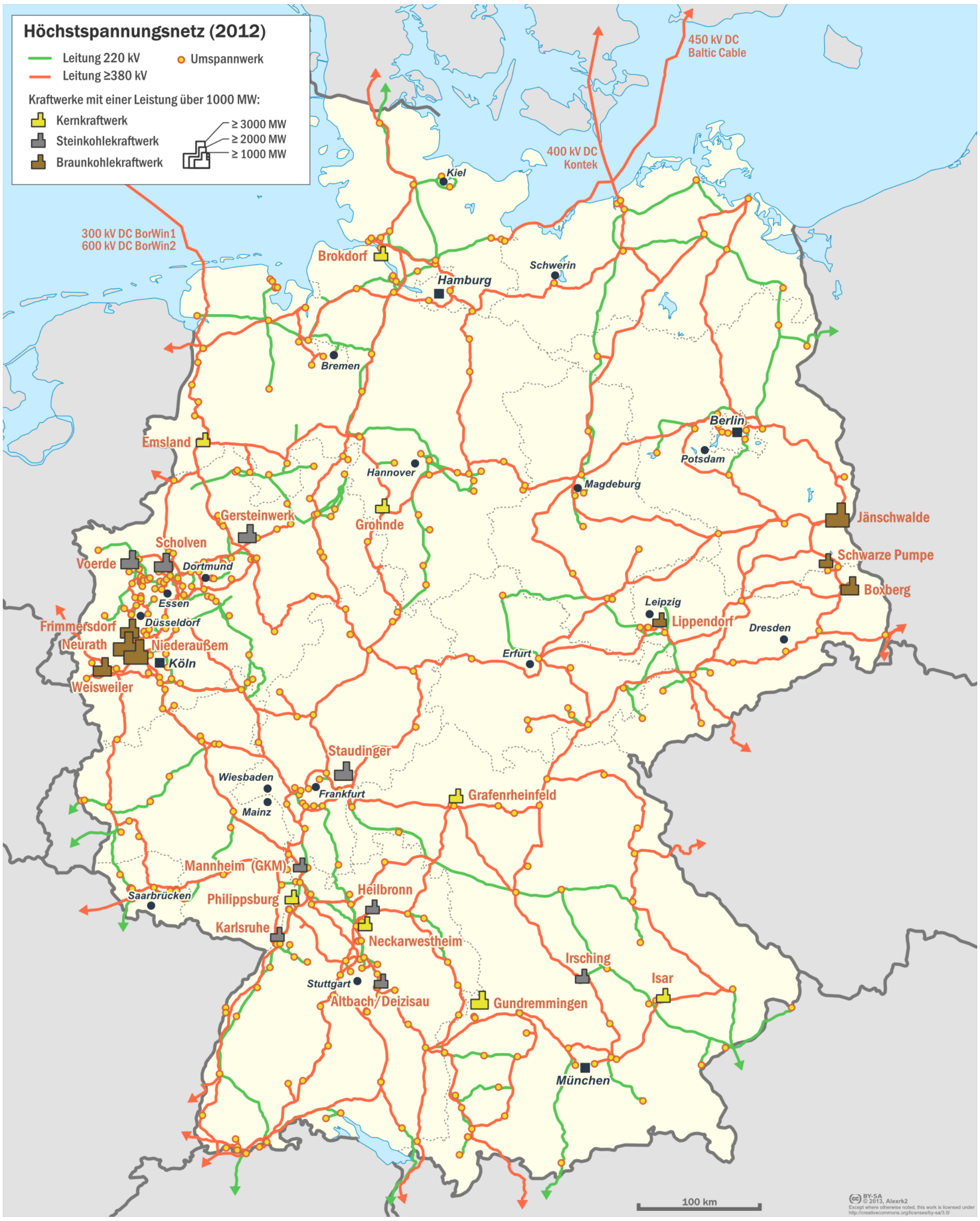


Figure 2.1: High-voltage grid map of Germany with high-voltage power lines above 220 kV and power plants above 1000 MW electric power. (2012)
Source: Alexrk2, 2018

plant failure or accidents is high, which is why a lot of countries decide against installing such power plants. Consequences of failures and accidents in nuclear power plants can best be seen in the examples of Chernobyl and Fukushima. Still nowadays, the aftermath of these two ultimate MCAs can be observed among people's and animals health as well as the vegetation.[2]

As a result, numerous countries change to renewable energy sources, like water, sun and wind energy. For such a step, it is necessary to change, adapt or (re-)build the grid, which is more or less already centralized. The above mentioned renewable energy sources do not produce constant and same amounts of electric energy like thermal or nuclear power plants. Besides, they need to be built very cautiously. For instance, it is irrational to build wind power plants in areas with low and rare winds, or photovoltaic systems in areas with insufficient sunshine hours.

Ergo, it is essential to link various parts of the country with an appropriate infrastructure and grid, which should be decentralized in order to facilitate an exchange of the prevalent energy. At the moment this grid is being worked on, which is called smart grid.

2.2 Smart grid

A Smart Grid System is: "A system which includes a variety of operational and energy measures- including smart meters, smart appliances, renewable

[2] Pachauri, 2015.

energy resources, and energy efficiency resources."[3]

"The smart grid refers to an advanced communication and information infrastructure that enables optimization in energy production, transmission, distribution and storage. Other benefits involve system management automation, educated planning, lower costs and effort, and electricity system reliability improvement."[4]

The smart grid brings along various advantages like efficiency, better control and storage of electric power and the possibility to use energy generated from renewable resources (i.e. wind and solar power). Another major benefit is the fact that considerably less energy gets lost when using the smart grid. Only in India[5] a loss of about 40% can be seen during the energy transfer from supplier to consumer.

To create a functioning smart grid in Austria and Europe huge investments are needed. Money is the key to optimizing and expanding the existing grid. The main problem why the current grid cannot be used adequately, is that the main sources of energy come from huge power plants, like thermal energy plants, which use fossil fuels like coal, or nuclear power plants. These vast power plants can generate enough energy to supply thousands of households and industries, even whole cities and regions. This is the reason the grids are only constructed from the power plants directly to the

[3] Commission, 2008.

[4] IEEE Smart Grid Big Data Analytics, n.d.

[5] Ferrari and Singh Dhingra, 2009.

cities and regions, where this energy source is used. It can therefore be said that the power supply system we have now is centralized.

If renewable energy sources should to be used henceforth, it is essential to decentralize the system. Looking at the case of Germany it can be seen that the north has a lot of strong winds, which represent the perfect ground for wind farms. The south and central parts, however, seem to have more sunny days over the year, which makes them appropriate for solar energy power storage. Also, industrial usage of energy is the highest in the west and south of Germany.[6] Consequently, they are not capable of producing enough energy only with renewable resources for the required demand, which of course could be solved, once the wind power from the north reaches the south. Therefore, advancing the grids is needed as quickly as possible.

The same problems account for the rest of the European Union, which can be tackled the same way. The ultimate goal is to create a smart grid that works in all of the member states, preferably cross-border. The concept aims to collect electric energy via winds in the north and via sun rays in the south of the European Union. Obviously, a lot of money will go on the realization of the smart grid project but in the long term it will pay off and result in great savings and conservation of resources and money, accompanied by manifold opportunities for the member states.

According to the European Commission, the transition towards a more sustainable and secure energy system would require an investment of

[6] *Energieverbrauch der Bundesländer* 2019.

200 billion Euro in the EU for generation, networks and energy efficiency developments.[7]

The energy consumption must hence be possible to be measured at any time in any part of a country or the EU. This should preferably be done every 15 minutes and be sent to and collected by the supplier in order to secure exact extraction of data on the energy consumption. To be able to do that, in return, each household and industrial company have to be equipped with smart meters. The exact details on how these work will be discussed in the following chapters of the thesis.

One of the characteristics of a smart meter in the smart grid is that a data exchange between supplier and consumer and vice versa needs to be enabled. Until today the data has only been flowing in one direction, namely from the supplier to the consumer. Again, enormous investments and ameliorations to the existing electricity grid must be performed. To facilitate useful controlling, monitoring and coordination of these smart meters, the constant exchange of data is obligatory, which conversely results in big data analysis to guarantee maximum efficiency in reading power consumption. In order to enable this, it is necessary to install a big number of "smart" devices, which would make the grid more efficient and enable the above mentioned - namely an exchange of data in both directions (supplier to consumer and vice versa).

[7] Zachariadis, 2018.

2.2.1 Smart grid benefits

There are many benefits that come with the smart grid. Some of them are[8]:

- Reliable electric grid with better power quality
- Better safety and more cyber security
- Less energy loss, more efficiency
- Better for the natural environment - less greenhouse gases
- Reduction of costs for supplier and consumer

2.3 Existing meter

The electric meter is a device which measures the consumption of electric energy in a household or company. Electricity is measured with the unit kilowatt hour (kWh). At the moment, households and factories use different forms of electric meters.[9]

2.3.1 Electromechanical induction meter

The most common type of electric meters is an electromechanical induction meter. "The electromechanical induction meter operates by counting the revolutions of an aluminium disc which is made to rotate at a

[8] (EPRI), 2011.

[9] Pflier, 2013.

speed proportional to the power. The number of revolutions is thus proportional to the energy usage. It consumes a small amount of power, typically around 2 watts."[10] There are also electric meters that measure two or more charges/rates.

2.3.2 Electronic meter

The next generation of electric meters that are used nowadays are electronic meters. They do not operate on the same principle as electromechanical induction meters. These types of meters do not contain any of the mechanical moving parts like the electromechanical induction meter. It is based on digital micro technology and controlled by an application specified integrated circuit. There are also analog circuits, voltage transformers and other parts that helps to measure electric power. The electronic meters contain an LCD or LED display that shows the actual power consumption in kilowatt per hour.[11]

In the following section the newest technology of electric meters will be described.

2.3.3 Smart meter

A smart meter is an electric meter that is connected to the suppliers via the smart grid. It sends information about the energy directly to suppliers

[10] *Mechanism Of Electromechanical Induction Meter Information Technology Essay* 2018.

[11] D.Mohankumar, 2011.

and allows them to remotely collect data in real-time. The smart meter also allows consumers to monitor their energy consumption in real-time and optimize the energy use accordingly in their household. The energy suppliers do not need to send a meter inspector to check the usage of a consumer any more.

Smart meters allow communication and exchange of files over few interfaces. Some of these interfaces that the smart meters support are: Infrared, S-Zero-Interface, dry contact, Meter-Bus, KNX, 20-mA-Current loop (connected with GSM-, PSTN-modem).[12][13]

There are many ways to connect a smart meter to the smart grid and to enable the communication directly with the energy supplier. One of them is connecting via Power Line Carrier (PLC), second, using a wireless modem (GSM or GPRS), third, using a permanent internet connection (ADSL).[14]

Smart Meter and Big Data

Due to the constant exchange and gathering of information on the consumption, the data can also be regarded as big data. Big data are huge amounts of information that come from households and the energy industry and precisely smart meter systems, which are stored, processed and analyzed with special solutions. In this project a solution from an external partner

[12] *Smart Metering mit KNX* 2018.

[13] Steinger, 2017.

[14] Kitzler, 2013.

is used. There will not be any details about this solution, because it is not really relevant to this project.

Smart meter benefits[15]

The aim is to show the benefits of using smart meter. For this reason, the advantages will be split up in various categories: benefits for the consumer, benefits for the supplier and finally, benefits for the environment.

Benefits for the consumer of electric energy:

- detailed analysis of the energy consumption
- minimization of consumption and cost cutting with the help of the analysis
- reduction of blackouts and other problems in the grid

Benefits for the supplier:

- efficient use of power resources
- better monitoring and controlling of power consumption
- reduction of power loss in the grid
- the manual meter reading by a supplier's technician is not needed anymore
- real-time data of power consumption for better balancing and optimizing

[15] ESMIG, 2018.

- dynamic pricing, the cost can be lower and higher, it will depend on demand and supply
- supports better scheduling of the construction of new power plants

Benefits for the environment:

- better allocation of energy
- less losses lead to reduced energy needs, which result in a reduction of capacities of fossil fuel power plants
- reduction of pollution by cars that technicians use to reach customers for meter inspection

2.4 Existing environment

In this section the focus lies on describing an actual environment from an Austrian energy power supplier. This supplier is working on a smart meter rollout. It is necessary to connect the existing applications and solutions that are used by the power supplier with the application that is used for communication and exchanging information and files with smart meters.

Like it is shown in the figure 2.2 - the existing environment that is used by the supplier is very complex. There are several parts in the environment that are relevant for the new solution that will be described in detail in the coming chapters of this thesis.

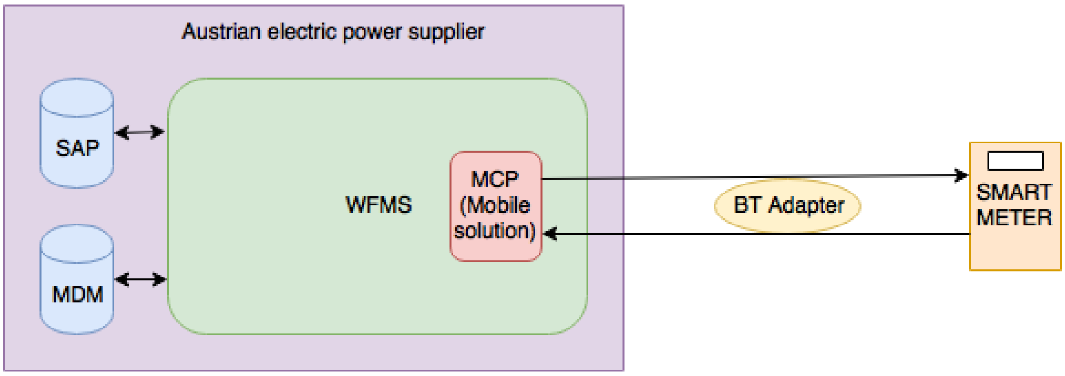


Figure 2.2: Existing environment schema - own illustration

The new solution does not change the existing. It just extends the actual environment and gives the possibility to communicate with the installed smart meter.

The left part of the picture portrays the existing infrastructure and environment where the new solution needs to be embedded. This part is the main theme of this master thesis.

The Austrian electric supplier is using many different solutions for organizing their processes. They are using an SAP solution optimized for utilities. All the organizational issues will be declared, handled and planned in this part. This system is the lead system that fills the other system with information. A huge amount of information that is afterwards used, handled and created will be returned to the SAP System and there processed and stored. The SAP system that is used by the Austrian power supplier has many other interfaces to other systems, that are not part of this master thesis. The reason why these systems are not mentioned in this thesis is because they do not have a direct relationship with the workforce[16] management system which the new solution is made for.

In the sub section above, information about smart meter and big data was mentioned. The solution that handles all the data that is collected from the smart meters that are installed in the households is maintained by other partner companies. This partner is also responsible for security issues, like collecting and handling all keys, that are needed for a secure communication with a smart meter. The other systems are used just as a transporter of an encrypted set of keys. These keys will be decrypted by the direct communication between the communication adapter and the smart meter. All other applications between the main system and the smart meter are used to transport the package.

The main solution is a product that is developed and provided by a big IT company and is used mainly for collecting, handling and processing of smart meter data. This product that the Austrian power supplier uses, is also used by other power suppliers all over the world.[17]

The middle part of the figure 2.2 is reserved for the Workforce Management System (WFMS) that is used to send the right field engineer at the right time to the right place. This WFM system is also used by two types of employees of the power supplier. One employee type is working in-house and are

[16] Maik, 2010.
[17] Siemens AG, 2017.

called dispatchers. The other type of employees is working outside and are called field engineers.

The dispatchers are using a web-based portal solution of the workforce management system. Their job is to coordinate the field engineers and provide them with calls. A typical call could be a smart meter installation, or fixing if there are any troubles with the installed meter at consumer's households. There are numerous other call types that are not directly related to smart meters but also handled in the WFM system by the dispatchers and field engineers. However, these will not be discussed in detail in this thesis.

In contrast to dispatchers who are using a web-based portal solution of the workforce management system, the field engineers are using a mobile solution of the workforce management system called, which is running on Windows, Android and iOS devices. This mobile solution is used by the field engineer to solve the calls that are assigned directly to him or his crew. The field engineer visits the consumer to install a smart meter, or to solve other calls, like reading a smart meter or setting some parameters on the smart meter, etc.

In order to solve a call that is related to the smart meter, the field engineer needs to have some device that allows him to establish and to achieve the communication directly to the smart meter. The field engineers of the Austrian power supplier that is used in this thesis use communication adapters from a partner company. The company provide them Bluetooth devices that communicates with the smart meter via the infra-red technology. The device is paired via Bluetooth with the mobile device of the field engineer and therefor the adapter is called a Bluetooth adapter.

2.4.1 Restrictions

There are numerous restrictions that impede communication with installed smart meters. Some examples include:

- little time for planning the smart meter rollout
- a great number of smart meters is needed that has to be installed first. The producers are late with deliveries.
- time-consuming exchange of old meters and assembly of new smart meters
- communication and infrastructure are still not implemented in order to enable all functions of already installed smart meters

Thus, a new solution had to be found with which it would be able to communicate with the smart meters. This solution was developed by myself and the company I worked for. The solution is called "Meter Communicator" or simple "Communicator".

2.4.2 Use case

A dispatcher that is working with the web-based portal solution of the WFMS assigns a call to a field engineer. This call could be for example to load and install a new firmware on the smart meter in one household in Austria. The field engineer gets a call with all necessary details on his device. After turning on the device and opening the application the field engineer accepts the call. After accepting the call the field engineer drives to the customer. Upon arrival he confirms on his mobile device, which runs the mobile solution of the workforce management system, that he has arrived and that he can start the call. The communication adapter that will be used for communicating with the smart meter is automatically connected and initialized after turning on and opening the call details on the mobile device.

The field engineer puts the communication adapter on the smart meter and presses a button on the mobile device in the running application that would probably be called "Load Firmware". The mobile solution of the WFMS then sends the data via the communication adapter with the new firmware to the smart meter and installs it. The smart meter returns some files to the mobile solution. These files would be parsed and forwarded to the mobile solution. If everything is finished successfully, the call can be closed and sent back to an enterprise IT-System such as SAP.

How this mobile solution communicates with the communication adapter and vice versa will be described in detail in some of the following chapters.

3 Functionality of the application

In this chapter the focus will be on the functionality of the practical part of the thesis, the requirements and aim and objectives of the thesis. At the beginning there will be few sentences about the aim and objective in this thesis. Furthermore, the requirements will be in the focus. The last part of the chapter is reserved for the functionality of the Meter Communicator.

3.1 Aim and Objectives

Like described in the previous chapters, the suppliers ultimate goal is to successfully finish the smart meter rollout, which means that every household switches from electromechanical induction meters, and electronical meters to smart meters. In addition, another aim is to create a fully functioning smart grid to enable communication and an exchange of data between suppliers (in the headquarters) and smart meters in random homes.

AIM: The aim of this thesis is to create a solution to expedite the proceedings around the rollout and which will facilitate communication of data until the finalization of thoroughly operative smart grid. Not every aspect of a smart grid can be used with this solution, but the defined requirements will be fulfilled. Rather, the idea is to use partial advantages a smart meter can bring although the infrastructure might not be ideal.

Objectives: Like described in the previous chapter, this solution called Meter Communicator (or simple Communicator) does not change the existing project environment but is rather used as an additional module, which will permit communication and exchange of data between the field engineers and smart meters, which has not been able before.

The prevailing products used to communicate and transfer information with different types of smart meters, which need to be linked in order to assist the Austrian supplier's field engineer are a mobile solution of a workforce management system and a communication adapter.

This solution is needed to install a new firmware, set parameters or optimize settings for the customer in situations when an energy supplier's employee sets up a new smart meter for a customer. To be capable to do this, they need a way to communicate with the smart meter in the given situation. If the smart meter connects automatically to the mobile application, the risk of making errors while reading the meter can be minimized.

3.2 Requirements

As already written in the chapter's introduction there will be a section with the requirements that can be divided in two subsections. The first section are the general requirements that are defined by the organization called "Österreichs Energie"[1]. This organisation prepares a list of requirements for all suppliers that are directly or indirectly working on a smart meter rollout.[2] This list is mentioned and described in the general requirements section of the thesis. The second part of the requirements they are described in the following sections are the software requirements. These requirements are provided by the Austrian supplier for whom the Communicator is build. There are some requirements that are necessary for every good software. Some of these that will be described are compatibility, scalability, maintainability and some other.

3.2.1 General requirements

For the purpose of realizing the project according to the supplier's needs and wishes, pre-agreed conditions by Austrian energy providers have to be met. Numerous electricity suppliers in Austria united and created an organization called "Österreichs Energie".

Together they defined several use cases, of which only a few will be mentioned here. The solution has to fulfill all requirements. Here just the re-

[1] *Oesterreichs Energie - Oesterreichs Energie* 2018.

[2] *AMCS Lastenheft und Smart Meter-Use Cases - Oesterreichs Energie* 2018.

quirements that are required for the solution will be listed. The other requirements that are related to the solution that is planed after the smart meter rollout are not described here.

All these requirements are related to a situation where the field engineer is working in front of the smart meter at the customer site. The engineer will have the mobile device on which the mobile solution is running. The requirements for the application in this solution are:[3]

- Readout of smart meters - it should be possible to read the power consumption of the smart meter with just one click on the mobile application.
- Deactivation of smart meters - it should be possible to deactivate the smart meter.
- Activation of smart meters - it should be possible to activate the smart meter.
- Obtaining load profiles of smart meters - it should be possible to get all load profiles that are stored on the smart meter or just between two given dates.
- Obtaining log book of smart meters - it should be possible to get the log book of the smart meter.
- Obtaining the installation log - it should be possible to get the installation log of the smart meter.
- Show the actual values - it should be possible to show actual values that are stored on the smart meter. An additional pop-up window

[3] *AMCS Lastenheft und Smart Meter-Use Cases - Oesterreichs Energie* 2018.

should pop up in the mobile application.

- Show the actual time - it should be possible to show the exact time that is used on the smart meter.
- Set the actual time - it should be possible to set the time. This requirement is necessary to assure and enable that the smart meter always works with the exact time and does not have huge differences between the real time and the time on the smart meter.
- Broadcast possibility - it should be possible to add broadcast actions that will be done by the devices in the system.
- Load parameters - it should be possible to load parameters on the smart meter, like the new Firmware, and to install them afterwards.
- Examine communication - it should be possible to test if the connection and communication are established with the smart meter.
- Show the status information - it should be possible to show the status of the smart meter.
- Readout of parameters - it should be possible to read all set parameters on the smart meter.
- Readout of data of the tariff switching table on smart meters - it should be possible to get all results from the tariff switching tables. There will be an additional register on the smart meter that is not read in the standard smart meter readout.
- Readout of load switching table values on smart meters - it should be possible to receive the load switching table values. There will be an additional register on the smart meter that is not read in the standard smart meter readout.

Additionally, there are also a lot of parameters that this solution should handle. The requirement is to handle following parameters and to assure following functionalities:

- Set tariff single parameter - it should be possible to set the tariff single parameter.
- Set tarif double parameter - it should be possible to set the double tariff parameter.
- Set tariff activation parameter - it should be possible to set a parameter that will activate the tariff on the smart meter.
- Set opt-in parameter - it should be possible to set a parameter that will trigger the opt-in process. That means that the smart meter will read and collect data over a defined period. On most smart meters this defined period is 15 minutes. The data will constantly be sent to the energy supplier. They are able to process and analyze the data that is received.
- Set opt-out parameter - it should be possible to set a parameter that will trigger the opt-out process. That means that the smart meter will not read and collect data over a defined period. On most smart meters this defined period is 15 minutes. The data will not be sent to the energy supplier. To obtain data from the smart meter, it is necessary to send a field engineer to the household where the meter is installed and to enjoin them to read out the data ″manually″ with the standard procedure that is defined above.
- Turn display on - it should be possible to send a parameter that will

be processed by the smart meter, and as a result the display of the smart meter, which is installed in the household, will be activated.

- Turn display off - it should be possible to send a parameter that will be processed by the smart meter, and as a result the display of the smart meter, which is installed in the household, will be deactivated.
- Turn the load switching functionality on - it should be possible to turn on the load switching functionality.
- Turn the load switching functionality off - it should be possible to turn off the load switching functionality.
- Set the load switching table parameter - it should be possible to set the load switching table parameter on the smart meter device.
- Set the load switching table activation parameter - it should be possible to set the load switching table activation parameter on the smart meter device.
- Turn the user interface on - it should be possible to turn the user interface on.
- Turn the user interface off - it should be possible to turn the user interface off.
- Set the defined display text - it should be possible to define a free text on the smart meter device. This text should be displayed on the smart meter device.
- Set the customer text on display - it should be possible for the customer to define a free text on the smart meter device. This text should be displayed on the smart meter.
- Reset the meter alerts - it should be possible to reset the defined meter

alerts on the smart meter device.

- Set the installation push parameter - it should be possible to set the installation push parameter.
- Load firmware parameter - it should be possible to transfer/upload the firmware file from the mobile device to the smart meter. Afterwards the uploaded firmware can be installed on the smart meter.
- Set firmware activation parameter - it should be possible to define a date on the smart meter when the firmware, which is already loaded and stored on the smart meter device, should be installed.
- Set SIM login parameter - it should be possible to set the sim login data on the smart meter. This requirement is only necessary for smart meters with a SIM card. The parameters that should be set are username and password.
- Set PLC parameter - it should be possible to set a PLC parameter on the smart meter.

3.2.2 Software requirements

Like every other software, this implementation needs to fulfill some of the basic software requirements, that will assure that the software is ready for productive using. The most important requirements that are defined by the customer will be listed and described in the following subsections.

Compatibility

One of the requirements was that the implementation needed to be compatible with different operating systems. The implementation supports different operating systems like Microsoft Windows and Google Android. An important feature is that the implementation is separated from the WFMS system and therefore can also be used by other system for reading smart meter. To get a useful response, the request needs to be defined with proper parameters.

Maintainability

Another requirement was that the software needed to be easily maintainable. The implementation needs to have exact exception handling and give enough information about warnings and errors that appear while using.

Reliability

One of the most important software requirements in general is that the software needs to be reliable. It was also an important requirement for this implementation. The software achieves this reliability with using reliable technologies for communication with other components in the system.

Scalability

To fulfill the goal to have a good software, the software needed to be also scalable. It is an important software attribute. The whole story about the smart grid and smart meter itself is pretty new, as well as the technology and the requirements. Hence, there will for sure be enough new functionalities that this implementation needs to handle in the future.

Usability

Last but not least, one of the most important requirements for a software is to be easily usable. Therefore the UI for the implementation is built very simple and intuitive. For each supported functionality the user has to press a different button. They are sorted by importance. The most common functionalities are displayed at the top of the page.

3.3 Functionality

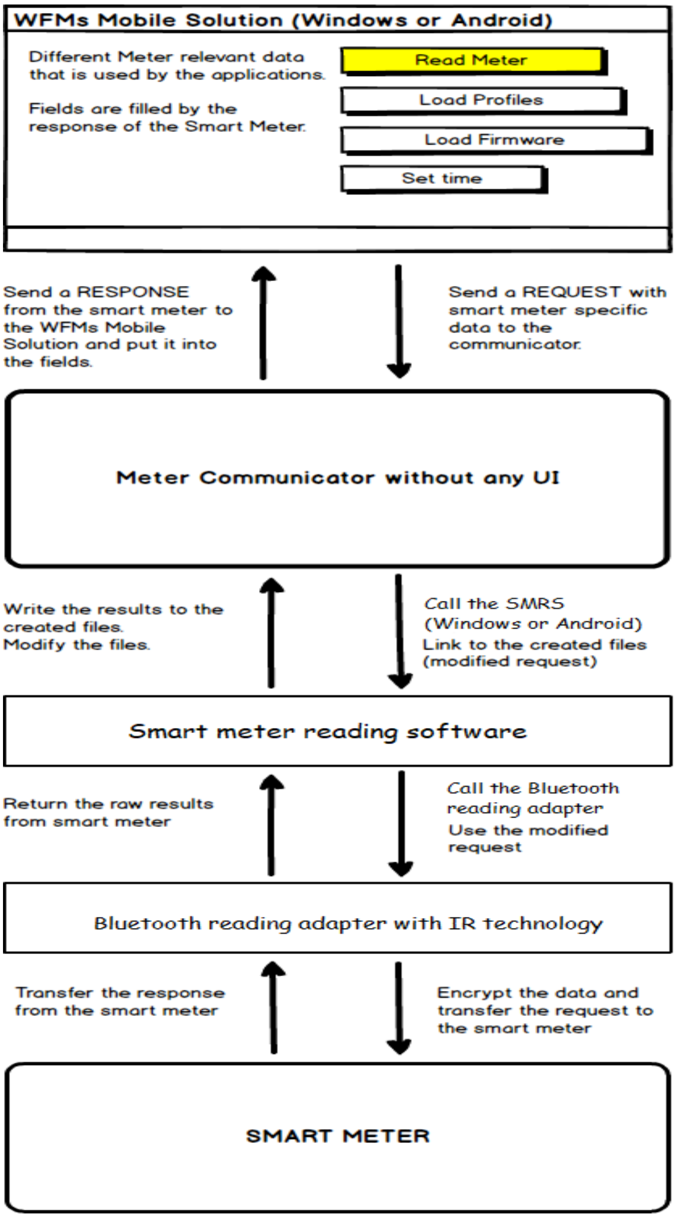

WFMs Mobile Solution (Windows or Android)
Different Meter relevant data that is used by the applications.
Fields are filled by the response of the Smart Meter.
Read Meter
Load Profiles
Load Firmware
Set time
Send a RESPONSE from the smart meter to the WFMs Mobile Solution and put it into the fields.
Send a REQUEST with smart meter specific data to the communicator.
Meter Communicator without any UI
Write the results to the created files.
Modify the files.
Call the SMRS (Windows or Android)
Link to the created files (modified request)
Smart meter reading software
Return the raw results from smart meter
Call the Bluetooth reading adapter
Use the modified request
Bluetooth reading adapter with IR technology
Transfer the response from the smart meter
Encrypt the data and transfer the request to the smart meter
SMART METER

Figure 3.1: Functionality schema - own illustration

The main purpose of this solution is the realization of communication and exchanging of data between a field engineer, the supplier and the smart meter, which will be or is already installed in the consumer's home. In comparison to old methods, i.e. old electric meters, data flows only in one direction, whereas smart meters allow data exchange in both directions.

Like shown in the figure 3.1 above, the field engineer can read the smart meter and can also do other actions, like setting parameters on the device, install a new firmware, or activate and deactivate functionalities or interfaces of the smart meter. To do this successfully, the engineer needs to have an Android or Windows device where all necessary applications are already installed and configured. After successfully configuring and starting the application, it suffices to select the right button in the user interface to trigger which of the above named actions to do on the smart meter. In the figure 3.1 above the functions that are offered are reading the smart meter, reading the load profiles, loading some firmware and setting the time of the smart meter.

After clicking on the right button the data will be collected from the screen and via a request sent to the Communicator. As discussed above, the Communicator is the main subject in this thesis. What exactly happens with the data in the Communicator will be described in the next chapter. Also, it will be described in detail how the data flows between the different applications.

After preparing the collected data that has come from the WFMs application, the next application will be activated from the Communicator. The data will be also sent to the next application in a way that will be described afterwards. The application that is being called by the Communicator is a software from another company which is involved in this process. They offer this product in combination with their hardware for communication and exchanging data with smart meters. The hardware that is used in this process is a Bluetooth device for communication with different smart meters.

As mentioned above the prepared data will be sent from the Communicator to the next application, which will be called SMR application in this thesis. This SMRS is already installed and running on the Windows or Android device that is used by the field engineer.

The mobile application and smart meter need a special file format to communicate with each other. This will be examined in detail in the course of the thesis.

The dataset is collected, prepared and sent from the SMR application to the communication device that is connected to the Android or Windows device where the application is running. While using this device and holding it against the smart meter, data and keys are being exchanged. It is necessary to hold the communication adapter against the smart meter, because the device is using the infrared technology to communicate with smart meters. After a successful transfer, the chosen action will be facilitated on the smart meter.

After executing the called function on the smart meter, there will be a response from the smart meter. After receiving a response, the communication adapter sends it to the SMR application. In this stage, the response will be parsed and formatted to the right way and written into files in a special file format, like already mentioned above.

After writing the response in the files, the response goes to the Communicator where the response will be again parsed and prepared in the format that is useful for the WFMs application.

After receiving the response from the Communicator the WFMs application parses the response again and fills the fields on the user interface with the response data. The collected data from the response will be used in the application for other processes. Some pieces of information are sent from mobile application via interfaces back to the supplier's SAP and then there further processed.

4 Design and technology stack

In this chapter it will be described which pool of technologies is used to create the solution.

As stated above, the main focus in this thesis is on the WFM System and how to get the data from and to a smart meter that is installed or should be installed in a household.

The first versions of the WFMS that are used in this thesis were developed in the nineties of the last century. In the IT world it means that the system is pretty old. There were numerous refactoring and upgrades of the WFMS, but some of the technologies that are used in the meanwhile do not allow to use the newest technology.

4.1 Java Enterprise Edition (JEE)

"The Java EE standard defines a group of Java-based APIs that are suited to creating Web applications for enterprises."[1]

[1] Chopra, Li, and Genender, 2007.

The term APIs can be found in the definition of Java EE. It is an abbreviation for Application Programming Interface and it used to describe parts of some application or service that are available for some other application to use it. The other term that is written in the definition is the “enterprise”, which in this case means large companies that are in need for a software solution. The applications that are based on Java Enterprise Edition are designed for large companies where the solution will be used often.

The Java Standard Edition (JSE) is used as a basis for building the Java Enterprise Edition. The functionalities like Java Virtual Machine (JVM) and the bytecode compiler are parts of the Java Standard Edition. Some of the most common Java EE APIs were in the past and some of them also today: Java Servlet, JavaServer Pages, Enterprise JavaBeans (EJB), Java Message Service (JMS), Java Management Extensions (JMX), Web service APIs (JAX-WS, JAXB, JAX-RCP, SAAJ, StAX), JavaMail, Java Transaction API (JTA). At the time some of the APIs are deprecated and some are new in the Java EE. Some of them were already included in the Java SE.

The two that are important for this thesis will be described in more detail in the following sections and subsections.

4.1.1 JavaServlet

“A Java servlet is a generic server extension - a Java class that can be loaded dynamically to expand the functionality of a server. Servlets are commonly used with web servers, where they can take the place of CGI scripts. A

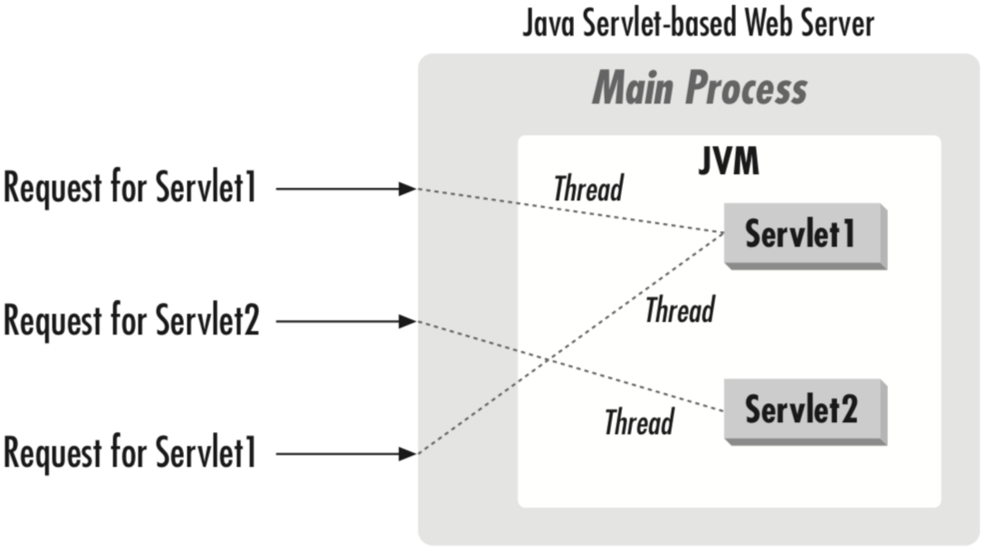


Figure 4.1: The servlet life cycle

Source: Hunter and Crawford, 1998

servlet is similar to a proprietary server extension, except that it runs inside a Java Virtual Machine (JVM) on the server (see figure 4.2), so it is safe and portable. Servlets operate solely within the domain of the server: unlike applets, they do not require support for Java in the web browser."[2]

Whoever uses Java servlets can also use all Java APIs that are offered nowadays. Additionally, there is the possibility to use the JDBC API that allows access to the enterprise databases like it is done in the WFMS. The WFMS uses the Oracle database. The web server and also the application server that are used communicate with the Java servlet over a servlet container. The servlet container is a component that is included in the

[2] Hunter and Crawford, 1998.

WebSphere and used in this environment. The functionality of Java servlets is extended by introducing Java Server Pages (JSP). This technology will be described in the following section.[3][4]

4.1.2 JavaServer Pages (JSP)

"JavaServer Pages is a technology for developing web pages that include dynamic content. Unlike a plain HTML page, which contains static content that always remains the same, a JSP page can change its content based on any number of variable items, including the identity of the user, the user's browser type, information provided by the user, and selections made by the user."[5]

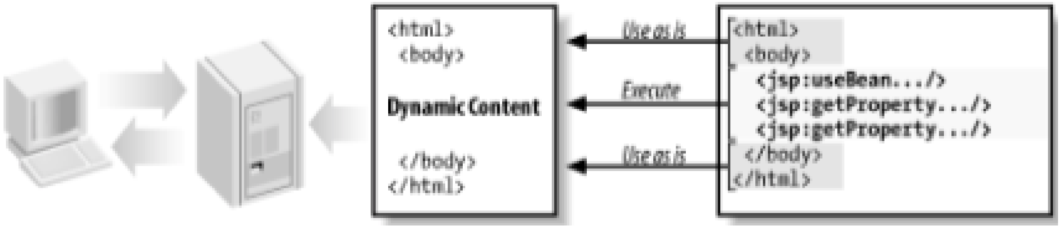


Figure 4.2: Generating dynamic content with JSP elements

Source: Bergsten, 2003

As already mentioned in the definition of JSP above, the JSP can be partly compared with the classic HTML web page because it contains also HTML tags. However, the main difference is that the classic web page is mostly static and has no dynamic elements like JSP. JSP includes elements that can

[3] Hunter and Crawford, 1998.
[4] Chopra, Li, and Genender, 2007.
[5] Bergsten, 2003.

be used and filled out by the server dynamically. It is also possible to get information directly from the used database in JSP.[6]

In figure 4.2 it is shown how the transaction between the web browser and a JSP is working. On the left part the user sends a request from his local device to the server. The user wants to get some information from the JSP. In it is shown in the right part of the figure 4.2, the JSP is built from the static part that is based on HTML with different HTML tags like "html" and "body" tag. These parts are static and will be return as it is to the user, but the middle part of the JSP is dynamic and will be fulfilled on server-side. This is the main difference between the classic web page and the JSP.

It is also possible to define other components in JSP. Some of these components are JavaBeans, which are also part of the Java EE. Another possibility is to define some elements to get information from the database directly in the JSP.

To make JSP easier to use the JSTL is introduced. It is an abbreviation for JSP Standard Tag Library. It enables common custom elements from different libraries which allows easier creation of powerful applications.

As already argued above, the WFMS uses older technologies compared to what is popular nowadays, which does not mean is bad. The system itself is already proven and well tested. As a consequence, it does not make much sense to shift to newer technologies. It would definitely be a good path to

[6] Bergsten, 2003.

follow, because the JavaServer Pages technology is already deprecated. It is harder to find engineers who are willing to work with JSP and on top of that the newer generations of engineers do not have much knowledge about the technology.

At the time when the first version of WFMS was released, it was using the latest technologies based on Java Enterprise Edition 2. In the following sentences it will be explained why the JSP was used.

4.2 Used servers

There were a few passages in this thesis where the structure of the WFMS was described. WFMS contains two solutions. The first one is the one that the in-house employees, called dispatchers, use. This solution is called the portal solution. The second one is the mobile solution, which is used by the field engineer on a Windows or Android operating system.

The portal solution is a web application that is based on JEE 6 Technology like the whole WFMS. The implementation contains JavaBeans, JavaServer Pages and more JEE technologies. For successful running of this kind of solution it is necessary to have a web server prepared. The web server that is used in this case is Apache Tomcat and it will be described in the following subsection.

The mobile solution is an application that runs on two operating system platforms. One of these is the Windows platform that is used by most customers. The second is the Android platform that is not so frequently used at the moment, but it has good chances to become the leader in the battle with the Windows application. The reason is that Android mobile devices are easier to use and less expensive in comparison to the Windows mobile devices. Nevertheless, in both cases Windows and Android need an application server where the whole logic is done. The application server that is used in this case is the IBM WebSphere Application Server and it will be described in some of the next subsections.

4.2.1 Apache HTTP Server

"A Web server is a server that is responsible for accepting HTTP requests from web clients and serving them HTTP responses, usually in the form of web pages containing static (text, images etc) and dynamic (scripts) content."[7]

"The whole business of a web server is to translate a URL either into a filename, and then send that file back over the Internet, or into a program name, and then run that program and send its output back. That is the meat of what it does: all the rest is trimming."[8]

As already written above, the WFMS uses the Apache HTTP Server for the portal solution. Apache HTTP Server was and is also nowadays one of the most used and trustful existing web servers[9]. It is also open source, which means that the financial aspect is likewise relevant for users who need to make a decision on what to use. Apache HTTP Server contains a high number of features that are implemented as compiled modules to provide more functionalities to the core. Just few of them will be named, like:

- mod_rewrite - is used to change/ rewrite the request to the web server

[7] Ljubuncic, 2011.
[8] B. Laurie and P. Laurie, 2002.
[9] *Web Server Survey — Netcraft* 2018.

- mod_proxy - use the web server as a proxy server
- mod _ssl - used for supporting of Secure Sockets Layer (SSL) and Transport Layer Security (TLS). It is used to encrypt the communication between internet browser and the web server.

Apache Tomcat[10][11][12]

Apache Tomcat is an open source web server and a web/ servlet container from the Apache Software Foundation family. Many JEE technologies are implemented in Apache Tomcat. Some of them are: Java Servlets, WebSockets, JavaServer Pages, Java Unified Expression Language and more. Apache Tomcat can easily process different web applications that are developed based on Java Servlets and JavaServer Pages.

Apache Tomcat is based on three parts:

- Catalina - is a Container that executes Java servlets and compiled JSP files. In combination with Coyote it can be used as a stand-alone web server.
- Jasper - JSP Engine is used to compile JSP code and to prepare it for further processing by Catalina.
- Coyote[13] - is a default connector in Tomcat who act as plain web server which basically serve static content to the client. It also handles all

[10] Chopra, Li, and Genender, 2007.

[11] Project, 2019.

[12] Darwin and Brittain, 2007.

[13] *Server Configuration Reference - The Coyote HTTP/1.1 Connector* 2018.

HTTP requests that are usually made on client side. Another feature is that Coyote enables the communication with other web servers. For example, it is possible to process some dynamic data on Tomcat and the static data on another server like Apache HTTP Server.

4.2.2 IBM WebSphere Application Server

In the introduction of the "Used server" section it was mentioned that the mobile solution of WFMS is working on two different operating systems. To successfully run the applications it is necessary to have an application server. The creator of the WFMS decided to use the IBM WebSphere Application Server (WAS), which is the leader of service-oriented architecture (SOA) applications. The IBM WAS was created to handle the enterprise applications. The main part of the IBM WAS is the application runtime environment. This part is used to execute applications. The IBM WAS works fine with other web server like Apache HTTP server that is also described in this chapter.[14]

Some of the main reasons why IBM WebSphere is used are:

- Easy build and deploy of applications
- Secure and scalable environment
- Easy management of the applications
- ...

[14] Sadtler et al., 2013.

4.3 ORACLE Database

In the last decades many IT companies rose from small start-ups to world-wide leaders. One of them is for sure Oracle. Today, Oracle is recognized as the database market leader, who also owns one of the most used databases named MySQL. The WFMS is built on Oracle database, which stands for one of the most powerful database solutions. One of the primary foci of Oracle is to constantly improve all important feature that every database should have. These features are functionality, manageability, security, scalability and many more.[15]

Some of the most powerful features that Oracle offers are listed below.

4.3.1 ORACLE features

Database Programming

This feature enables manipulation of data directly in the database. It is used by developers who work on applications that use Oracle as their database. Oracle supports different technologies like SQL, XML, XQuery and WebDAV.

SQL

"The ANSI standard Structured Query Language (SQL) provides basic

[15] Greenwald, Stackowiak, and Stern, 2013.

functions for data manipulation, transaction control, and record retrieval from the database."[16]

PL/SQL

A feature that is often used in the WFMS is the PL/SQL. This is an extension to SQL that is implemented by Oracle. PL means procedural language and it is used to build stored procedures and triggers and is also good for error handling.

There are several ways to compile PL/SQL. A common way is to use SQL*Plus and also directly in the database.

Java

This feature enables Java to be used as a procedural language in the database.

Large objects

The Oracle database supports storing of large objects since Oracle8.

Object-oriented programming

Another feature that was introduced with Oracle8i was the support for an object-oriented approach. Oracle's Object Views can make use of relational

[16] Greenwald, Stackowiak, and Stern, 2013.

data that is stored in the database.

Database drivers

To allow access to Oracle from different applications the database drivers were introduced with the first version of Oracle. There are two drivers. The first one is based on Java and is called Java Database Connectivity open standard (JDBC). The second is the Open Database Connectivity standard (ODBC).

4.4 Software recommendations

All these technologies that are described in the previous sections and subsections are used in the existing environment where the part that is the core of the thesis is implemented. The WFMS can be divided in two parts. The first one is the portal application that is used by the employees that are working in-house and who coordinate the work of the field engineers. The field engineers who are working outside are using the mobile application that is the second part of the WFMS.

Like already described above, the portal web application runs in a web browser. The web browsers that are supported with the last edition of the WFMS are:

- Windows Internet Explorer 11 and Windows Edge
- Google Chrome

- Mozilla Firefox.

Until the latest version, the WFMS was just able to support the Windows Internet Explorer 11 and the older versions. Now also other web browsers are supported, but the ones mainly used are the Windows products like Windows Internet Explorer 11 and Windows Edge. Google Chrome is also used by quite a few users. Mozilla Firefox is definitely the web browser that is the least supported of all web browsers in the list above.

Most WFMS customers are energy suppliers that are working with privacy information from the end consumers. They have a strong privacy and security guideline and most of them are using Microsoft products. They are not so familiar with open source solutions like Mozilla Firefox. They do not trust solutions from other companies like Google because this company is collecting a lot of privacy information and maybe share them with other partners for analyzing without the agreement of the energy supplier. This is also the reason why most of the energy suppliers in the German-speaking world use Microsoft mobile devices, based on Windows as the operating system.

In the following sections the focus will be on how the main part of the thesis called the Meter Communicator is implemented on a Windows mobile device.

4.5 Prerequisites

As already described, to solve the problem "How to communicate with the installed smart meter by visiting the household" the solution requires numerous applications that are working together. It has already been described that the WFMS is an application from one company, the smart meter reading software (in the following text just SMRS called) is also a product from another company. These two products work strongly together to solve this issue.

The SMRS is installed as a Windows executable file (.exe) on the mobile device that is used by the field engineer. The software has its own folder where it is stored in the system. This folder contains also an executable file (.exe) as well as the batch file of the application (.bat) that is commonly used by the WFMS. If the batch file is not available, the executable file will be used for calling the SMRS. This installation path will be necessary for calling the software to execute some actions. This application needs to be called in the right way by the WFMS system. How WFMS calls the SMRS will be described in the following sections.

The WFMS is also installed on the mobile device and needs to be started before using it. This application has its own folder too, where is it stored in the system. It is necessary to install the WFMS application because it has an own storage for the data that is processed and fetched from the database. The data is collected locally to furthermore make it possible to work in areas where the mobile device that is used might not have an internet connection. The engineer is forced to work offline without a direct connection to the server and to the database. All necessary data is stored in the database and therefore the application makes a full synchronization with the database with every login of the user to get the latest information. After finishing some actions in the application like clicking on buttons send or save the application will store the collected data and change it in the local storage on the mobile device. If the device is connected to the internet, the data will be automatically synchronized with the database and written or deleted in it.

In the following use case it will be described how the applications works on a mobile device that uses a Microsoft Windows operating system.

5 Project use cases

To sum up the previous chapters and to make an illustration of the system, this chapter will explain the system usage in full detail. In addition to the system functionality, the following sections will describe the concrete use cases in more detail.

5.1 Main use cases

There is a big number of use cases that can be described in this section. The plan is to describe a couple of them that will be most frequently used by users.

5.1.1 Read meter data

The field engineer gets an action where he needs to read out the data from the smart meter. He starts the mobile application on his device and goes to the assigned action. He set the communication adapter on the smart meter. By pressing the "Read meter data" button he starts the process. After a few seconds the screen is automatically updated. If the process runs successfully, the meter relevant fields will be filled out with meter reading data.

5.1.2 Read load profiles

The same scenario is described in the "Read meter data" use case will happen. The difference is which button is pressed. The engineer will press the "Read load profiles" button and all defined registers will be shown on the screen. All fields will be filled out with data.

5.1.3 Read logbook

A similar scenario to the "Read meter data" use case will happen. The button that is pressed makes the difference. The engineer will press the "Read logbook" button and a logbook will be attached to the active action.

5.1.4 Read installation log

An identical scenario like described in the "Read meter data" use case will happen. With the press of a different button, namely "Read installation log", an installation log will be attached to the active action.

5.1.5 Show actual values

An analogous scenario like described in the "Read meter data" use case can be seen. The pressed button makes the difference. The engineer will press the “Show actual values” button and a new window will be displayed with the actual values but nothing will be written in the action fields.

5.1.6 Set time

A matching scene like in the “Read meter data” use case will happen. The difference is the pressed button. The engineer will press the “Set time” button and the actual time of the mobile device will be compared to the time on the smart meter. If it is not the same, the new time will be set on the smart meter.

5.1.7 Read time

A resembling case will happen. The difference is the pressed button. The engineer will press the “Read time” button and the actual time of the smart meter will be set in the action field.

5.1.8 Examine communication

A comparable use case will happen. The difference is the pressed button. The engineer will press the “Examine communication” button and the communication will be checked. The response is just a status.

5.1.9 Set parameters

The same scenario like described in the “Read meter data” use case will happen. The button that is pressed makes the difference. The engineer will fill out the field with some parameter value and press the “Set parameters” button. This parameter value will be set on the smart meter. The response is just a status message. This functionality is supported for more than twenty parameters. Some of them are:

- Activate smart meter
- Deactivate smart meter
- Set Opt-In or Opt-Out
- Turn the display on
- Turn the display off
- Set interface off
- Set interface on
- Set display text
- Set PIN Code
- Start Firmware
- ...

5.1.10 Load parameters

The same scenario like described in the “Set parameter” use case will happen. The difference is the pressed button. The engineer will also fill out the field with a parameter value and press the “Load parameters” button. This parameter value will be loaded on the smart meter. The response is just a status message. This functionality is supported for different parameters. Some of them are:

- Load Firmware
- Load Settings
- ...

5.1.11 Read parameters

The engineer will press the button "Read parameters". The set parameters will be read from the smart meter and shown in the mask.

5.2 Detailed use case example

The field engineer has started the WFMS application on his mobile device. They sign in into the WFMS mobile application with their username and password. The engineer gets the initial screen of the mobile application. On top of the screen various tabs are placed. As shown in the figure 5.1, the selected default tab is "MyActions". There are more tabs, but for the thesis the "MyActions" and "Settings" are relevant. The "Settings" (see figure 5.2) tab is used to show where the SMRS specific settings are set in the application. The relevant parameters for this process are "COM-Port" and the "Connection type". These parameters need to be set in the application. The parameter "Connection type" says what kind of connection is used to

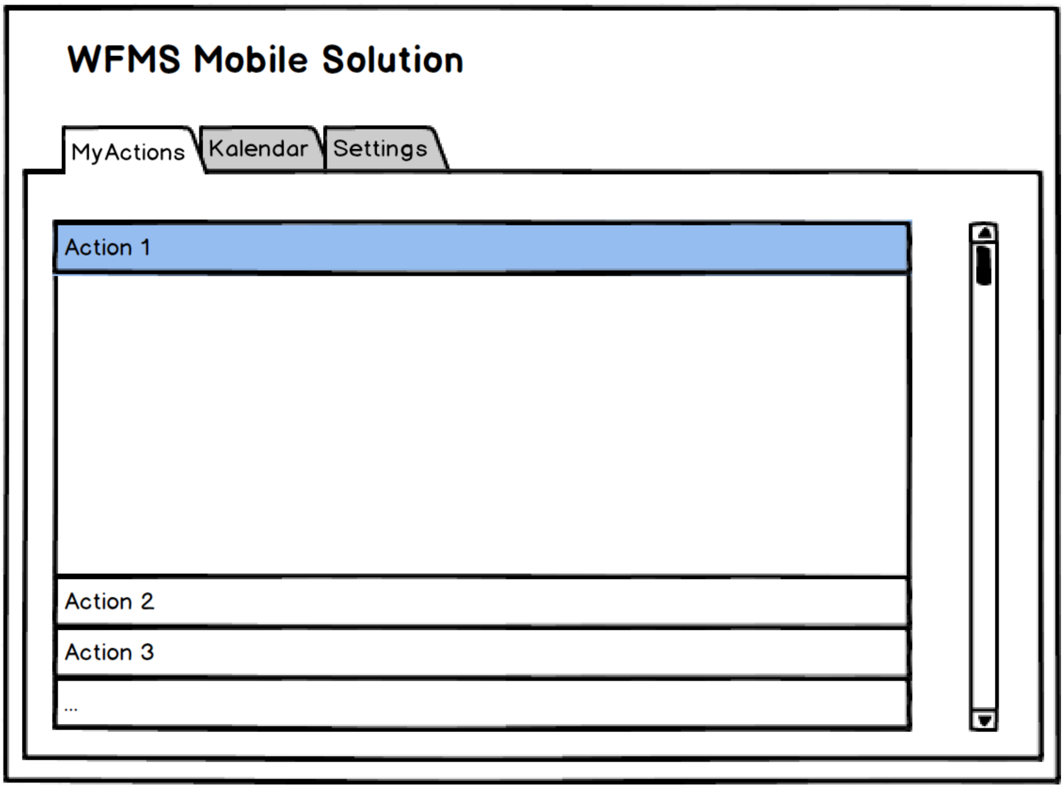
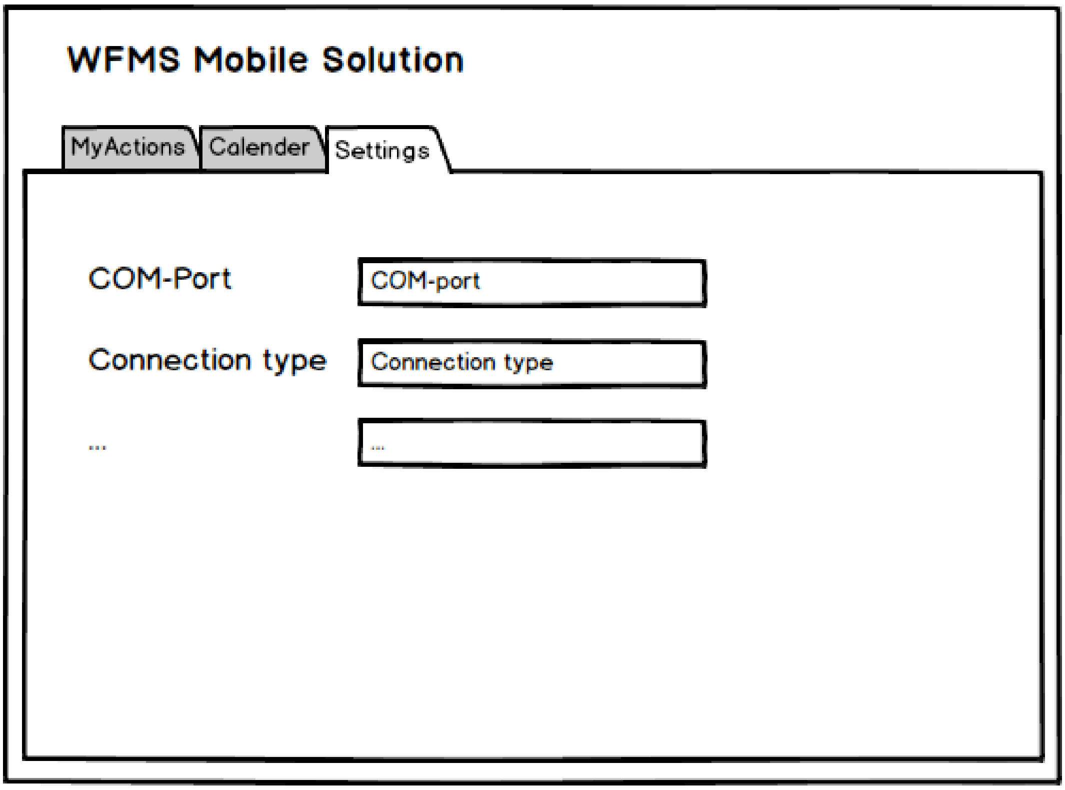


Figure 5.1: WFMS Mobile solution - MyActions screen - own illustration

connect the smart meter reading device. Some of them are supported by the SMRS, but the most commonly used are USB and Bluetooth. In this case the energy supplier uses the Bluetooth device. The "COM-Port" is important to be set because it says on which COM Port on the mobile device the Bluetooth device is connected. Both parameters are important for the SMR software. The WFMS calls the SMR application every time with the action INIT with these parameters. The meaning of this action will be defined in the following sections.

Back to the default tab, which is called "MyActions". In this figure 5.1 all



COM-Port: COM-port

Connection type: Connection type

...: ...

Figure 5.2: WFMS Mobile solution - Settings screen - own illustration

assigned actions are listed. They are directly attributed to the field engineer who uses the mobile solution. The actions are assigned from the dispatcher who is working on the portal solution of WFMS. If the field engineer clicks on any of the actions from the list a new screen will be opened. The new screen is called “Action Details”. Each time this screen is opened and a smart meter related action is started, the WFMS calls the SMR application with the INIT request. With this request a couple of parameters will also be sent to SMRS. These parameters are the two that are defined in the “Settings” tab and paths to the SMR application. The paths to the SMR application folder are important for the communication because the WFMS and the Meter Communicator use these paths to find the executable files for calling SMRS. The SMRS returns just a status if everything was successful or not.

When the “Action Details” screen is open, different requests to SMRS can be started depending on what the field engineer chooses. Which functionality can be used depends also on which type of an action is opened and which smart meter is installed. All functionalities that are defined in the requirements section can be requested.

Depending on that what button is clicked, the right request will be built, processed and called on SMRS. The SMRS processes this request and delivers a response that after processing in the Meter Communicator will be parsed to the right fields and shown in the “Action Details”.

6 Implementation

In this chapter it will be portrayed how the implementation of the practical part of the thesis has been done. There will be one section about the communication between the components and also how the communication is implemented and shown with one simple process. The last part will be used to show some details of the implementation.

6.1 Communication between actuators

In the following subsections will be outlined how the communication looks like on two different layers. The first part is the communication between the physical components in the environment. The second part is used to feature the communication between the software units in the implemented module.

6.1.1 Physical components communication

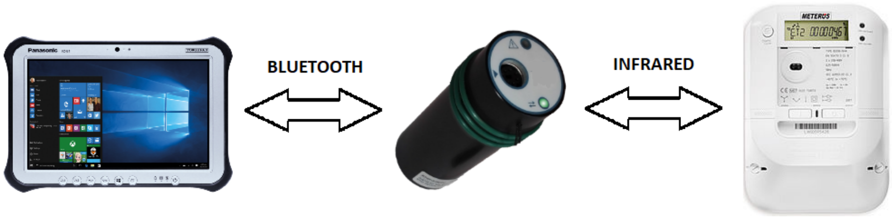


Figure 6.1: Communication of the physical components - own illustration

Like shown in the figure 6.1 the focus of this subsection lies in the representation of the communication of the physical components in the project. There are three main components that are involved in the whole process. The first component is the mobile device that is used by the field engineer. Like already explained a few times, the WFMS and the SMR application run on this device. The mobile device is directly connected to the second physical component, which is the reading adapter. The communication between these two components is realized over the Bluetooth technology. The third component is the smart meter. The reading adapter is connected to the smart meter via the InfraRed technology. To assure the communication with the smart meter, the reading adapter needs to be placed on the right point on the smart meter device. The data flows in both directions from the mobile device to the reading adapter to the end station/smart meter. The way back is from the smart meter to the reading adapter to the mobile device. The mobile device synchronizes the data to the server and there stores the data in an Oracle database.

6.1.2 Software units communication

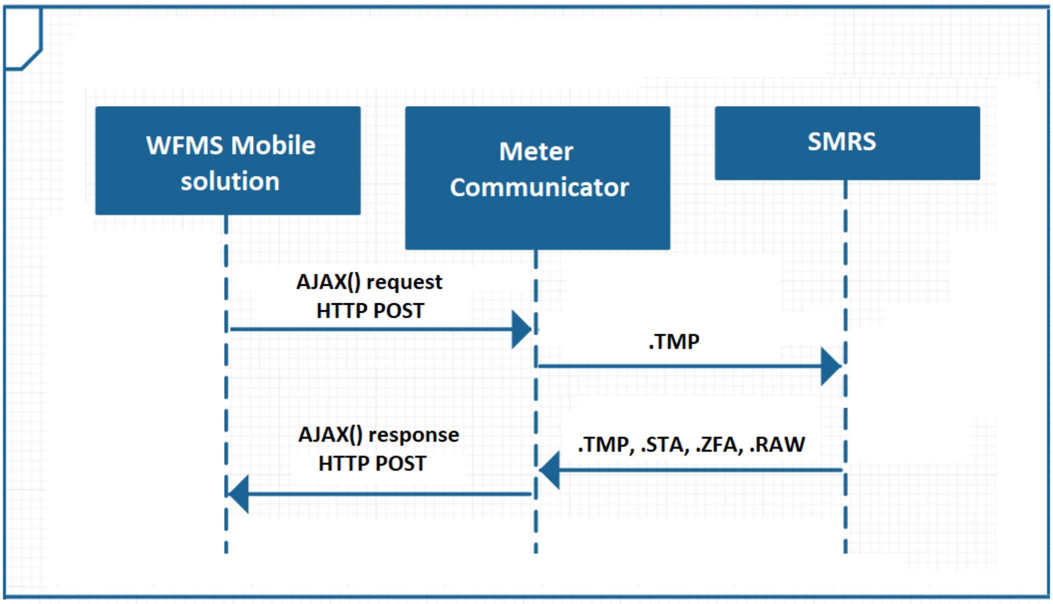


Figure 6.2: Communication of the software units - own illustration

In this subsection it will be exemplified how communication is implemented between the software units in this project.

Like demonstrated in the figure 6.2, the focus is the communication between three main parts. The first one is the WFMS that runs on a mobile device. When the field engineer is triggering some of the actions that are described in the chapter "Use cases" in the section 5.1 in the background the whole communication is started. For calling the next module in the figure 6.2 (Meter Communicator), an AJAX request, which will be explained in detail with a code snippet, in the "Application details" section 6.3 is used. The request calls the JavaServlet of the Meter Communicator with all important parameters. The data will be parsed and used to build a temporary file with a ".tmp" extension. This file is the most important part for the communication with the SMRS. Therefore an example how the file is build and how it look like is shown in the next subsection 6.1.2.

All communication between the Meter Communicator and the SMRS, visible in the figure 6.2, goes over this temporary file, that will be stored in the installation folder of the SMRS. The figure 6.2 also reveals that the Meter Communicator writes the temporary file and the SMRS consumes it. After consuming the content of the temporary file the SMRS processed the information and uses it to communicate with the communication adapter and at the end with the smart meter.

After receiving the raw data from the smart meter, the SMRS parses the data and overwrites the existing temporary file with fresh information. The SMRS also prepares other files that are relevant to the executed action on the smart meter. Also other types of files, like ".raw", ".log" and ".sta" can be generated. All these files are stored in the installation folder of the SMRS. Like displayed in the figure 6.2, the temporary file is used to enable the communication between the Meter Communicator and SMRS in both directions - request and response.

After receiving, the data will be parsed and processed by the Meter Communicator module and written as a response in the JavaServlet. The JavaScript file consumes the response of the AJAX request.

Temporary file (.TMP)

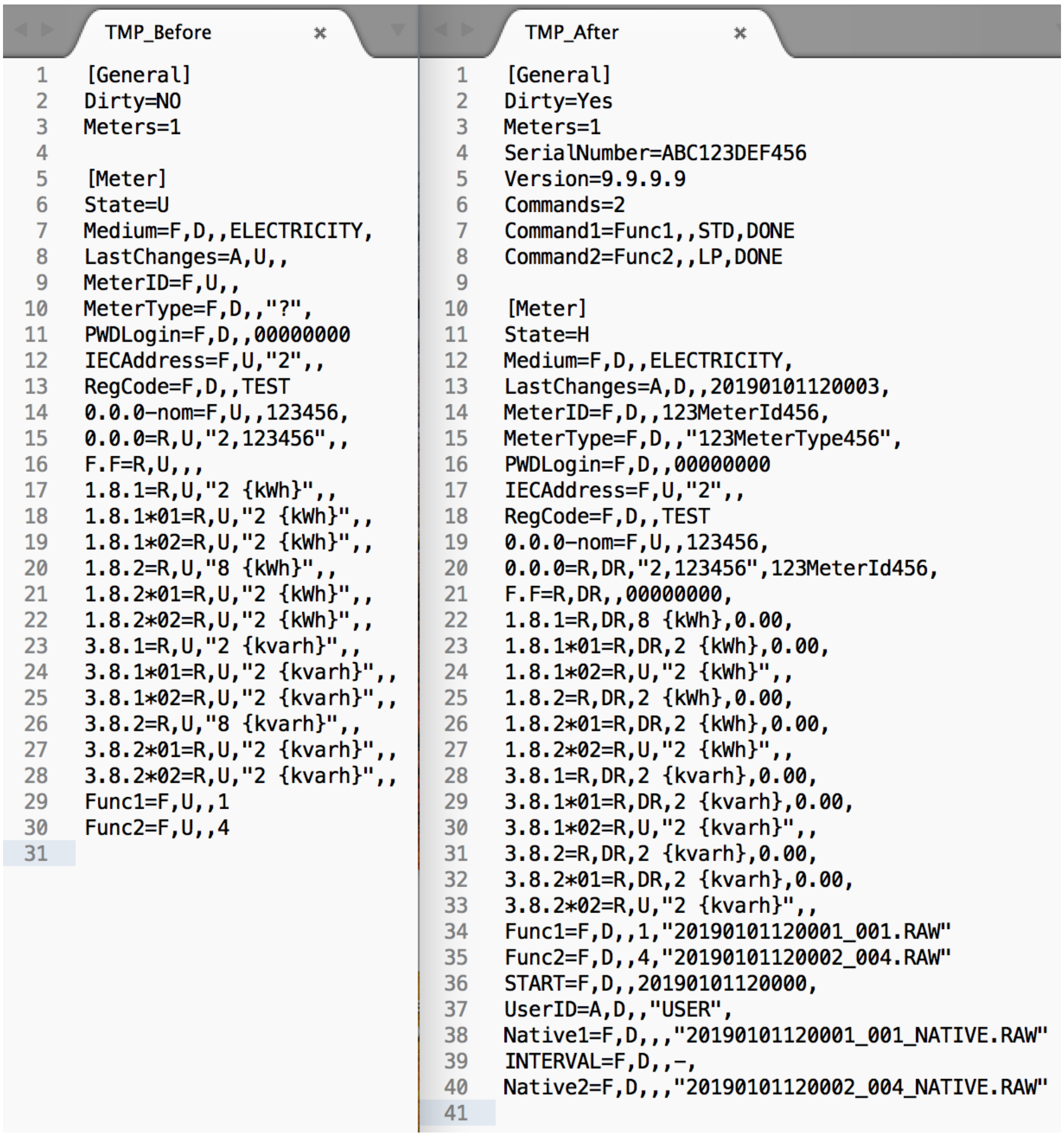
Like written in the subsection before, the main part of the communication between the Meter Communicator and the SMRS is done via the temporary files.

Examples of how the temporary file is built and how it looks like before and after processing are shown in figure 6.3.

The temporary file is built from two parts. The first part is the "General" and the second part is the "Meter".

The "General" part provides general information about the temporary file, the meter and the application itself. The first field "Dirty" explains if the temporary file has been overwritten. If the value is "No", it means that this file has just been created and the SMRS did not write anything in it. After the SMRS has written something in the .TMP file, the current version of the SMR application is shown in the "Version" field. After running the application there are also few interesting fields, like "SerialNumber" of the smart meter. The fields "Command" describe the functions that are processed on the smart meter and their state. In the example there are two functions, one of them is "STD" - MeterReading and the second is "LP" - LoadProfile. Both functions are completed with the state "DONE", which means that they are completed without warnings or errors.

The "Meter" part is built from meter relevant fields. Some of the more important fields will be listed and explained in the following part.



```
[General]
Dirty=NO
Meters=1

[Meter]
State=U
Medium=F,D,,ELECTRICITY,
LastChanges=A,U,,
MeterID=F,U,,
MeterType=F,D,,"?",
PWDLogin=F,D,,00000000
IECAddress=F,U,"2",,
RegCode=F,D,,TEST
0.0.0-nom=F,U,,123456,
0.0.0=R,U,"2,123456",,
F.F=R,U,,,
1.8.1=R,U,"2 {kWh}",,
1.8.1*01=R,U,"2 {kWh}",,
1.8.1*02=R,U,"2 {kWh}",,
1.8.2=R,U,"8 {kWh}",,
1.8.2*01=R,U,"2 {kWh}",,
1.8.2*02=R,U,"2 {kWh}",,
3.8.1=R,U,"2 {kvarh}",,
3.8.1*01=R,U,"2 {kvarh}",,
3.8.1*02=R,U,"2 {kvarh}",,
3.8.2=R,U,"8 {kvarh}",,
3.8.2*01=R,U,"2 {kvarh}",,
3.8.2*02=R,U,"2 {kvarh}",,
Func1=F,U,,1
Func2=F,U,,4

```

```
[General]
Dirty=Yes
Meters=1
SerialNumber=ABC123DEF456
Version=9.9.9.9
Commands=2
Command1=Func1,,STD,DONE
Command2=Func2,,LP,DONE

[Meter]
State=H
Medium=F,D,,ELECTRICITY,
LastChanges=A,D,,20190101120003,
MeterID=F,D,,123MeterId456,
MeterType=F,D,,"123MeterType456",
PWDLogin=F,D,,00000000
IECAddress=F,U,"2",,
RegCode=F,D,,TEST
0.0.0-nom=F,U,,123456,
0.0.0=R,DR,"2,123456",123MeterId456,
F.F=R,DR,,00000000,
1.8.1=R,DR,8 {kWh},0.00,
1.8.1*01=R,DR,2 {kWh},0.00,
1.8.1*02=R,U,"2 {kWh}",,
1.8.2=R,DR,2 {kWh},0.00,
1.8.2*01=R,DR,2 {kWh},0.00,
1.8.2*02=R,U,"2 {kWh}",,
3.8.1=R,DR,2 {kvarh},0.00,
3.8.1*01=R,DR,2 {kvarh},0.00,
3.8.1*02=R,U,"2 {kvarh}",,
3.8.2=R,DR,2 {kvarh},0.00,
3.8.2*01=R,DR,2 {kvarh},0.00,
3.8.2*02=R,U,"2 {kvarh}",,
Func1=F,D,,1,"20190101120001_001.RAW"
Func2=F,D,,4,"20190101120002_004.RAW"
START=F,D,,20190101120000,
UserID=A,D,,"USER",
Native1=F,D,,,"20190101120001_001_NATIVE.RAW"
INTERVAL=F,D,,-,
Native2=F,D,,,"20190101120002_004_NATIVE.RAW"

```

Figure 6.3: Comparation of the temporary files, before and after the processing - own illustration

- State - describes the processing state of the meter, "U" - unprocessed
- PWDLogin - password for entering the smart meter
- All fields from "0.0.0" to "3.8.2" are obis codes that represent the

registers in the smart meter where most of the information is stored. In these registers all energy values that are consumed by the household are stored. These values are important for the energy supplier. They can use the data for analyzing and evaluating and use it for forecasting and planning.

- Func - these fields represent the functions that the field engineer triggers with pressing a specific button on the screen. As already described in the "General" part, in this case the functions were meter reading and reading load profiles.

6.1.3 Action Details - JSP screen

The "Action details" screen is a user interface, based on JavaServer Pages technology. This screen uses different JavaScript files and their functions. The amount of necessary parameters that are relevant for each call are collected in one JavaScript (.js) file. From this file the request is sent with all necessary parameters via an AJAX request to a JavaServlet. What will happen with the request will be explained in the section 6.2.

6.1.4 Meter Communicator

The Meter Communicator is completely implemented as a Java Project and can be used as a JAR file in many applications. In this case the .jar file is included in the library of the WFMS package. Therefore it can be simply called and used from the WFMS like other Java libraries. The input arrives via an HTTP POST request to the Meter Communicator Java Servlet. At the end the servlet writes into the response parameter. This response will be used by the sender, in this case the .js file, for further processing on the user interface. Parameters that are sent to the Meter Communicator depend on which functionality is called by the user interface, because each of the functionalities has their own couple of parameters that are needed for successful execution.

The Meter Communicator project is the main part of this thesis and for this reason it will be explained in detail.

6.2 Description of Reading meter - full process

In the following sequence diagram in figure 6.4 it is shown how the requests are moving and in which order. It is an example of initialization and reading a smart meter.

The represented parts are "Field engineer", "WFMS Mobile solution", "Meter Communicator", "SMRS", "BT device" and the "Smart meter". As already mentioned in this subsection the sequence diagram visualized in figure 6.4 will be described.

The process begins when the field engineer opens an action in the WFMS mobile solution and the "Action details" screen is loaded. On loading of the screen the initialization of SMRS is started. The initialization (INIT) request

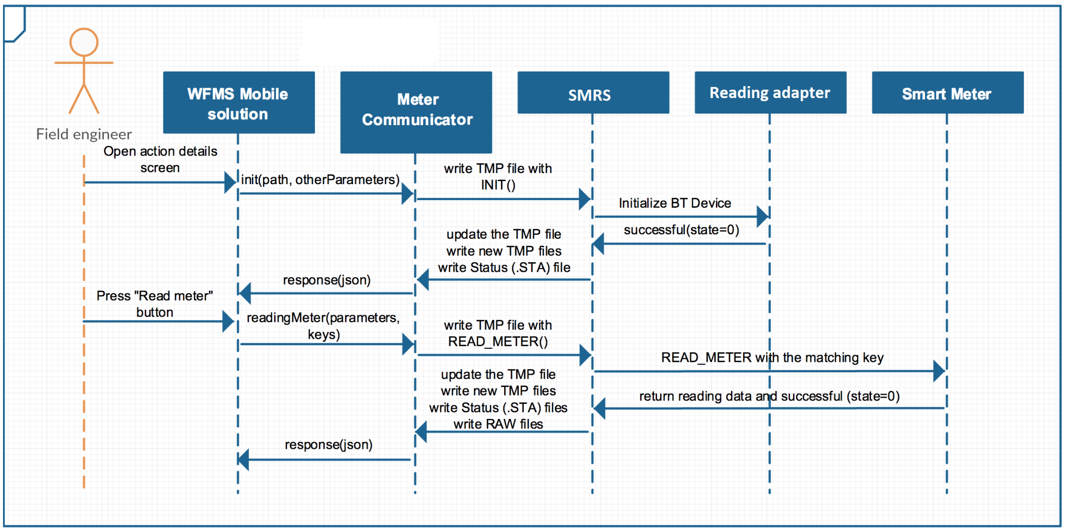

Field engineer
WFMS Mobile solution
Meter Communicator
SMRS
Reading adapter
Smart Meter
Open action details screen
init(path, otherParameters)
write TMP file with INIT()
Initialize BT Device
successful(state=0)
update the TMP file
write new TMP files
write Status (.STA) file
response(json)
Press "Read meter" button
readingMeter(parameters, keys)
write TMP file with READ_METER()
READ_METER with the matching key
return reading data and successful (state=0)
update the TMP file
write new TMP files
write Status (.STA) files
write RAW files
response(json)

Figure 6.4: Reading meter - full process - own illustration

is sent to the Meter Communicator with the parameters that are required for a successful initialization of the SMR application and communication device. Some of these are the parameters from the “Settings” tab and also the paths to the SMRS folder where the SMR application is installed on the mobile device.

After sending this request from the “Action details” JSP, the Meter Communicator receives the request in a JavaServlet where the handling with the received data begins. The data will be parsed and the internal process started. The detailed handling in the Meter Communicator will be written in the following sections. The Meter Communicator writes a temporary file, with the .tmp extension and saves it in the SMRS installation folder. The content in the temporary file has a specific format. The SMR application requires this specific format for handling it in their own business logic. After creating the temporary file and saving it in the SMRS installation folder, the Meter Communicator calls the SMR application on two different ways. How to call the application depends on which operating system the field engineer is currently working. In this case the field engineer decided to work on a Microsoft Windows operating system. Therefore, the Meter Communicator calls a batch file if such a file exists in the SMRS installation folder. If not, the Meter Communicator executes a "smrs.exe" file.

After starting the SMR application, the SMRS looks for temporary files in the folder and if it finds something, it will be processed. In this case, it will just initialize the SMRS and the reading device. If everything is ok, the SMRS modifies the existing temporary file and maybe writes some new .tmp files with the content from the response and writes a status file with the .sta extension.

Meter Communicator parses these new temporary files and writes everything to the response in a JSON format. The response will be sent back to the starting point and it is the "Action details" screen.

If the status of the response is ok, other functions can be called on the UI. In this case the "Read meter" button is clicked. The "readingMeter" action is sent in the request to the Meter Communicator with all necessary parameters and with the keys for a secure communication with the smart meter. The internal process is similar to the initialization process. The steps are the same. The data will be parsed and the internal process started. The temporary file is created and the SMRS is called. The SMRS is now trying to establish a communication path to the smart meter. Therefore the keys are necessary. The WFMS receives the bundle of keys and transfers it to the communication device. This device has an algorithm that decrypts the bundle and makes it usable. If something goes wrong, an exception is thrown and transferred back to WFMS and shown on the "Action Details". If the reading is successful, the SMRS updates the existing temporary file and also writes new temporary files, also a status file and a new type of files, called RAW-files. The RAW-files have the ".raw" extension and are a special format that is written by the smart meter. The temporary files are parsed and processed by the Meter Communicator and sent back to the "Action details" screen where it will be displayed on the user interface.

6.3 Application details

In the first step, the action is triggered on a JavaServer page. The whole logic for collecting necessary parameters is done in a JavaScript file. This file is included in the JavaServer page where the the field engineer is working on. These parameters are used for calling the right functionality in the Meter Communicator. Like described in the chapters before, the Meter Communicator is called with jQuery AJAX request.[1] The request is shown in the code snippet below.

```
$.ajax({
  type : "POST",
  url : servletUrl,
  data : parametersData,
  success : function(data, textStatus, jqXHR) {
    doSuccessHandling(data, jqXHR, pressedButton, serialNo,
   smartMeterId);
  },
  error : function(jqXHR, textStatus, error) {
    doSomeErrorHandling(jqXHR, error, serialNo);
  },
});
```

For calling a jQuery AJAX request it is necessary to define some parameters. In the code snippet are used three of them. The first one is the “type”, which

[1] *jQuery ref AJAX() Methods* 2019.

specifies the type of the request. In our case that is that a HTTP “POST” request. The second parameter is the “url”, which defines where to send the request. In our case that is that the Meter Communicator JavaServlet. The third parameter that is defined in the request is the “data”. This parameter contains the data that will be sent with the HTTP request to the server. In this parameter all parameters that are relevant for calling and processing the right action in the Meter Communicator are defined and at the end called on the smart meter.[2]

After sending the AJAX request from the JavaScript file, the JavaServlet receives the HTTP POST request and continues to process it. In the following code snippet it is visible that the request is not yet fully parsed, just the function/button is parsed and called the right following function. In case the “Read meter data” button is pressed by the field engineer, the switch case will match “readingMeter”. The full http request and the actual http session will be forwarded for further processing. After creating the temporary file and waiting until the SMR application overwrites the .TMP file; the overwritten file will be parsed and returned into the servlet. The result will be written in the “out” map. The content of the “out” map will be written in a JSON object and forwarded to the HTTP response. This response will receive the JavaScript file and will continue to parse it and fill the fields on the user interface/JavaServer page.

[2] *jQuery AJAX() Method* 2019.

```java
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  HttpSession session = getSession(request);
  Map<String, Object> out = new HashMap<String, Object>();
  String button = extractRequestParameter(request, REQUEST_KEY.BUTTON);
  switch (button) {
    case "init":
      out.putAll(init(request, session));
      break;
    case "readingMeter":
      out.putAll(readingMeter(request, session));
      break;
    case "readLoadProfiles":
      out.putAll(readLoadProfiles(request, session));
      break;
    case ...
    default:
      ...
  }
  JSONObject json = new JSONObject(out);
  PrintWriter output = response.getWriter();
  output.print(json);
  output.flush();
}
```

7 Conclusion

As mentioned in the introduction chapter 1 of this master thesis, my goal was to solve the problem that the energy supplier, who we worked with, had. The goal was to develop a solution that will ensure a communication with smart meters in the areas where the consumers live. The challenge I had to deal with was to build a solution in an existing environment that would be used by the energy supplier. This environment is described in the "Existing Environment" chapter 2 of this master's thesis.

With my Meter Communicator I show that all preexisting specifications at this moment, which are defined by the organization of Austrian energy suppliers, called "Österreichs Energie", were achieved. These requirements are listed and described in the "Requirements" section 3.2. The given environment was adapted and extended in this task in such a way that all requirements could be fulfilled.

The Meter Communicator, which is the main part of the thesis, and all other necessary changes in the existing environment, have already been in function a certain period of time. The feedback of the energy supplier is positive and they are using it on a daily basis with only minor issues, which are commonplace with using software in general.

In conclusion, it therefore can be said that the assignment has been a success and my hypothesis has been verified by the results.

8 Outlook and Future Work

The implementation will not be just a temporary solution like it maybe sounds. After completing the smart grid and enabling the option to remotely access the smart meters and communicate with them, the Meter Communicator will be used on daily basis. In case that an installed smart meter has troubles and defects and needs to be “manually” read, the field engineer will use the standard routine to do that like explained in this thesis on many occasions. The standard routine with the field engineer standing in front of the smart meter and communicating with it over the existing technology will also be used by consumers that are using the Opt-Out possibility. For these types of consumers, the smart meters will be offline and can not be accessed remotely. There is a need for these cases to have the possibility to communicate with the smart meter over the method that is used nowadays.

To fulfill the compatibility requirement, the plan is to adapt the Meter Communicator to work also with the Apple iOS operating system. The problem here is that the SMRS is not yet ready to support this operating system, but they are working on it. Once the SMRS supports the iOS operating system, the Meter Communicator can additionally be easily extended to support it. A placeholder has already been implemented for this sort of improvement.

As written in the software requirements 3.2.2, the system is maintainable and scalable. Consequently, the plan is to add more functionalities that will come in the future. The smart meters will get supplementary features and will support more functionalities, and also the system itself needs to be able to handle it. Therefore is the plan to expand the existing functionality with more components.

An additional characteristic of the implementation is that the module is a .jar file and could be used in combination with other software solutions. It does not depend on the system that processes the data before calling the module. It could for sure be another system that calls the module and forward some mandatory parameters. The system just depends on the following SMRS and hence can be used and adapted for all functionalities that the SMR supports.

As written in the conclusion, the system is productively used by the energy supplier and the feedback is more than positive. The system fits the reliability standards that are required for such a software. In combination with the fact that is written in the paragraph above, there is a reason to believe that the software can also be used for other meter types, not just in the energy supplier field. It could be used by other utility suppliers. There are different types of smart gas meters. This type of meters is compatible with the SMRS and can be easily adapted for successful interaction with the Meter Communicator. There are also different types of water meters that can communicate with SMRS and the Meter Communicator.

Bibliography

(EPRI), Electric Power Research Institute (2011). *Estimating the Costs and Benefits of the Smart Grid*. Tech. rep., pp. 11–12. URL: https://www.smartgrid.gov/files/Estimating_Costs_Benefits_Smart_Grid_Preliminary_Estimate_In_201103.pdf (cit. on p. 14).

Alexrk2 (2018). *Karte Höchstspannungsnetz Deutschland*. URL: https://commons.wikimedia.org/wiki/File:Karte_H%C3%B6chstspannungsnetz_Deutschland.png (visited on 06/10/2018) (cit. on p. 9).

AMCS Lastenheft und Smart Meter-Use Cases - Oesterreichs Energie (2018). URL: https://oesterreichsenergie.at/amcs-lastenheft-und-smart-meter-use-cases.html (visited on 07/17/2018) (cit. on pp. 27, 28).

Bergsten, Hans (Dec. 2003). *JavaServer Pages, 3rd Edition*, pp. 1–28. ISBN: 9780596005634 (cit. on pp. 42, 43).

Cederlöf, Magnus (2018). *The European Unions climate policy*. URL: https://www.ym.fi/en-US/The_environment/Climate_and_air/Mitigation_of_climate_change/The_European_Unions_climate_policy (visited on 09/20/2018) (cit. on p. 2).

Chopra, Vivek, Sing Li, and Jeff Genender (Aug. 2007). *Professional Apache Tomcat 6*, pp. 3–27. ISBN: 9780471753612 (cit. on pp. 39, 42, 47).

Commission, US Federal Energy Regulatory (2008). *Assessment of Demand Response and Advanced Metering*. URL: https://www.ferc.gov/legal/staff-reports/12-08-demand-response.pdf (cit. on p. 11).

D.Mohankumar (2011). *How Electronic Energy Meter Works?* URL: https://dmohankumar.files.wordpress.com/2011/05/how-electronic-energy-meter-works.pdf (visited on 06/07/2018) (cit. on p. 15).

Darwin, Ian F. and Jason Brittain (2007). *Tomcat: The Definitive Guide, 2nd Edition*, pp. 37–46. ISBN: 9780596101060 (cit. on p. 47).

Digitalisierung und Wirtschaftsstandort, Bundesministerium für (2017). *Verordnung mit der die Intelligente Messgeräte-Einführungsverordnung (IME-VO) geändert wird (IME-VO Novelle 2017)*. URL: https://www.ris.bka.gv.at/Dokumente/Begut/BEGUT_COO_2026_100_2_1430903/COO_2026_100_2_1430927.pdf (visited on 12/15/2017) (cit. on pp. 3, 4).

Dyer, S.A. (2001). *Wiley Survey of Instrumentation and Measurement*. Wiley. ISBN: 9780471394846. URL: https://books.google.at/books?id=ezvHNAEACAAJ (cit. on p. 1).

Energieverbrauch der Bundesländer (2019). URL: https://www.verivox.de/strom/themen/energieverbrauch-bundeslaender (visited on 03/11/2019) (cit. on p. 12).

ESMIG (2018). *Benefits of smart meters*. URL: http://esmig.eu/page/benefits-smart-meters (visited on 06/07/2018) (cit. on p. 17).

Ferrari, Aurora and Inderbir Singh Dhingra (2009). *India's Investment Climate: Voices of Indian Business*. World Bank Publications. ISBN: 9780821378083. URL: https://books.google.at/books?id=thAHyZx4Y6EC (cit. on p. 11).

Gonen, Turan (2014). *Electric power distribution engineering*. Boca Raton, FL: CRC Press. ISBN: 9781482207002 (cit. on p. 8).

Greenwald, Rick, Robert Stackowiak, and Jonathan Stern (May 2013). *Oracle Essentials*, pp. 10–13. ISBN: 9781449343033 (cit. on pp. 49, 50).

Hunter, Jason and William Crawford (Oct. 1998). *Java Servlet Programming*, p. 6. ISBN: 9781565923911 (cit. on pp. 41, 42).

IEEE Smart Grid Big Data Analytics Machine Learning, Artificial Intelligence in the Smart Grid Working Group (n.d.). *Big Data Analytics in the Smart Grid*. Tech. rep. URL: https://de.scribd.com/document/378847431/Big-Data-Analytics-White-Paper (cit. on p. 11).

jQuery AJAX() Method (2019). URL: https://www.w3schools.com/jquery/ajax_ajax.asp (visited on 02/10/2019) (cit. on p. 75).

jQuery ref AJAX() Methods (2019). URL: https://www.w3schools.com/jquery/jquery_ref_ajax.asp (visited on 02/10/2019) (cit. on p. 74).

Kitzler, Severin (2013). *Smart Meter - Aufgaben, Fähigkeiten und Nutzen für das zukünftige Smart Grid*. URL: https://www.eal.ei.tum.de/fileadmin/tueieal/www/courses/UEEML/lecture/2014-2015-W/HauptseminarAusarbeitung_Smart_Meter.pdf (visited on 11/10/2018) (cit. on p. 16).

Laurie, Ben and Peter Laurie (2002). *Apache: The Definitive Guide, 3rd Edition*, pp. 1–17. ISBN: 9780596002039 (cit. on p. 46).

Ljubuncic, Igor (2011). *Apache Web Server Complete Guide*. Tech. rep., p. 11. URL: https://www.dedoimedo.com/computers/www.dedoimedo.com-apache-web-server-lm.pdf (cit. on p. 46).

Maik, Guenther (June 2010). "Hochflexibles Workforce Management: Herausforderungen und Loesungsverfahren." PhD thesis. Technischen Universität Ilmenau (cit. on p. 20).

Mechanism Of Electromechanical Induction Meter Information Technology Essay (2018). URL: https://www.ukessays.com/essays/information-technology/mechanism-of-electromechanical-induction-meter-information-technology-essay.php (visited on 06/06/2018) (cit. on p. 15).

Oesterreichs Energie - Oesterreichs Energie (2018). URL: https://oesterreichsenergie.at/ (visited on 07/17/2018) (cit. on p. 27).

Pachauri, R. K. (2015). *Climate change 2014 : synthesis report*. Geneva, Switzerland: Intergovernmental Panel on Climate Change. ISBN: 978-92-9169-143-2 (cit. on p. 10).

Pflier, M.P. (Apr. 2013). *Elektrizitätszähler. Tarifgeräte, MeBwandler, Schaltuhren*, pp. 56–192. ISBN: 9783662013144 (cit. on p. 14).

Project, Apache Tomcat (2019). *Apache Tomcat® - Welcome!* URL: https://tomcat.apache.org (visited on 04/02/2019) (cit. on p. 47).

Sadtler, Carla et al. (Aug. 2013). *WebSphere Application Server V8.5 Concepts, Planning, and Design Guide*, pp. 2–32. ISBN: 9780738438467 (cit. on p. 48).

Server Configuration Reference - The Coyote HTTP/1.1 Connector (2018). URL: http://tomcat.apache.org/tomcat-4.1-doc/config/coyote.html (visited on 07/12/2018) (cit. on p. 47).

Siemens AG, Energy Management Division (2017). *EnergyIP MDM Application*. URL: https://assets.new.siemens.com/siemens/assets/public.1500380412.a5875238c1c44acebc92d012833f191a5c97ae80.7629-02-rz-onepager-mdm-170327-3-ansicht.pdf (visited on 11/11/2018) (cit. on p. 20).

Smart Metering mit KNX (2018). URL: https://www2.knx.org/media/docs/Flyers/Smart-Metering-With-KNX/Smart-Metering-With-KNX_de.pdf (visited on 11/10/2018) (cit. on p. 16).

Steinger, Nils (2017). *Measuring, Visualizing, and Optimizing the Energy Consumption of Computer Clusters*. Tech. rep. URL: https://www.sosy-lab.org/research/bsc/2017.Steinger.Measuring,_Visualizing,_and_Optimizing_the_Energy_Consumption_of_Computer_Clusters.pdf (cit. on p. 16).

Thiele, Timothy (2019). *How an Electric Meter Reads Power Usage*. URL: https://www.thespruce.com/how-electric-meters-read-power-1152754 (visited on 01/05/2019) (cit. on p. 1).

Web Server Survey — Netcraft (2018). URL: https://news.netcraft.com/archives/category/web-server-survey/ (visited on 06/10/2018) (cit. on p. 46).

Zachariadis, Ioannis (Oct. 2018). *Investment in infrastructure in the EU*. URL: http://www.iberglobal.com/files/2018-2/infrastructure_eu.pdf (visited on 11/10/2018) (cit. on p. 13).