Informatics

Modulares Energiemanagementsystem

Entwurf eines theoretischen Modells und Verifikation mittels Proof of Concept

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Maximilian Sutrich, BSc
Matrikelnummer 11719759

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Schahram Dustdar
Mitwirkung: Univ.Lektor Dipl.-Ing. Dr.techn. Manfred Siegl

Wien, 1. April 2024

Maximilian Sutrich

Schahram Dustdar

Modular Energy Management System

Design of a Theoretical Model and Verification Using a Proof of Concept

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Maximilian Sutrich, BSc
Registration Number 11719759

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Schahram Dustdar
Assistance: Univ.Lektor Dipl.-Ing. Dr.techn. Manfred Siegl

Vienna, 1st April, 2024

Maximilian Sutrich

Schahram Dustdar

Erklärung zur Verfassung der Arbeit

Maximilian Sutrich, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. April 2024

Maximilian Sutrich

Danksagung

Zuerst und vor allem möchte ich mich bei meinen Eltern bedanken, die mir immer alles ermöglicht haben, was ich machen wollte. Sowohl im schulischen als auch im privaten Bereich konnte ich ausnahmslos allen Interessen, Versuchen und Wegen nachgehen, die mein Interesse geweckt haben. Die Geduld und die Zeit, die gemeinsam für Schularbeiten, Fahrten zu Sportveranstaltungen, Unfälle und deren Genesung aufgebracht wurden, sind unermesslich. Das sehe ich nicht als selbstverständlich an und weiß ich sehr zu schätzen - DANKE!

Ein großes Dankeschön geht auch an den Rest meiner Familie und Freunde, die mich in jeder Lebenslage bisher unterstützt haben und auf die ich mich immer verlassen konnte; die mich durch mein Studium begleitet haben und mit mir Erfolge, Misserfolge und Prokrastinationszeiten geteilt und gefeiert haben.

Danke an meine Kollegen Dominik, Michael und Sven, mit denen ich seit der Schulzeit den Ausbildungsweg gemeinsam gegangen bin. Von unzähligen Nächten im Labor bis hin zu stundenlangen Lern- und Projektsitzungen, um Aufgaben, Herausforderungen und Probleme zu meistern - ohne diesen intensiven Zusammenhalt und die gegenseitige Aufmunterung und Motivation wäre das Studium nicht so gewesen, wie wir es gemeinsam gemeistert haben.

Danke auch an meinen Begleiter und Betreuer Dr. Siegl, der viel Zeit und Wissen in diese Arbeit investiert hat, und an Prof. Dustdar für die Betreuung dieser Arbeit.

Acknowledgements

First and foremost, I would like to thank my parents, who have always made everything I wanted to do possible. Whether academically or privately, I was able to pursue all interests, pursuits, and opportunities that piqued my interest without exception. The patience and time spent studying together for schoolwork, transportation to sporting events, accidents and recoveries are immeasurable. I don't take this for granted and I really appreciate it - THANK YOU!

Many thanks to the rest of my family and friends, who have supported me in every situation of my life so far and on whom I could always count; who have accompanied me throughout my studies and who have shared and celebrated successes, failures and procrastination times with me.

Thanks to my colleagues Dominik, Michael and Sven, with whom I have been on the same educational path since my school days. From countless nights in the lab to hours of study and project sessions to master tasks, challenges and problems - without this intense cohesion and mutual encouragement and motivation, my studies would not have been the way we mastered them together.

I would also like to thank my advisor, Dr. Siegl, who put a lot of time and knowledge into this thesis, and Professor Dustdar for supervising it.

Kurzfassung

Die Innovationskurve und der Einsatz von Technologien in verschiedensten Sektoren und damit auch die Möglichkeiten zur Individualität steigt immer weiter. Diese Individualität basiert jedoch meistens auf elektronischen Geräten oder ausgelagerten Diensten und macht die Nutzung unterschiedlicher und manchmal auch einfacher Anwendungen abhängig von unbeeinflussbaren Faktoren. Darunter fallen auch die allgemeine Stromversorgung sowie die Steuerung und Automatisierung von Gebäuden. Ein immer instabiler werdendes öffentliches Stromnetz, bedingt durch ansteigenden Verbrauch und wechselseitigen Stromaustausch durch private Kleinkraftwerke, ist ebenfalls einer der unbeeinflussbaren Faktoren. Schutz und Unabhängigkeit vor Stromausfällen oder Energieengpässen kann durch diverse Gebäudeinstallationen verhindert werden. Photovoltaik-Anlagen, Windräder, Energiespeicher und Wärmepumpen sind Beispielkomponenten, um (teil-) unabhängig vom öffentlichen Stromnetz zu werden. Kombiniert mit der Steuerung der Verbraucher und möglichen bidirektionalen Komponenten (z.B. Elektroauto) können Gebäude das Stromnetz entlasten und gleichzeitig Absicherung von kurz- und mittelfristigen Abhängigkeiten geschaffen werden.

Die Kombination der erzeugenden und verbrauchenden Systeme stellt jedoch einige Herausforderungen dar, die es zu bewältigen gibt. Verschiedene Systeme, darunter analoge und digitale Komponenten mit keinen oder unterschiedlichen Kommunikationsnetzwerken, müssen miteinander verbunden und gemeinsam gesteuert werden. Damit können verschiedene Ziele wie Energieunabhängigkeit, Kosteneffizienz oder Umweltschutz erreicht werden. Zur Kombination und Steuerung dieser Systeme wird ein Energiemanagementsystem benötigt. Dieses kann zielgerichtet Befehle erteilen, um die einzelnen Komponenten an die aktuelle Situation anzupassen und die gewünschten Ziele zu erreichen.

Unterschiedliche Energiemanagementsysteme werden auf dem Markt angeboten. Diese sind jedoch oftmals auf Hersteller, Kommunikationsnetzwerke oder Anwendungsgebiete limitiert oder abhängig von einer Verbindung mit der Außenwelt. Diese Diplomarbeit setzt sich zum Ziel ein Energiemanagementsystem zu entwerfen, das diesen Einschränkungen entgegenwirkt. Der Autor entwirft mehrere Modelle zur individuellen Erreichung von (Teil-)Unabhängigkeit, bewertet diese anhand verschiedener Parameter, die den gesamten Entwicklungs- und Verwendungszyklus abdecken, und evaluiert das gewählte Modell mittels Proof of Concept, genannt Modulares Energiemanagementsystem.

Abstract

The innovation curve and the use of technologies in a wide range of sectors, and therefore the opportunities for individuality, are constantly increasing. However, this individuality is mostly based on electronic devices or outsourced services and makes the use of different and sometimes simple applications dependent on factors that cannot be influenced. This also includes the general power supply and the control and automation of buildings. An increasingly unstable public power grid, caused by rising consumption and the mutual exchange of electricity by small private power plants, is also one of the factors that cannot be influenced. Protection and independence from power outages or energy shortages can be prevented by various building installations. Photovoltaic systems, wind turbines, energy storage systems and heat pumps are examples of components that can be used to become (partially) independent of the public power grid. Combined with the control of consumers and possible bidirectional components (e.g. electric cars), buildings can relieve the load on the electricity grid and at the same time create protection against short and medium-term dependencies.

However, the combination of generating and consuming systems poses a number of challenges that need to be overcome. Different systems, including analog and digital components with no or different communication networks, must be connected and controlled together. In this way, various goals such as energy independence, cost efficiency or sustainability can be achieved. An energy management system is required to combine and control these systems. This can issue targeted commands to adapt the individual components to the current situation and achieve the desired goals.

Various energy management systems are available on the market. However, these are often limited to manufacturers, communication networks or areas of application or are dependent on a connection to the outside world. This diploma thesis aims to design an energy management system that counteracts these limitations. The author designs several models for the individual achievement of (partial) independence, evaluates these on the basis of various parameters covering the entire development and usage cycle, and evaluates the selected model by means of a proof of concept called a modular energy management system.

Contents

CHAPTER 1

Introduction

The reliability of the electricity supply in Austria is very high. Nevertheless, interruptions are possible. Various media reports indicate that the frequency of supply interruptions will increase and so will the probability of a blackout [ZA]. It can be assumed that the loss of availability of electrical energy will cause significant problems in coping with regular daily activities. To be partially free from dependence on Energy Supply Companys (ESCs), there are various options for generating own electricity. Examples would be Photovoltaic Stations (PVSs), wind power plants and fuel generators, to name a few. On the basis of electrical energy, it is very easy to generate thermal energy with the help of heat pumps, electric heaters, and similar devices. This makes a building self-sufficient in its basic functions.

The infrastructure required to maintain the basic functions mentioned above is very expensive, but in the long run, the investment pays for itself in terms of safety, as well as economics and disasters. The economic factor plays a simple role here. With a higher investment but lower running costs, this type of system is financially self-sustaining after a few years. The logical use of pricing models, energy storages, and control mechanisms of the local infrastructure can actively contribute to a faster amortization.

In the event of a disaster, it is not enough to provide energy in order to use the available resources as sparingly and for as long as possible. The remaining energy must be supplied as quickly as possible to the most essential components of the building. Everything else has a lower priority or should be shut down in a controlled manner.

A building-wide Energy Management System (EMS) is the solution to cover different needs and situations and connect as many systems as possible. This chapter explains the problems and requirements of such a system and provides general information on this thesis.

1.1 Problem Statement

Connection between Home Automation and Energy Systems

Home Automation Systems (HASs) and Energy Systems (ESs), described in more detail in section 2.1 and section 2.2, are usually two or more separate systems. Full integration is often not planned or not possible. Only a few ESs offer interfaces for various HASs protocols and integrations for them to interact with each other more precisely or to carry out more complex command sequences. Even if HASs offer integrations to individual ESs, it is possible that not all systems can be integrated or are not fully compatible within the HASs mask.

Range of Functions of Home Automation Systems

There may be various causes for using a HAS (see section 2.2) in a building. Comfort, security or building maintenance are common reasons. There are different systems with different functionalities. However, dependent actions, more complex state machines and the like are often only possible in more expensive and more comprehensive HASs like KNX [Assa] or similar. These systems are often not compatible with easy to use systems or require a lot of know-how.

Compatibility and Control of Various Manufacturers

Different systems use different communication protocols and logics in the background. There are some major standards such as KNX, ZigBee [Ros] or Thread [BBP23] that enable compatibility between systems. Regardless, functions or datapoints are often lost in the compatibilities and must be controlled in the mask of each system.

Single Point of Controllability

When connecting several systems, whether Home Automation Systems or Energy Systems, the control often remains on the respective platforms. This means that centralized control of all components is rarely possible from a human perspective. Attempts are often made to find a system that acts as the main ecosystem for the entire infrastructure and takes over the entire management. Nonetheless, as mentioned above, the full range of functions of the system may be lost in the process or more complex processing and actions of different systems are not possible.

Neglect of Home Automation Systems

Various manufacturers as well as institutes, funding programs and ESCs are developing and have developed Energy Management Systems (EMSs). However, the aims of individual groups are usually very limited. Manufacturers try to increase the range of functions and thus the Unique Selling Proposition (USP). Funding programs and ESCs usually focus on larger areas such as the electricity network and the general energy supply. As a result, the building itself is often seen as the smallest datapoint. When the development is more focused on buildings, the focus is usually solely on large consumers, such as e-car charging stations or batteries. As self-generated energy feed did not exist in grid planning at the time, this is also a growing issue. Unfortunately, this issue is mostly considered in relation to grid stability only. In these projects, the economic and comfort benefits for individual customers are usually not considered. For more details, see section 2.1.

Extensibility

The system operator or manufacturer decides which systems are and remain compatible with which. With private HASs, there are rarely guarantees as to how long a service will be operated and updated. Regardless of this, detailed planning is required to determine which systems are compatible with which functions and with what range of functions. In addition to the existing compatibilities, it is usually not possible to upgrade or include non-integrated systems afterwards. Manufacturers generally do not offer development interfaces to connect various systems. Planning is therefore limited and during long-term operations, one is dependent on the system operator to ensure that compatibilities continue to be maintained.

Various Restrictions due to Manufacturer Specifications

The manufacturer often imposes restrictions not only on compatibility, but also on the system itself. This may be in regards to the range of functions and options within the individual datapoints or a feature of the system. For example, Fronius installs a 30-second shutdown as a safety mechanism in the inverter if there is no more power coming from the ESCs. This means that the building is completely without power for a short time, even with existing energy sources.

Security

Home Automation Systems or Energy Systems can be used to collect a lot of information about current and long-term behavior in the building or even control datapoints. The security of these systems is of primary importance. Manufacturers attach great importance to security, but more and more systems have connections to the internet or to a manufacturer's cloud. This has both convenience advantages for the user and often also manufacturer reasons for collecting data and information. Often, only a limited range of functions can be used if the system is not connected to the internet. Convenient controls via apps and the like are frequently based on the cloud. The same applies to the compatibility of different systems. A local solution with several conventional HASs or ESs is very complex and often not possible.

1.2 Aim of Work

Based on the problem statement, the aim of this thesis is to develop a comprehensive solution proposal. This is examined through a practical evaluation in the form of a Proof of Concept (PoC) and subsequently proven or rejected. To begin with the planning and the designing of a theoretical model development and assesment, the functional scope of the Modular Energy Management System (MEMS) must be defined. The following list describes the most important basic functions and requirements for the final PoC:

1. Combine different Home Automation Systems and Energy Systems.
2. Enable dependencies and more complex queries between different datapoints of different systems. Possibility of configuration of different situations and reactions of the system.
    - Executing commands in different systems.
    - Possibility to define complex conditions (e.g. using Boolean operators).
    - Reactions to changes in the connected HASs or ESs (further called systems).
3. Subsequent expansion of the MEMS to different protocols and systems.
4. Simple operation for end users and limitless operation for professionals.
    - An amateur or end-user can use it.
    - Interested end-user can manage it.
    - Technicians can install it.
    - Professionals can improve and extend it.
5. Subsequently expandable range of functions and controls by professionals.
    - Various value comparison operations can be added.
    - New data types can be added.
    - New functions can be added.
    - Functions such as extended logging or Artificial Intelligence (AI) suggestions can be added.
6. Security in form of a local operating mode.
    - The MEMS does not require an external network connection for its own operation.
    - Connections to the systems can be secure direct connections.
    - Management can take place via a physical direct connection.
    - Alternatively, a connection to the local network can be established.

  - Additional routing options can be provided by external services such as Virtual Private Network (VPN) and co.

7. Controllability using various visual and mechanical methods.
   - An Human-Machine Interface (HMI) can be connected to control the MEMS.
   - The MEMS is configurable using a HMI.
   - Control can be implemented afterwards using various methods.
   - The individual systems are still controllable via their own mask.
8. Combination of useful/economical and comfortable functions of different systems.

1.3 Motivation

Security, disasters and sustainability are issues that are currently of great concern to society and to the author. Security does not necessarily have to be related to crime. The term security is also used on topics such as security of supply and survival. Security of supply in terms of electricity, thermal energy and, to a certain extent, mobility is already possible for end user with today's technologies. For a long time, some households have had PVS or similar for various reasons. Due to rapid development and progress in innovation, systems from five years ago are sometimes outdated in terms of their possibilities and compatibility. Many end users are enthusiastic and interested in technology, but are now unable to make full use of their newest upgrades and enhancements.

Alongside university studies, the author's occupation is electric mobility consultant for Audi. In that role, the author deals with electromobility itself, but also with the charging infrastructure and the electrical grid behind it. Electric vehicles are increasingly being charged at home and are therefore huge power consumers in private households. Consumers of that amount of power were not taken into account in electricity grid planning until a few years ago. The grid is becoming increasingly unstable [GW] and is under a rising load due to new consumers. It is almost impossible to constantly upgrade the grid sufficiently. Therefore, alternative models can be assumed to be necessary to relieve the load on the electricity grid. Flexible pricing models, variable maximum provided loads and load management systems are potential solutions. These are solutions that are often being developed by the ESCs, but are only planned up to the building connection. In order to not lose flexibility and to make the best of the situation in monetary terms, an EMS should exist based on the parameters of the ESCs that optimally adapts the building components to the situation.

Building management, HASs and ESs have always been a constant in the author's studies, in addition to private interests and an owned HAS network. Due to the high level of interest and the modular structure of the study topic, the author attended as many courses as possible that dealt with these topics. The knowledge gained there has been very intensively reflected in this thesis. Regardless of this, projects and experiments related to the environment or the security of supply will continue to be very important in the coming years [Mei23].

1.4 Type of Thesis

The process of a thesis begins with the selection of a field of interest, the corresponding institute and supervisors and finally with a specific topic. The author's field of interest was clear from the beginning: Distributed Systems, Home Automation, Networks, Internet Computing. A field of interest that was also reflected in the modular eligible courses. An appropriate institute for these areas is the Distributed Systems Group (DSG). The DSG offers many courses in these topics. Advanced Internet Computing, Computer Networks, Network Engineering, Pervasive and Mobile Computing are amongst them. A common person in many of these courses is Dr. Manfred Siegl.

Dr. Siegl initially distinguished between two types of Master's theses: Theses that deal with a research background and relevant literature research or theses that verify and apply the content and methods learned during the courses of study to acquire knowledge. This work is written opting for the latter, the applied thesis. Together with Dr. Siegl, a number of subject areas were examined and specific topics identified whose results would be relevant in DSGs field and in the future. One of these topics was the plausibility check of how and whether different HASs or ESs can interact with each other without losing functions and comfort. To this end, various models were to be designed, a PoC developed and verified by use cases. This proposal was submitted by Dr. Siegl to Professor Schahram Dustdar and was accepted.

Amidst regular exchange, research was started on model possibilities and models were designed. These were discussed together with Dr. Siegl, defended by the author and subsequently improved. The model selection and all subsequent steps were also discussed with Dr. Siegl at regular intervals until the thesis could be evaluated and a final result achieved.

1.5 Structure of Thesis

Due to the applied type of this thesis, it deals intensively with the design of models, their assessment and subsequently with the development and evaluation of the selected model. Before a solution is designed, the current state of the art and relevant background is summarized in chapter 2. Chapter 3 describes and compares the models developed. Additionally, the comparison parameters are described and the final model to be implemented and verified is selected. The development phase and the description of the selected programming languages, frameworks, algorithms and data structures and the finished PoC are described in chapter 4. The verification of the solution is outlined in chapter 5 and concluded in chapter 6. Possible further development stages and improvements are listed in section 6.2.

1.6 Relevance to the Curricula

The author’s academic career began in secondary school at a technical college (HTL) specializing in network technology. This was followed by a Bachelor’s degree in Computer Science at the Vienna University of Technologies. Remaining in the subject area of computer science, this thesis is aimed at a Master’s degree in "Software Engineering and Internet Computing". The entire degree program was designed with network technology, distributed systems and smart homes in mind. These subject areas were explored in greater depth at modularly selectable stages of the course. The following list names specific courses attended that are related to this thesis:

- **Advanced Internet Computing.**
  This course covers the areas of service-oriented computing, web services, cloud computing and Internet of Things (IoT) cloud systems and imparts knowledge about theoretical foundations, technologies, architectures and standards. The knowledge gained here was used in the development of models for combining different systems. Furthermore, the know-how of cloud and edge computing was used for the different possibilities of decentralized or centralized logic.
- **Advanced Software Engineering.**
  This project-based course comprised a comprehensive software project, from problem description to implementation. Planning and work steps were learned. Supplemented by technological know-how, such as the use of Java and several frameworks, the content of this course was part of this thesis from start to finish.
- **Computer Networks & Network Engineering.**
  The courses "Computer Networks" and "Network Engineering" complement each other and provide a comprehensive overview of network technology and distributed systems. Different connections, including the ip stack, are discussed in detail. In addition, a general understanding of communication networks is taught, which serves as a basis for many, but also for the MEMS network.
- **Distributed Systems Technologies.**
  Client-server, object-relational mappings, presentation layer technologies (web frameworks), integration technologies (EAI, web services), container technologies (e.g. Docker), aspect-oriented middleware and message-oriented middleware are just some of the skills taught. These were applied at different times and in different parts of the thesis.
- **Pervasive and Mobile Computing.**
  While many technologies are now based on the ip stack, other protocols are also still in use. Many industry standards as well as HASs and ESs use such protocols to exchange messages. KNX is also taught for the first time in the course. This course paves the way for this entire thesis. The understanding of the interaction between HASs and ESs as well as the different protocols are explained.

- **System and Application Security.**
  System and application security have become an integral part of any development project. This course deals with the most common vulnerabilities and how they can be avoided. The security of the HMI component as well as the general application security have been done with the help of this course.

CHAPTER 2

Background

Home Automation Systems and Energy Systems often go hand in hand. Home Automation Systems allow various devices or components in a building to be controlled for various purposes. Energy Systems enable the generation, distribution and storage of energy in buildings. Energy Systems are also controlled systems, but this function is not the primary focus for its main purpose. In both areas there are different functionalities, manufacturers, qualities, price ranges, protocols and compatibilities.

These systems are often installed and expanded gradually. This means that precise and future-proof planning should be carried out. However, this is not always possible due to new technologies, cost savings, or the like. As a result, existing systems can only be expanded and used under certain conditions.

This chapter explains the current general tasks and differences of the systems and summarizes their possibilities. In addition, researches, problems and limitations of current approaches and systems will be shown.

2.1 Home Automation Systems in General

Home Automation Systems, or Smart Homes, refers to a residential concept based on the use of HASs to enhance the comfort, efficiency, and security of its inhabitants. In HASs, various components and devices of a building can be externally controllable or monitorable [DPG06]. These components are often devices that have nothing to do with technology or electronics in a normal building. These include electrically controlled windows, heaters or water pipe valves, for example. The controllability and possible uses of conventional electrical components such as lights, cameras, ventilation systems and the like are thus extended. HASs connect these components with each other, visualize and control them for the various reasons mentioned above.

Components and Functions

HASs are based on various technological components and standards [JK17, Alh04]:

- **Sensors and Actuators.**
  Sensors collect data and statuses from components. A component often only has sensory tasks. Actuators execute activities to achieve the specified actuator state. This can then be verified again via the sensor. In most cases, each component that is an actuator also has a corresponding sensor value. However, an actuator can also be verified by another external sensor component.

- **Communication Technology.**
  The sensors and actuators must communicate with the controller. This can be done via wired or wireless connection. Several standards and protocols such as Local Area Network (LAN), Wi-Fi [Wik24d], ZigBee, Bluetooth Low Energy (BLE) [BS], Thread or KNX [Assa] have become established [SJ19].

- **Control Unit.**
  Mobile phones, tablets or voice control systems are often used to monitor and control statuses and to configure the system. However, depending on the type of component, there may also be dedicated control components, such as smart light switches.

- **Cloud Service and Artificial Intelligence.**
  Most HASs use cloud services to enable advanced functions, data analysis and remote access [DRD16]. These systems frequently do not work without a connection to the internet. Due to cloud services, data gets analyzed and based on known models or AI, suggestions will be made [GSZW19]. This is the first time that HASs for private and business use are separated. KNX, for example, operates as an isolated system and can therefore form a stand-alone system [TC19].

An exemplary and common topology is shown in figure 2.1.

Areas of Application [SFDR20]

HAS technologies are applied in various areas:

- **Energy Efficiency.**
  Through intelligent control of heating thermostats and lights, the system adapts to the needs and living situation of the user. The lights are turned off when not in use and the heating only heats up as soon as the first person is on their way to the building [PGFGB$^+$21].

- **Security.**
  Alarm systems, motion detectors and cameras are just some of the components that

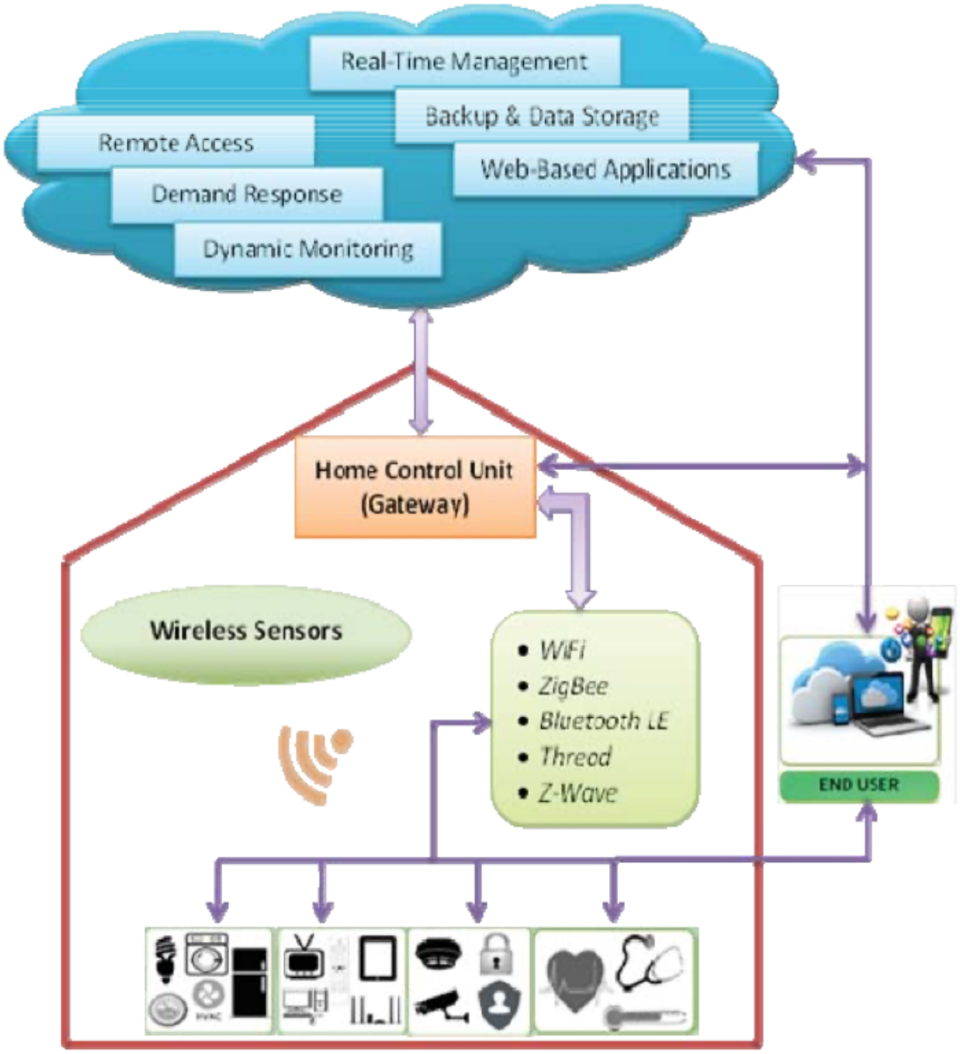


Figure 2.1: Common topology of a Home Automation System network [PIP+22]

contribute to a secure home. But security can also be upgraded from a different angle. For example, a broken water pipe can be detected very quickly if the system knows that no one is currently in the building and that no appliance is currently using water [NS19, MSO17].

- **Comfort.**
  Many users use HASs to conveniently control practical components such as lightning, radios and the like.

- **Health and Feel-Good Factor.**

Special sensors can be used to improve quality of life. Oxygen content, humidity, or temperature can be regularly monitored and adjusted.

Challenges and Concerns

Despite the many benefits, there are also challenges and concerns regarding HASs [AZ19]:

- **Security and Privacy.**
  By expanding or adding components, daily routines and personal habits amongst others can be monitored more easily. Combined with the fact that many systems store data in the cloud, there are always major security concerns with these solutions.
- **Interoperability.**
  Different components support different protocols and standards, which means that it is often not possible to add special components or a subsequent changeover is very complex and costly. If a protocol or standard is no longer supported, complete changeover and replacement may be necessary.
- **Costs and Environment.**
  Smart components are more expensive than components that only fulfill their primary purpose and often have to be retrofitted and replaced. This results in additional cost and waste of functional components. Operation may also involve costs.

2.2 Energy Systems in General

Energy Systems are complex combinations of technical, economic and social components that serve to generate, distribute and use energy in various forms. Their operation, development, and research involve many different disciplines such as electrical engineering, computer science, mechanical engineering, economics and many more. ES can be viewed and used at different levels; from international energy networks [LM17] to ESC networks to a building ES [YBMK18]. The latter is called an ES in this thesis. An ES refers to the infrastructure and technology used to supply buildings with energy for heating, cooling, lighting, electrical appliances and other energy needs. This can create an energy-efficient, renewable, economical or fail-safe energy supply for a building. In the following, the word energy refers to electrical energy.

Components and Functions [YBMK18]

- **Energy Generation.**
  In conventional buildings, the energy comes from the house supply, i.e. the ESC. This source is usually retained in an ES. Furthermore, alternative energy sources such as PVSs, wind turbines or other types of small power plants are increasingly being integrated into buildings. In addition to electrical energy, thermal energy can

also be generated as a form of energy by means of a gas boiler, heat pump, electric heating or similar.

- **Energy Storage.**
  For self-sufficiency, but also for economic reasons, alternative energy sources also create a demand for energy storage. The energy fed into the ESC grid generates much less money than the energy drawn from the grid. However, alternative energy sources usually produce non-periodic and often surplus energy, which is ideally used immediately. Therefore, battery storage systems or thermal energy storage systems are used to store surplus energy.

- **Energy Distribution.**
  Energy is distributed through the electricity network of the building. Thermal energy is distributed through gas or liquid-filled (usually water-filled) pipes.

Challenges and Concerns [EA21]

- **Costs.**
  The investment and installation costs for alternative energy sources, storage systems and the like are often very high and only pay for themselves after a few decades. Furthermore, all maintenance obligations and necessities are no longer completely outsourced to the ESCs.

- **Integration of Renewable Energies.**
  Integration of renewable energy sources requires suitable storage solutions and management strategies to compensate for fluctuations in energy generation and ensure reliable energy supply [DSdHR16].

- **Grid Integration.**
  Feeding surplus energy into the grid is a major challenge for ESCs [EM15] and is often very poorly compensated or even limited. Separate contracts and tariffs must be concluded. The ESCs, which were previously able to base their production on current consumption and many years of forecasting experience, now have to manage production more flexibly. Thousands of small power plants are being integrated into the electricity grid as a result of households feeding into the grid. In addition, these small power plants supply electricity very inconsistently and at times of day when little electricity is needed.

- **Regulatory and Technical Standards.**
  Developing and adhering to standards for the installation, maintenance and safety of Energy Systems in private buildings is crucial to their efficiency and sustainability.

- **Interoperability.**
  An ES usually consists of large and expensive components and is often installed with a finished set of functions or is only rarely upgraded. Many manufacturers

offer a wide range of device functions. This is often very expensive or other sector manufacturers sometimes have better products. This makes interoperability necessary in order to combine different energy sources and energy storage systems [DSdHR16]. However, this is often not implemented or can only be realized at high additional cost. This means that optimum energy use cannot be achieved.

2.3 Energy Management Systems in General

Energy Management Systems in buildings are integrated software and hardware solutions that are designed to monitor, control and optimize energy consumption within a building or building complex [BZ15]. In addition, these systems can integrate other aspects such as comfort, automation et cetera. These systems play a crucial role in minimizing energy and therefore costs [LC16], improving sustainability [LCLP15] or making a building self-sufficient [ERDW16] in the event of a disaster. A comprehensive understanding of these systems requires consideration of various aspects, including their components, functionality and benefits.

Components

Similar to an HAS, an EMS consists of sensors, actuators, a communication network and a management [LGRC20, LCLP15]. The devices and components of an ES are also regarded as simple sensors and actuators. Furthermore, EMSs offer extended functions for data management and analysis. An examplary and common topology is shown in figure 2.2.

Functions

In order to exploit all the functions and possibilities of an EMS, it needs as much data as possible. This data is collected by sensors and actuators from a wide variety of sources, systems and device types [ZLC$^+$16]. One of the most important and crucial tasks here is to collect data from different systems and then normalize the data into a general and comparable dataset. Once all data is comparable, an EMS performs the following main tasks:

- **Data Acquisition.**
  At regular intervals or event-based, the system collects data regarding energy consumption, surrounding conditions, user behaviors, user definitions and much more.
- **Data Analysis.**
  The data is analyzed and evaluated. Direct or time-delayed tasks are then executed or models are generated based on user behavior using machine learning, suggesting potential savings.

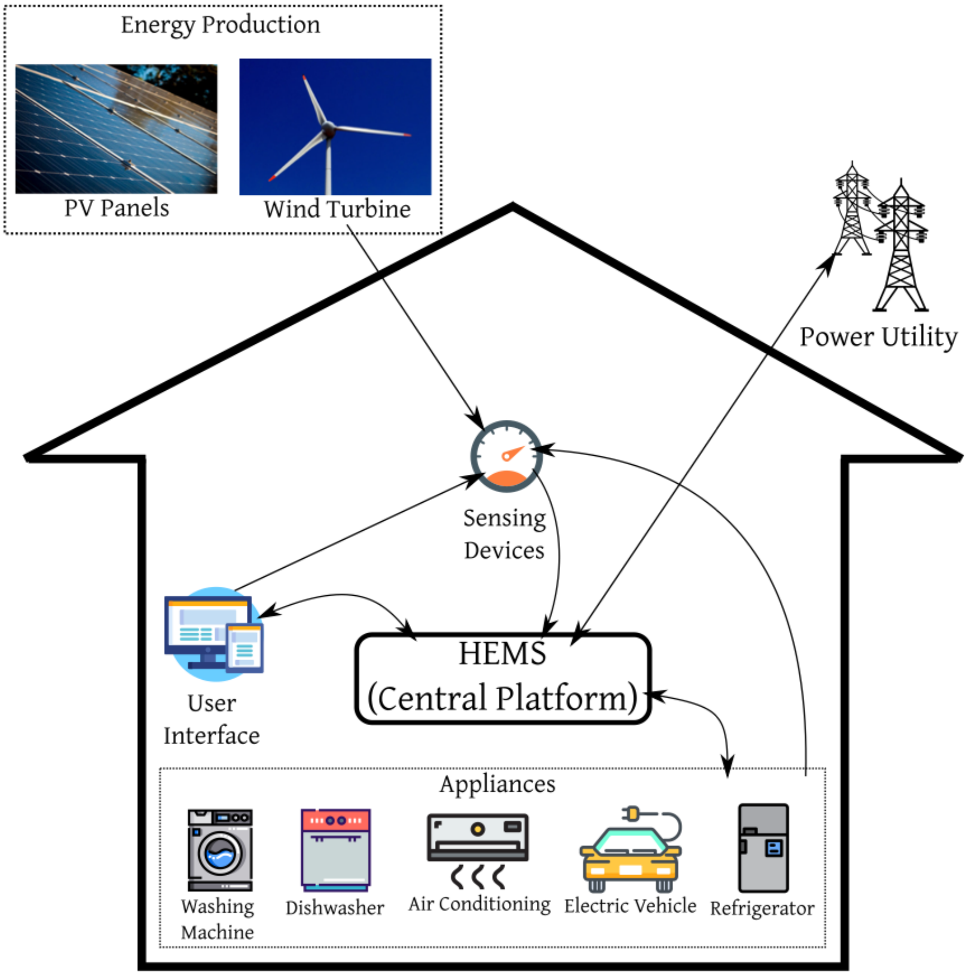


Figure 2.2: Common topology of a (Home) Energy Management System network [LGRC20]

- **Energy Control and Optimization.**
  Based on the data analysis, the system actively regulates energy consumers in order to promote cost efficiency, environmental protection [MSO17] or crisis safety. This can be done automatically or by manual intervention.

- **Reporting and Notification.**
  An EMS can generate reports on the building's energy consumption and efficiency. The system can also provide notifications of unusual consumption or technical problems.

Benefits

- **Energy Saving.**
  Annual energy consumption and peak power consumption can be reduced through the targeted use and storage of self-generated energy, as well as through power reduction by consumers.

- **Environmental Protection.**
  Alternative and renewable forms of energy help reduce emissions. ESCs can only generate renewable energy in balance but cannot supply it in a targeted manner. When self-generated energy is used, the origin is obvious. In addition, attention can be paid to which components are installed at the time of purchase and thus more environmentally friendly devices can be obtained.

- **Cost Savings.**
  After a higher initial investment, the running costs are usually much lower than with conventional types of supply. There are also subsidies for renewable forms of energy in various countries [AMM15].

- **User Motive.**
  Depending on what the user wants to achieve with the EMS, the system can control the components in such a way that comfort, self-sufficiency or other motives are fulfilled.

- **Analysis.**
  Until now, annual energy consumption has been known to be the only absolute value. Thanks to the sensors and components used in the system, much more accurate reports can be generated and analyzes and precautions can be taken based on them.

- **Equipment Protection and Preventive Maintenance.**
  Usage-based maintenance work, allocation of loads to equivalent components and much more are made possible by centralized data collection.

Challenges and Concerns

- **Integration and Compatibility.**
  One of the biggest challenges for EMS is the integration of different systems and technologies within a building [AKM22]. Many buildings or installations built at an early stage of automation have outdated systems. These are often not compatible with modern EMSs, ESs and HASs. This results in extensive, costly modernization or the use of interfaces.

- **Data Quality and Availability.**
  The efficiency of EMSs depends heavily on the quality and availability of the

collected data. Inaccurate, incomplete or delayed data can significantly impact the effectiveness of EMSs by encouraging incorrect analysis and decisions. Data must also be normalized to compare it.

- **Security.**
  Equivalent to HASs and ESs, data security is an important issue. Many EMSs are also operated in the cloud and are therefore under critical scrutiny by some interested parties.

Further information, researches and limitations, as well as concrete approaches are discussed in section 2.5.

2.4 Challenges and Possibilities

Research and projects that promise a link to or direct improvement of the environment and sustainability will generally be the most important and promising topics in the coming decades. This is confirmed by a study by the management consulting company McKinsey [Mei23]. If we limit the scope to energy, or even more precisely to electrical energy, new challenges, but also opportunities will arise in the coming years. Energy consumption will continue to increase in the coming decades [IEA], but the production of energy in the form of large power plants will not be able to keep up to the same extent. In addition, this energy must not only be produced but also transported. This is not possible through selective expansion, but must be done across the board, right down to the individual buildings. The following points list and discuss some of the challenges, but also the outlook and opportunities.

2.4.1 Grid Flexibility and Stability

The grid operated by the ESCs and the power plants behind it conventionally supply buildings with electrical energy (see figure 2.3). Simplified sequence of ESCs production and distribution cycle [Wik24b]:

1. Demand for energy.
2. Power plants produce more energy or energy is purchased.
3. Energy is provided to the grid and transformed into transportable energy.
4. Energy is transformed to usable energy at a substation.
5. Energy is distributed to the power lines and can be drawn.

To produce more energy, the power plants must be operated under higher load or more power plants must be activated. However, there are differences in power plants. The base

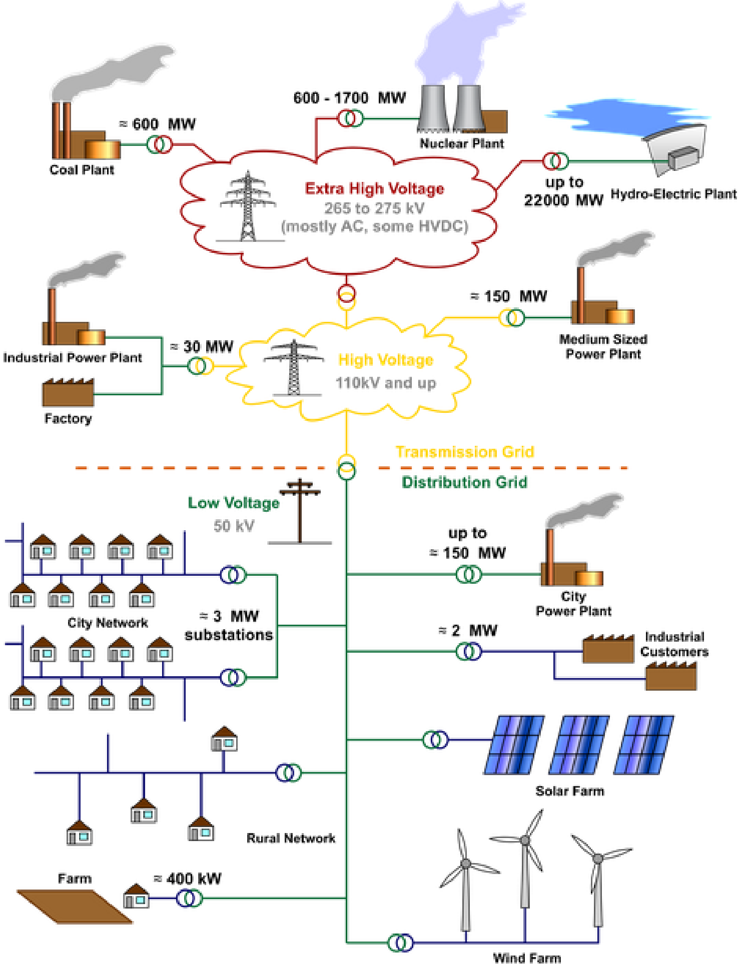


Figure 2.3: General layout of electricity grids. Voltages and depictions of electrical lines are typical for Germany and other European systems. [Wik24b]

material for producing energy is not only the base material but also the speed at which energy is provided and the flexibility [MS17]. There are lazier and more rapid power plants [IVNO20]. A rapid increase in energy production in the short term is therefore a very complex process.

Once the energy has been produced, it must be supplied by the grid. This consists of a comprehensive network of lines that extends to the individual buildings (see figure 2.3). These lines have line resistances and associated maximum loads that can be transported. Higher output is therefore only possible by expanding the network. This is more complicated than with power plants, as it affects not only a local, but the entire geographical area. Roads have to be dug up and lines added. This is a very large, costly and traffic-impacting effort.

However, according to the World Energy Outlook [IEA] of the International Energy Agency, energy demand will increase significantly in the coming decades. This is due to the increasing number of technologies and innovations and also to the phasing out of various types of energy and the promotion of electromobility. Due to the regular additions and expansion of the existing grid in recent decades, the energy grid is already very unstable and the risk of partial grid failures in the medium term is increasing. In addition, the demand for flexibility in the grid is increasing. Industrial buildings with machines and large consumers have long been taken into account when planning the grid. Their high starting currents and short-term voltage peaks have also been known for a long time. However, private buildings and office buildings have so far been regarded as constant and lazy consumers. Through various applications, but above all through electromobility and the associated charging, new large consumers are emerging in private households and office buildings. More on this in section 2.4.2.

The grid and the energy production behind it will therefore pose a major challenge in the near future [OB16]. In addition to expansion and extension, other solutions are also needed to distribute and outsource the grid.

2.4.2 Electric Cars

More and more people are using electric cars (e-car) [AO15, Age]. Political traffic and environmental planning in many countries is also based on electric vehicles. These require fewer fossil resources and can be charged using renewable energy, either at public or private charging stations. At private charging stations, an electric car is often charged with 11 or 22 kilowatts. This is a power consumption that no other consumer in a household has for a longer period of time. The high consumption is required in a very short time [Neu21] and without warning the grid and therefore requires high flexibility from the grid.

Furthermore, the usage behavior of electric vehicle users is usually very similar [FHNB19, QTOL15]. This means that high power is required occasionally and that it reaches a high level at certain peak times. For example, in the evening when everyone arrives at home. As the car is often driven to work, a large part of the renewable energy generated by PVSs [Sch15] cannot be used to charge the electric car during the week. Surplus energy is fed into the grid and needed in the evening when no renewable energy is available. For more details, see section 2.4.3. Charging electric cars is therefore a major challenge for ESCs. This is true especially since the growth of e-car use and charging is increasing rapidly.

If you look at this problem not from the perspective of ESCs, but as a building operator, there is a similar bottleneck: The power provided by ESCs for the building. This is based on various factors: What load can the power line carry, how is the line protected, how much power was agreed in the energy supply contract. Based on this and the usual electricity consumption, the available power that can be used to charge electric cars is calculated. As the available power is often not similar to the scaling of e-charging stations (e.g. multiples of 11 kW), care must be taken to ensure that the charging stations do not draw too much energy.

Local load management for chargers can help here [AST11]. It monitors the individual charging stations and their current power consumption [OFHB16]. This is compared with the available energy and the charging stations' power consumption gets limited. This is one of many approaches to relieving or protecting a power grid, locally or globally. However, it does not solve all problems, as it cannot provide more energy than is supplied if there are too many simultaneous charges. As a result, many electric cars charge very slowly over many hours.

2.4.3 Bidirectional Energy Exchange

The conventional electricity grid is a unidirectional network. With alternative forms of energy, many buildings produce energy themselves and will add surplus energy to the grid [Fou24]. Therefore, the grid will be a bidirectional system in future (see figure 2.4). However, the alternatively generated energy is usually very inconsistent and depends on various uncontrollable factors. Wind, sun and clouds play a major role. This means that a lot of energy is produced at certain times. Usually the energy amounts to so much that you cannot use it yourself and therefore want to sell it. This is possible by concluding contracts with PVSs and using smart meters to measure the energy exchanged. However, the energy sold, from the PVS surplus, is priced much lower than the energy purchased.

In addition to economic factors, mutual exchange causes the grid to fluctuate and destabilizes it. The PVSs can only use forecasts (sunshine, wind, etc.) to make assumptions about how much energy will be fed into the grid by the buildings and how much will have to be produced by the power plants. This is a very complex calculation. Furthermore, renewable forms of energy are rapid and cannot be controlled. So if there is a large-scale change in the weather or a similar incident, the PVSs lazy power plants have to react faster than they possibly can. Amongst others, these are major challenges that need to be considered in the bidirectional exchange of energy.

2.4.4 Load Management and Variable Grid Pricing

The challenges of available and produceable energy mentioned in section 2.4.1 could be solved in the future by many different approaches. One of these approaches is called Green the Flex [Com, Mat], a cooperation project between EVN AG, EVN Energievertrieb GmbH & Co KG and CyberGrid GmbH, and is an innovation program funded by the European Commission. The aim is to combine 3000 private households in Lower Austria with a PVS and energy storage system from Fronius [Gmba] into a swarm storage system by 2025. The external control of the batteries is intended to avoid load peaks in the grid. Based on forecasts of the expected generation of PVSs, the use of the energy storage systems and the grid, the network is relieved by charging and discharging the batteries.

The approach distinguishes between good weather operation and bad weather operation. In good weather operation and an additionally expected PVS high due to weather

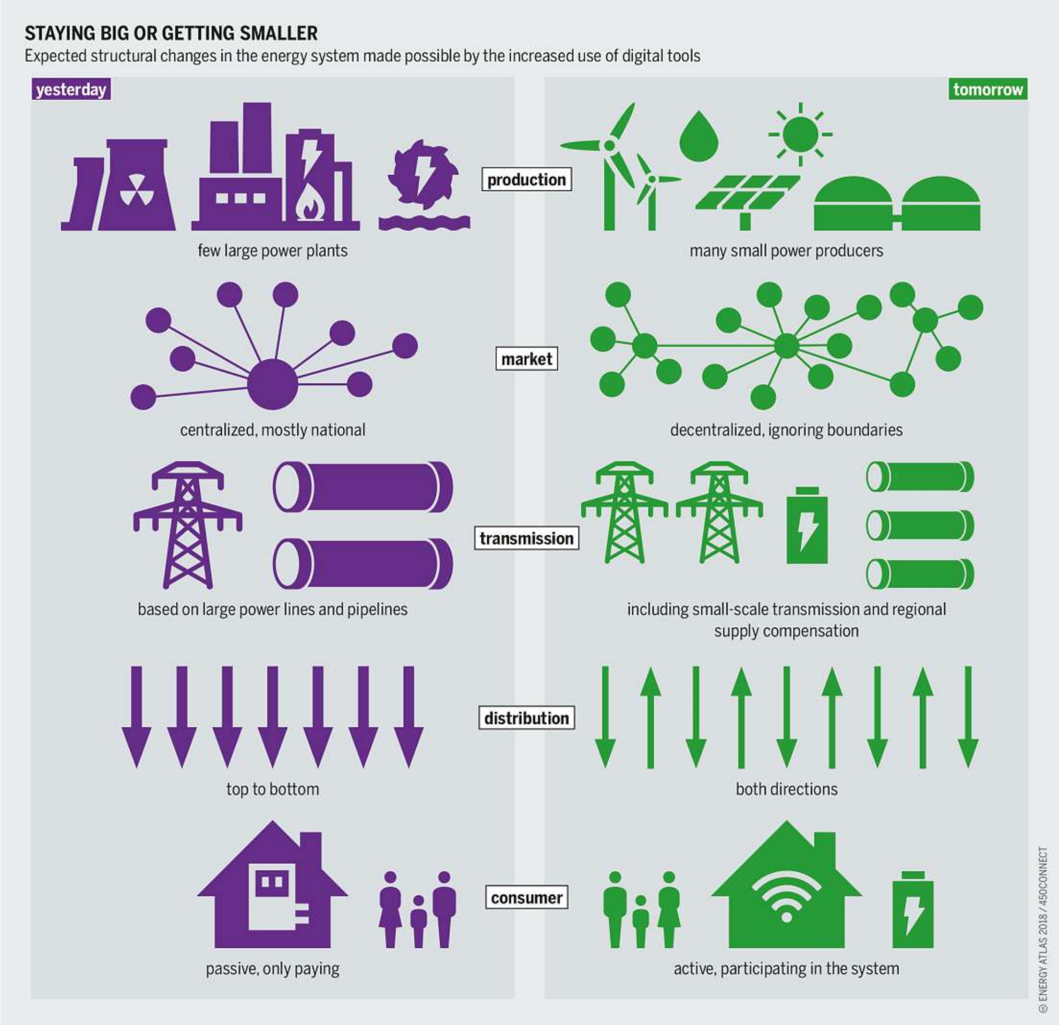


Figure 2.4: Expected structural changes in the energy system made possible by the increased use of digital tools. [Fou24]

forecasts, battery charging is delayed in the morning hours. The low surplus is fed into the grid and only the high overloads at midday are loaded into the empty energy storage. In the evening hours, the energy storage is used for own use but also for grid relief.

During bad weather, surplus renewable energy is stored in the batteries in the grid so that it can be discharged again when energy is short. This is an additional advantage if you have a day/night tariff (see section 2.4.5) and the energy storage is charged at night by e.g. surplus wind energy.

However, other more radical methods and precautions are also possible to ensure grid stability in the future. Similarly to the electric charging load management mentioned in section 2.4.2, the ESC could use load management at building level [SJL15, EMASA17].

The power of the house connection could be limited variably. Depending on how much energy the ESC can supply, the building receives a certain amount of power. If such a system is used, it is almost essential to install an EMS in the building that distributes and uses the available energy as effectively as possible.

Variable prices for grid fees are also currently the subject of much discussion [SHGS17, MJKR15]. Not only the amount of energy supplied, but also the actual output is taken into account in the billing. This increases the motivation of users to strain the grid evenly and distribute consumption throughout the day. However, this does not provide a guarantee for the grids relief. If people accepted that their electric car can only be charged in the evening, for example, this would only partially relieve the grid.

2.4.5 Day/Night Tariff Models

ESCs try to prevent grid surpluses and grid bottlenecks through various precautions. Many of the power plants operated are very lazy and, therefore, produce energy 24 hours a day. This circumstance is not only due to the laziness of power plants, but also to evenly distribute the load throughout the day. In addition, ESCs have had tariff models for decades in which the electricity price is different during the day than at night [AMM15]. The advantage of this tariff in combination with energy storage has already been mentioned in section 2.4.4. Regardless of the project, you can also use this to your advantage as an individual building. With the help of an EMS, the loads can be shifted to the more favorable tariff time. In addition, the energy storage can be charged with the cheap tariff and used at times of the expensive tariff phase. Use case 3 from section 5.3.4 represents exactly this case.

2.4.6 Renewable Energy Obligation

In Austria, Germany and many other countries, more and more obligations for renewable energy are emerging. In some federal states in Germany, it has been mandatory since 2023 for new buildings and roof renovations to have a PVS or solar thermal system installed on the roof [noa23, Koc]. In Austria, the installation of new gas heating systems has also been banned since 2023 [BfK]. A turnaround for the environment, simultaneously resulting new challenges and great opportunities.

The guidelines will not become less stringent in the coming years. It can therefore be assumed that more and more existing systems will have to be replaced, supplemented or extended. These extensions are often state-of-the-art systems with many possibilities. In order to exploit these, older, existing or analog installations and systems must be combined and merged with the new ones. This can be very expensive or can be exploited by manufacturers to sell additional modules. A Modular Energy Management System can counteract this. By using a wide variety of systems, sensors and actuators, systems can be combined as well as possible.

There are many challenges to solve, but there are also many new opportunities that arise from new technologies and innovations. An EMS is not the solution to all energy problems, but it does help to take a step forward and optimize the use of existing systems.

2.5 Related Work

Research and industry have been working on EMSs for many years. Various approaches and different models have been designed and tested in recent years. Some of these designs are briefly summarized here.

T. Maragatham et al. in "IoT Based Home Automation System using Raspberry Pi 4" [MBV21] developed an HAS that connects different systems using the TCP/IP stack; centrally controlled, executed on a Raspberry Pi microcomputer and displayed on a LCD screen. In addition, it is possible to view the data via the internet by accessing the microcomputer's storage. The precise control, data processing and normalization are only dealt with very shallowly. However, it is evident that this approach is only based on TCP/IP components. In addition, complex automation is not discussed. Nonetheless, the approach proved to be a success and is an interesting method for connecting different systems using TCP/IP.

In "Remote control of a domestic equipment from an Android application based on Raspberry pi card", H. Lamine et al. [LA14] also use Raspberry Pis. Although in this case, they are used as sensors and actuators. This means that there are no restrictions on communication protocols. Using SQL, the individual Raspberry Pis communicate with a central database, which can also be hosted on a microcomputer. A web interface is addressed using an Android client application and REST calls, allowing the values to be visualized on cell phones and tablets. Further front-end programming allows for more complex automation, but this was not the scope of the approach. Restrictions or additional work arise with this approach due to the exclusive use of microcomputers as components. This means that no existing or externally developed components can be used. The effort required to develop an application for every use in a building is immense. Still, the experiment proves the function of a central data collection point in controlling various components.

A very similar approach was developed by P. Pawar et al. in "Design and development of advanced smart energy management system integrated with IoT framework in smart grid environment" [PVK19]. They included the option of using different protocols such as ZigBee, Wi-Fi or Bluetooth separately or simultaneously. A central server controls the components and executes simple automation steps. More complex logic connections must be supplemented by external services or additional development steps. The visualization of the data was developed as an HTML front-end using a REST interface.

A concrete EMS was used in a field test in Freiamt, Germany [Exn23]. 23 households were equipped with the EMS "PSIngo" [noab] from PSI GridConnect. More than 40 measuring points were installed in each building and connected via a smart meter gateway. The test was carried out for 17 months and showed a useful result. With the help of the information obtained, energy bottlenecks were avoided and the grid was relieved.

However, in addition to this main objective, secondary findings were also obtained. One of these was compatibility problems with sensors and smart meter gateways. Similar devices with different software generations or existing components could often not be used or required intensive intervention. It turned out that the system has sufficient functionality, but struggles with interoperability.

S.-P. Tseng et al. solve the problem of interoperability in the HAS category in "An Application of Internet of Things with Motion Sensing on Smart House" [TLPL14]. In this approach, the components communicate with each other via ZigBee and have a connection to the developed Smart House Monitoring and Management (SHMM). This is hosted in a cloud specially designed for ZigBee gateway applications. By accessing the SHMM, the components can be controlled and further automation can be configured. There are limits to this approach due to the limitation to ZigBee devices. This could mean that older devices cannot be integrated or that current components do not support ZigBee. In addition, there is no guarantee of data security and the system is dependent on the cloud operator.

In "Design and Fabrication of Smart Home with Internet of Things Enabled Automation System" by W. A. Jabbar et al. [JKR+19], the components are connected to a central controller using MQTT. Various HMIs can be connected to the EMS using independent methods. MQTT is widely used and is supported by many manufacturers and protocols. Nevertheless, the system is limited due to MQTT. The advantage of this system is the possibility of a high factor of high complexity in automation. The approach integrates the web-based service IFTTT ("IF This Then That")[IFT]. This makes it possible to configure various action sequences and thus enables a wide range of functions. The disadvantage of this method is, once again, the dependency on the web service and the running costs that IFTTT charges.

The referenced approaches are just a few of the scientific attempts to connect HASs and design an EMS. Different models and approaches are chosen depending on the access and scope of the research. However, all approaches are limited in at least one of the motives mentioned in section 1.2. Regardless, by researching the selected and rejected papers, some knowledge and ideas for model design were gained.

CHAPTER 3

Model Design

This thesis covers the entire development cycle of a system. The externally defined problems and limitations are stated in section 1.1. Based on these problems, the main functionalities and requirements are specified in section 1.2. In regard to the requirements, consideration must be given to how a system can be expanded as far as possible and be open to new or other technologies. To this end, various models are developed that connect different systems with each other. To simplify this, an exemplary infrastructure is defined that represents a cross-section of a current Energy System. All models are analyzed, rated and compared with each other to find an appropriate model to implement as a Proof of Concept. It subsequently gets validated by use cases.

3.1 Methodology

The methodological approach consists of the following steps:

1. **Problem Statement.**
   Due to the problem definition specified for the Master's thesis, the problems are formulated textually for clarification and unambiguous definition.

2. **Aim of the Work.**
   The type and scope of the thesis are defined. Based on the desired verification method Proof of Concept, the functionalities and tasks of the final model and the implementation based on it are specified. These are confirmed by the supervisor.

3. **Literature Review.**
   Backgrounds and concepts from scientific theses, journals and companies in the field are researched. These serve the design phase for the models and show the underlying problems of current systems.

4. **Model Design.**
   Different approaches, both from literature research and on the basis of what has been learned during the studies, are designed. These are discussed and improved with the supervisor.

5. **Model Comparison.**
   The designed models are compared with each other and evaluated. Various factors and circumstances are taken into account during the comparison, like interoperability, scalability and security.

6. **Proof of Concept Engineering.**
   Based on the final model, a Proof of Concept is developed.

7. **Evaluation.**
   Use cases are applied to verify whether the PoC fulfills the objectives of the thesis.

3.2 Model Fundamentals

Various models are designed to implement and verify the Modular Energy Management System. The model designs can be divided into the following components in advance:

- **Communication Model.**
  One of the central tasks of the MEMS is to connect the various systems. The protocols, transmission media and message formats are defined here. The more protocols are supported or can be extended, the higher the interoperability of the MEMS will be.

- **Logic.**
  The logic is the executive element of the system. It stores current and past values of the components, sends and receives data and executes the automation of the system. Before the final decision on the model is made, it cannot be defined whether it will be a centralized, decentralized or hybrid solution.

- **HMI Model.**
  A user-friendly interface to control the system, configure it, monitor it and manage automation rules.

The communication model and the executing logic play together. Based on the way component types are used for communication, it is partly automatically defined how the logic controls the MEMS. The HMI model, on the other hand, can be primarily seen and developed independently. For this reason, the communication model and the logic were considered and developed separately from the HMI model.

3.3 Fundamental Infrastructure

In order to have both a practical verification and a simpler understanding of the requirements, an exemplary basic building infrastructure was defined. The added parts are different components used in the household and in home energy production. They are exemplary and are intended to represent an average of different possibilities. The components used do not necessarily have to be implemented as concrete components in the implementation phase.

There are conventional consuming components, hybrid components that can consume as well as provide energy and components for producing, converting and providing energy.

Table 3.1 lists the exemplary components of the fundamental infrastructure.

Figure 3.1 shows the fundamental infrastructure with the components used. Based on these components, the infrastructure was divided into the Home Automation Systems layer (grey) and Energy Systems layer (green). Components of the HASs layer are variable in usage and replacement and a higher component scaling can be assumed. The ESs layer components primarily organize and distribute the available energy and supply the entire building with energy. They are connected to a 40 Volts Direct Current (DC) line. The inverter converts the Direct Current into a 3-phase 230V/400 Volts Alternating Current (AC) and supplies energy to the building. The ESs layer can also be extended, but it is assumed that this occurs less frequently than at the HASs layer.

Component	Type	Protocols	Description
E-Charger 1	Bidirectional	TCP/IP or KNX	Generic e-car charging station
E-Charger 2	Bidirectional	TCP/IP or KNX	Generic e-car charging station
Refrigorator	Consumer	-	Ordinary generic household refrigerator
Washingmachine	Consumer	TCP/IP or KNX	Smart generic washing machine
PV-Panel	Producer	-	Generic photovoltaic panels
Grid	Bidirectional	-	ESC grid connection
AC-Generator	Producer	-	AC generator with AC/DC converter
Battery	Bidirectional	-	Generic battery
Charging-Management	-	UART	Generic charging management
Studer Innotec XTM Inverter	-	UART	Connects and organizes all generators on a 40V DC busbar
Studer Innotec VS120	-	UART	Solar charge controller
Studer Innotec XCOM LAN/CAN	-	TCP/IP or CAN	Propritary Gateway from Studer Innotech communication network to LAN/CAN
Studer Innotec BSP	-	UART	Battery monitor

Table 3.1: Fundamental Infrastructure Components

3.4 Communication Models

3.4.1 Communication Model 1

Communication model 1 (see figure 3.2) is a centralized approach based on a High-level Programming Language (HLP). The logic is hosted on a server and the data is stored on a local database. This can be a conventional computer, a microcomputer or could even be hosted in the cloud under certain conditions. The main communication protocol is a TCP/IP stack connected via LAN. Various messages from different systems can be exchanged on this stack. Systems that have a Representational State Transfer (REST) or TCP/IP connection only need to be implemented in the HLP. Other systems, such

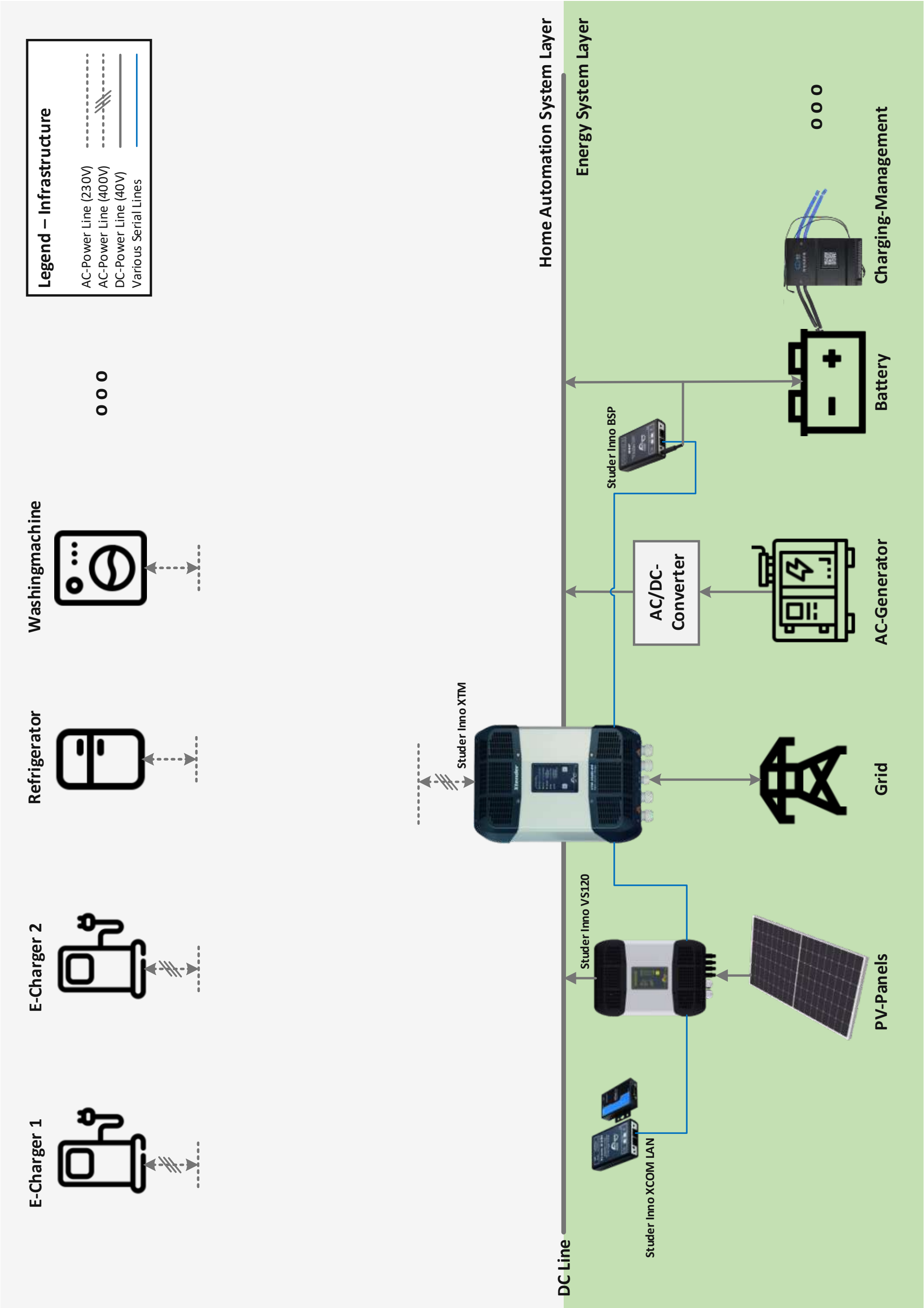


Figure 3.1: Fundamental Infrastructure

as KNX, can be connected to the basic network using an external LAN gateway (e.g. Weinzierl KNX IP BAOS 777[1]). This underlines the modularity of this model. Almost all systems and protocols can be added to the system by expanding hardware modules and the associated HLP implementation. However, the expansion of hardware modules is only possible for locally operated systems. For cloud-based systems, the optinal hardware modules would have to convert the protocols to TCP/IP in order to send them to the cloud.The HMI is connected via LAN and can therefore be used in various ways.

Referring to the fundamental infrastructure, the HASs layer can be expanded modularly and flexibly. The ESs layer is connected by gateways. The components from Studer Innotec use a proprietary communication module as a LAN gateway (Studer Innotech XCOM LAN). The battery storage and battery management are converted from serial to LAN using an ESP 32 microcontroller and integrated into the system. The generator is controlled by a KNX actuator.

Communication Model 1	
Computing	Central
Logic	High-level programming language on server
Protocols & Systems	TCP/IP (e.g. all REST controllable systems), KNX (Gateway: Weinzierl KNX IP BAOS 777[1])
Storage	Local database
HMI	Various
Extendability	Every TCP/IP-based system can be implmented; Various protocols can be extended with hardware and implemented in the high-level programming language

Table 3.2: Communication Model 1 Description

1: Weinzierl KNX IP BAOS 777: With a Bus Access and Object Server (BAOS) gateway KNX objects can be conervted into JavaScript Object Notation (JSON) objects bidirectionally. These are sent via REST or websockets on a TCP/IP stack. Therefore a bridge can be created between KNX and TCP/IP. This gateway is event-triggered and displays all data updates on a web socket. Request for data or changes can be done by REST calls. In addition, it also offers a web interface for visualizing and controlling the assigned group addresses.

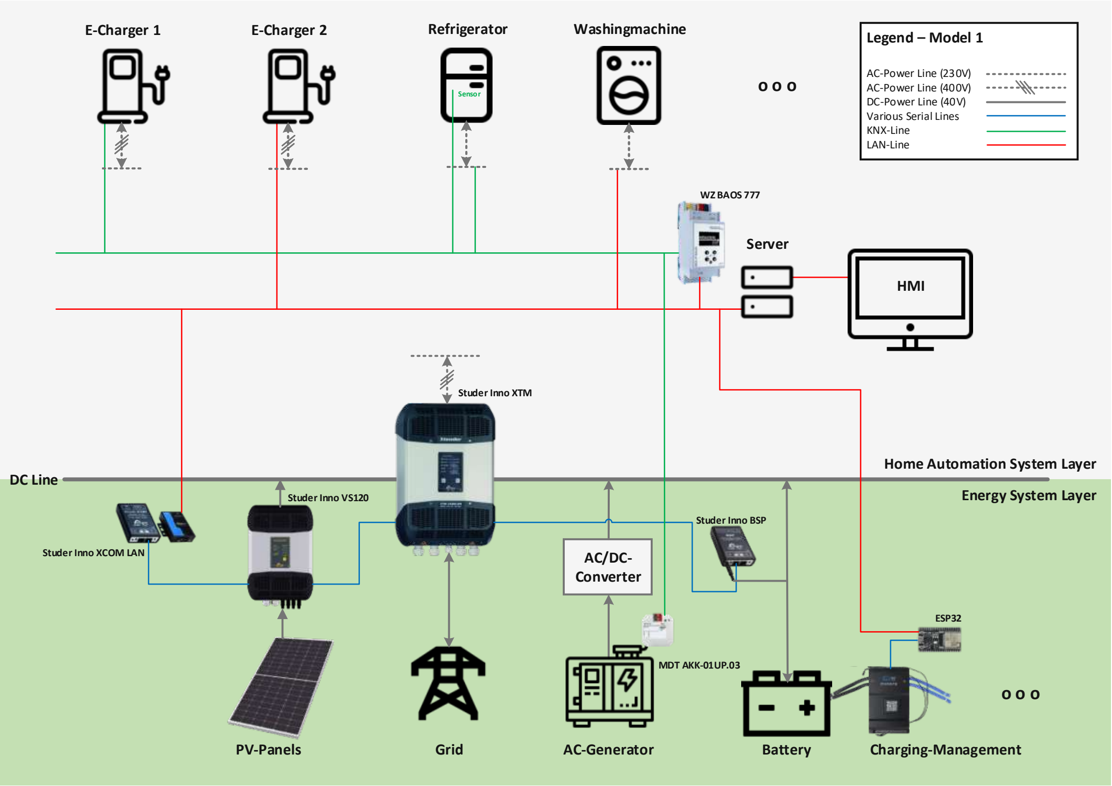


Figure 3.2: Communication Model 1

3.4.2 Communication Model 2

Like communication model 1, communication model 2 (see figure 3.3) is also a centralized approach based on a HLP with a local database as energy storage. However, communication between the components is limited to one protocol. The entire exchange of messages takes place via KNX. All messages are converted using a KNX BAOS gateway and connected to the central server. The benefit here is to use the reliability of the KNX industry standard, but to retain the calculation and configuration possibilities of a HLP system. In addition, the data can be made available to the HMI in a simple and suitable way. Control and visualization in various ways is thus possible. If further systems are to be integrated, these must be connected and converted via a KNX gateway. It can then be configured and integrated in the KNX configuration mask Engineering Tool Software (ETS). The advantage is that nothing has to be changed in the logic and the control of an additional system can be done without active engineering.

Due to the limitation to KNX, suitable gateways must be used in the fundamental infrastructure. Sensors and actuators with KNX functionality are used at the HASs layer. However, other HASs are partially excluded or can only be integrated into KNX at great expense. However, gateways are available for common HASs [Assb]. The ESs layer, which also had to be converted in communication model 1, is connected via various KNX gateways. The Studer Innotech components, which have an independent communication network, are integrated into the KNX system using a Controller Area Network (CAN) [Wik24a] gateway (Studer Innotech CAN and KNX CAN BUS gateway). The generator is controlled using a KNX actuator and the battery management is converted from serial to KNX using the Weinzierl KNX RS-485 88X module.

Communication Model 2	
Computing	Central
Logic	High-level programming language on server
Protocols & Systems	KNX (Gateway: Weinzierl KNX IP BAOS 777[1])
Storage	Local database
HMI	Various
Extendability	If exists, special KNX modules used as gateway can be used to extand the system.

Table 3.3: Communication Model 2 Description

3.4.3 Communication Model 3

Communication model 3 (see figure 3.4) is based on KNX. Both the communication and the logic and control are implemented using KNX, therefore no HLP is required. All configurations are made in ETS and executed decentralizedly by KNX. The system is therefore very robust and secure, as it has been the industry standard for years. KNX is based on the functions of the individual components. To implement more extensive automations and sequences, it requires so-called logic modules . Depending on the model, these can define boolean logic, control time sequences or store and log values. The MDT SCN-LOG1.02, for example, is a logic module with a wide range of functions and manages the complex automations of the model. The whole model can be controlled and visualized via a KNX-integrated (touch) display.

As with communication model 2, the same challenges arise when integrating the HASs and ESs into KNX. KNX gateways are also used in this model. An explanation of this can be found in communication model 2 (see section 3.4.2).

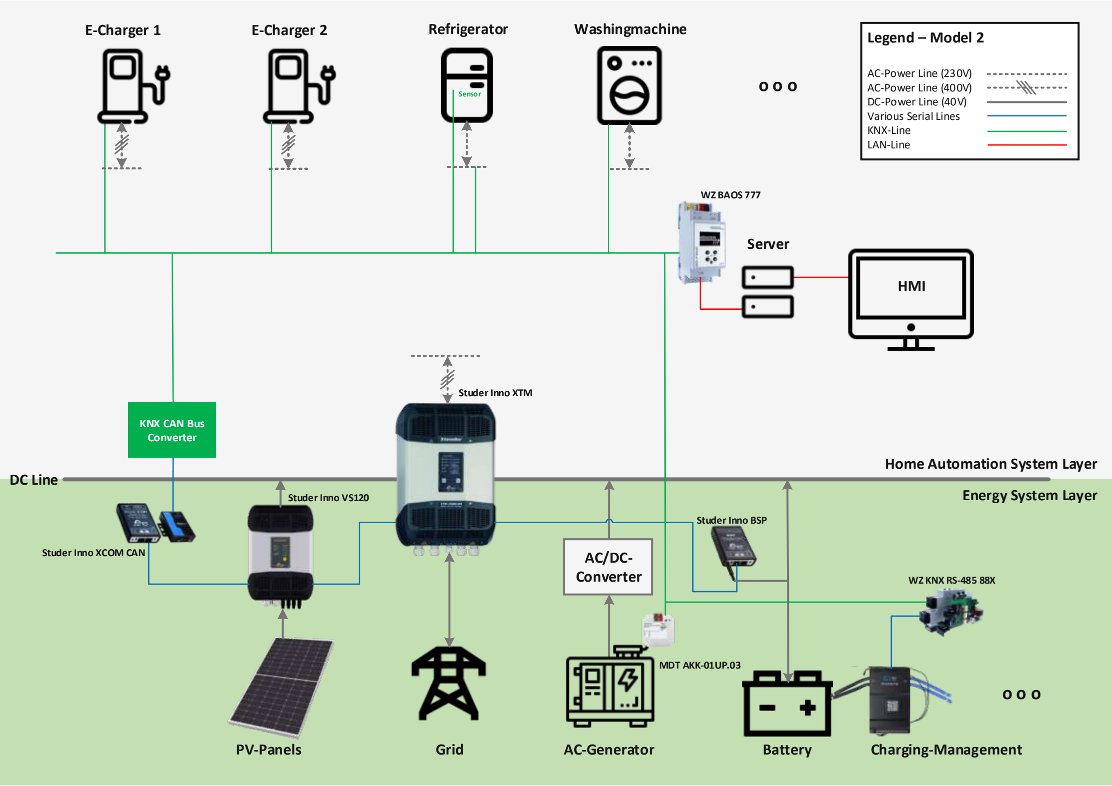


Figure 3.3: Communication Model 2

Communication Model 3	
Computing	Decentral
Logic	KNX logic modules
Protocols & Systems	KNX
Storage	KNX proprietary
HMI	KNX touchdisplay
Extendability	If exists, special KNX modules used as gateway can be used to extand the system.

Table 3.4: Communication Model 3 Description

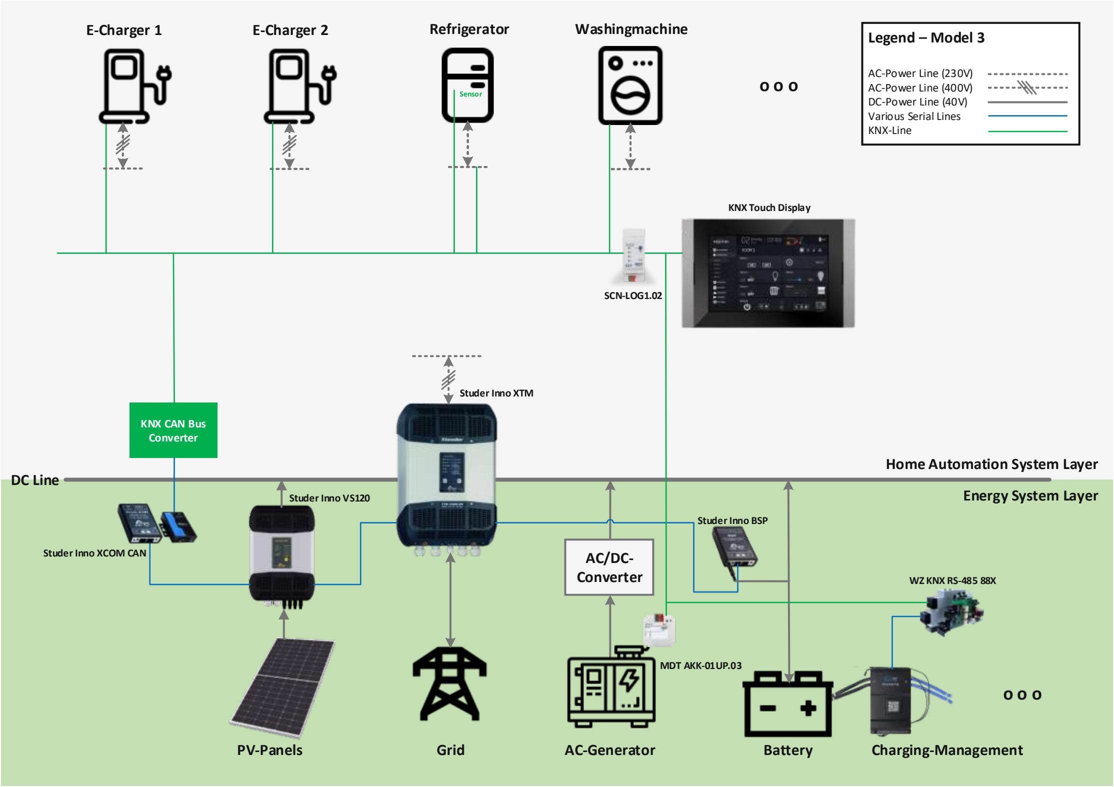


Figure 3.4: Communication Model 3

3.4.4 Communication Model 4

Similar to communication model 1, communication model 4 (see figure 3.5) is based on centralized control using HLP and the TCP/IP protocol. It is limited to the TCP/IP stack and does not support any other communication protocols. However, it enables a large base of systems right from the start, as many systems have a REST interface. In addition, the system can use edge-based gateways and converters to convert other protocols and connect them to the TCP/IP network. Still, there must be a reformatting of the message format for each system that is integrated. This means that every newly extended system, whether TCP/IP or not, must be implemented in the HLP. The HMI is flexible and uses REST to connect to the logic, control the components and visualize the data.

In terms of the fundamental infrastructure, the HASs layer is implemented using REST-based components. These are widely used in HASs and limit the system just slightly. At the ESs layer, the Studer Innotech components and the battery management are connected as in communication model 1 using a LAN gateway and ESP 32. The generator is operated for the first time using ESP 32 and a serial switching actuator.

Communication Model 4	
Computing	Central
Logic	High-level programming language on server
Protocols & Systems	TCP/IP (REST)
Storage	Local database
HMI	Various
Extendability	Every TCP/IP-based system can be implmented; Edge-based microcontroller can translate other protocols to TCP/IP

Table 3.5: Communication Model 4 Description

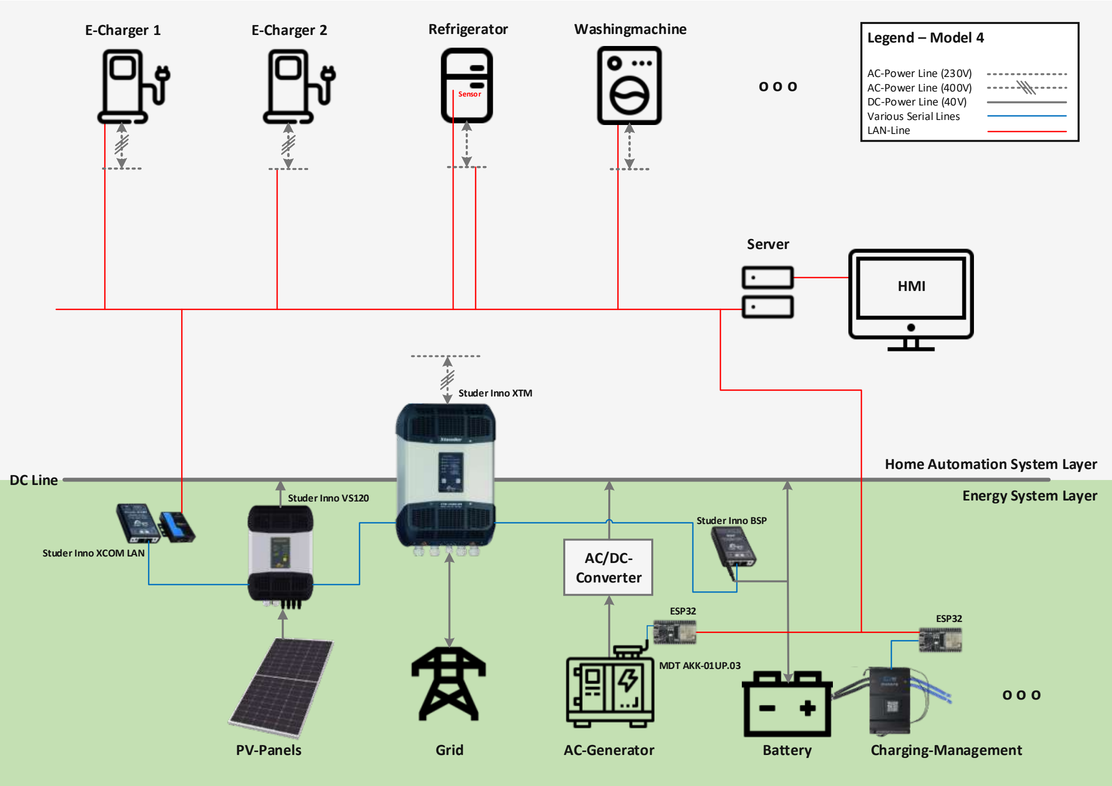


Figure 3.5: Communication Model 4

3.4.5 Communication Model 5

Communication model 5 (see figure 3.6) keeps the logic centralized, but must host it on a microcomputer. Communication with the components is established via a serial Universal Asynchronous Receiver Transmitter (UART) interface. This means that a serial line is laid to each component, which participates in UART. The logic stores all changes and can control the various components via the UART line. Depending on the approach, storage can take place using a database structure or local storage. Each value must be measured or controlled using micro electronic components and implemented in the central logic. The sensors and actuators are presumably very inexpensive, as no end-user development work has been done and no licenses have to be paid for ecosystems-acceptance, etc. However, values and controls that are available independently from some components cannot be used, as most systems do not have a UART interface. The HMI can be viewed independently and visualized either via REST and a front-end or via local displays.

The effort required for integration into the fundamental infrastructure is high. Each measurement and each actuator at the HASs layer must be developed and implemented individually. At the ESs layer, the Studer Innotech communication network is integrated directly into the UART. The battery management and the generator are controlled via ESP 32.

Communication Model 5	
Computing	Central
Logic	High-level programming language on Microcontroller (e.g. Raspberry PI)
Protocols & Systems	Serial line (UART)
Storage	Local storage/database
HMI	Various
Extendability	Edge-based microcontroller can translate other protocols to UART

Table 3.6: Communication Model 5 Description

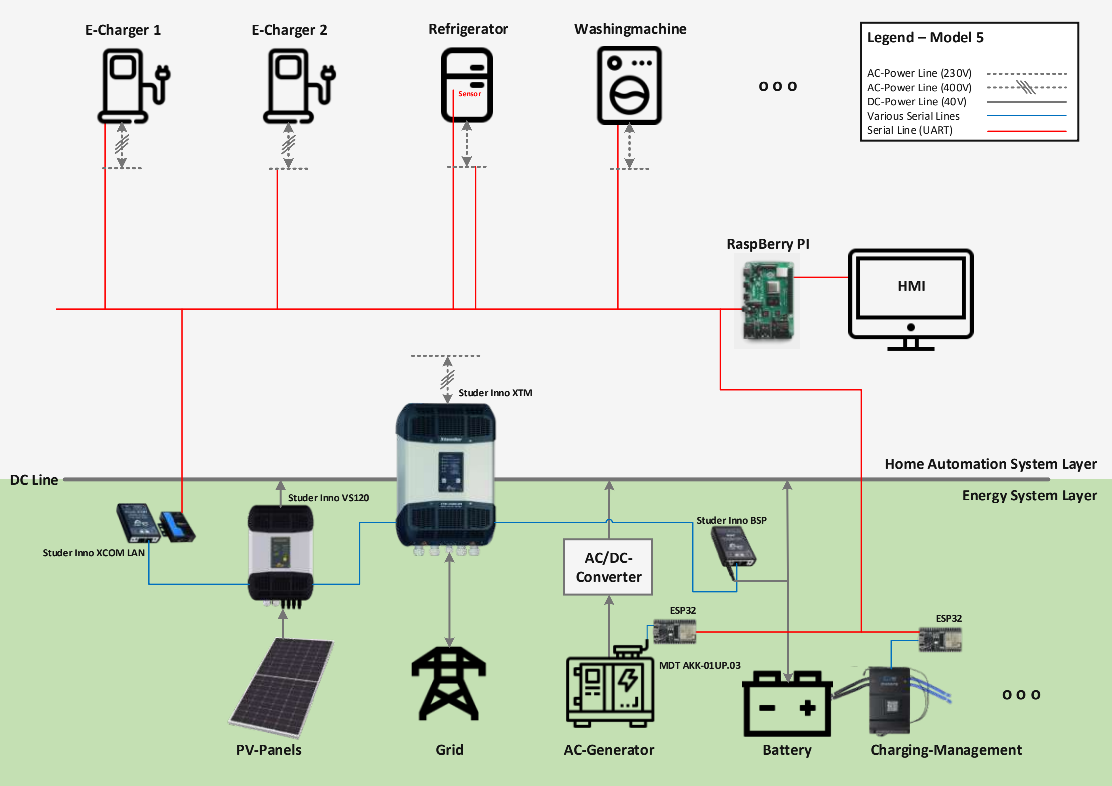


Figure 3.6: Communication Model 5

3.5 HMI Models

The Human-Machine Interface models are used to manually control the system and visualize the data, as well as to configure of the system. As, with a few exception, this does not affect the communication network, the HMI models can be considered independently.

3.5.1 HMI Model 1

Based on a Nextion display, a touch display which is very easy to design and controlled directly by a microcontroller, the entire management system can be controlled via a LAN network or a UART. This model is therefore not suitable for the pure KNX model 3 (see section 3.4.3). Relevant information is displayed on different pages. The advantage of this variant is the closed nature of the system and the associated security. However, external control from outside the network and further automation are excluded.

The advantages are reflected in simple configuration and the associated good overview. These displays can be distributed at different locations in the building and thus offer several control points, as long as a LAN connectivity can be established.

HMI Model 1	
Visualization	Limited layered structure
Possible Devices	Supported (touch) displays connected by microcontroller
Flexibility	Only supported displays can be used; System is limited to the provided funntions of the layout design suite

Table 3.7: HMI Model 1 Description

3.5.2 HMI Model 2

A device-independent web platform can also be developed that evaluates the data packets (depending on the communication model selected) and then displays them graphically in any way.

Assuming that there is an external accessible web server or VPN connection, the system can be controlled from anywhere and centralized remote management can be created. However, the platform and the server must be adequately secured. This system is therefore not suitable for the pure KNX model communication model 3 (see section 3.4.3).

There are no limits to display types and automation, but the platform must be developed in-house or adapted using a Content Management System (CMS).

HMI Model 2	
Visualization	JSON objects objects visualized according to necessity and requirements by using a web front-end framework or other methods
Possible Devices	Any device having a broser or REST console
Flexibility	Different types of graphical viszualization can be done using extended frameworks or own designs. Based on the raw dataset, future technologies can be applied to upgrade the visualization in the future

Table 3.8: HMI Model 2 Description

3.5.3 HMI Model 3

Assuming that communication model 3 (see section 3.4.3) is selected, direct control can be carried out using KNX's ETS. By directly connecting the display to the KNX network, it visualizes all group addresses configured. The ETS also configures the display and its layout. Therefore, it can be simply designed. This model limits the visualization possibilities to the provided graphics and functions. Due to apps and graphic modules, it is additionally possible to use a phone or a tablet to control the system. The configuration of the system, the addition of new devices, and the automation have to be done directly in ETS and cannot be outsourced. ETS programming is mostly a professional task. Therefore, the usage and control of the system is very simple, an adaption or the addition of some automations need know-how and experts.

HMI Model 3	
Visualization	Limited layered structure and smartphone/tablet app
Possible Devices	KNX (touch) displays and smartphones/tablets, Computer using the ETS
Flexibility	Only supported displays can be used; System is limited to the provided funntions of the layout design suite or the app

Table 3.9: HMI Model 3 Description

3.6 Model Comparison

The designed models are compared with each other and evaluated. In the following sections, the communication models and the HMI models are compared and the results are described. However, the main focus is placed on the communication models, as these are more extensive, more complex and more important for the functionality. In addition, the number of HMI models is limited and can be evaluated by simpler analyses.

3.6.1 Communication Models

Comparison Parameters

The basis for comparing and evaluating the different communication models are the comparison parameters. These must be selected and defined in advance. Various parameters are taken into consideration, which take into account the development phase, economic aspects, costs and the environment. The entire cycle is considered; from development and production to installation, operation and disposal. The individual parameters are weighted and subsequently rated from 0 to 10 for each model. 0 is the worst rating and 10 is the best.

The following listed parameters were included in the assessment. The weighting of the individual parameters is given in brackets (1.0 is the standard value). The parameters and weightings were discussed with the supervisor in an intermediate step.

- **Costs (0.7)**
  The costs factor is included in several parameters. This parameter is specifically concerned with the costs of the components. Both the components for basic operation and logic and the individual sensors and actuators, which can be a variable amount, are considered.
- **Interoperability & Functions (1.1)**
  One of the main requirements is interoperability and the ability to incorporate as many different (legacy) components and systems as possible. The possible transmission media and communication models are evaluated together with the complexity of the respective integration. In addition, the range of functions made possible by the logic is included.
- **Installation Effort (0.4)**
  The effort required to install the logic, including the integration of the different systems, as well as the effort required to install and connect the individual components in the building and integrate them into the communication networks.
- **Reliability (1.3)**
  The reliability of the used and usable systems, protocols and transmission media.
- **Security (1.4)**
  The data and system security of the logic, control and visualization.
- **Component Maintainability (1.3)**
  Depending on the sensors, actuators and additional components used, can maintenance work, updates and minor troubleshooting be carried out by the end user or by experts and what effort is needed.
- **Logic Maintainability (1.0)**
  The effort of maintaining and extending the logic and the effort to configure and automate the components by the MEMS.
- **Extendability (1.5)**
  The possibilities of implementing other communication protocols or transmission media in the MEMS.
- **Manufacturer Dependency (0.5)**
  Which dependencies arise from the selected model in the basic functions and which in the operation of the sensors and actuators.
- **Development Effort (0.5)**
  The effort required to develop the PoC and thus a complete MEMS in the future.

- **Disposal (0.3)**
  What quantities and types of waste are generated, what costs are incurred and what effort and damage is involved in disposing of the various materials.

Communication Model 1

Parameter	Rating	Description
Cost	5	A server or microcomputer ($$) and the BAOS gateway must be purchased as the basic infrastructure ($$). Based on this, both the more expensive KNX solution and the inexpensive LAN solution can be implemented or further systems can be implemented ($).
Interoperability & Functions	9	The fact that nearly any HAS device and the KNX network can be added and expanded means that there are almost no limits to interoperability. Because of the HLP, almost any functionality can be implemented in the logic.
Installation Effort	5	The set up effort for the logic is low. Depending on the components, most TCP/IP components can be controlled by wireless. KNX is usually connected with data cables that have to be installed throughout the building. However, there are also alternative wireless solutions like KNX RF. The different systems must be configured in the logic and the group address of KNX must be assigned to the BAOS as a data point in the ETS with medium effort.
Reliability	7	If necessary, the industry standard KNX can be used in advance to ensure reliability. As a self-developed program, the logic can be seen as a reliable component.
Security	6	Due to the mixture of different systems, many attack vectors are available. However, the KNX network itself is known to be secure. The logic can be completely disconnected from the internet or protected from external logical access by means of authentication options.
Component Maintainability	8	The system is very easy to maintain, as most HAS devices can be maintained by yourself and many experts for KNX are on the market.

Parameter	Rating	Description
Logic Maintainability	4	The software is very complex, as different ecosystems have to be combined. This can only be done through the pre-defined HLP. Know-how and orientation in the software itself is necessary for this. Therefore, only experts can extend and troubleshoot the system in depth.
Extendability	10	The combination of KNX and LAN means that almost anything can be integrated. Additional transmission media and systems can also be implemented and expanded.
Manufacturer Dependency	7	There are hardly any dependencies on specific manufacturers. The simple frameworks used in the logic must remain supported or be replaced. The most important components that which can limit the system are the gateways such as the KNX IP BAOS gateway.
Development Effort	6	The development effort is high, as the entire functionality must be implemented. Both the inter-connection of the systems and the calculation and execution of the (automated) actions must be implemented.
Disposal	7	By mixing different systems, it may be possible to reuse or recycle parts of the infrastructure. Depending on the installation, very few cables need to be installed and disposed of again.

Table 3.10: Communication Model 1 Rating

Communication Model 2

Parameter	Rating	Description
Cost	3	A server or microcomputer ($$) and the BAOS gateway must be purchased as the basic infrastructure ($$). Based on this more expensive KNX components have to be used ($$$).
Interoperability & Functions	4	Only KNX based components or with a KNX gateway converted systems are usable. Because of the HLP, almost any functionality can be implemented in the logic.
Installation Effort	3	The set up effort for the logic is low. By using KNX as protocol, cables must be installed throughout the building or the more expensive and less reliable wireless alternative KNF RF must be used. However, as each building is unique, the basic configuration effort is high. The group address of KNX must be assigned to the BAOS as a data point in the ETS with medium effort.
Reliability	8	The industry standard KNX is used and known as very reliable. As a self-developed program, the logic can be seen as a reliable component.
Security	8	The KNX network itself is known to be secure. The logic can be completely disconnected from the internet or protected from external logical access by means of authentication options.
Component Maintainability	8	Trained KNX personnel are required to maintain the system. However, this is easy for them and can be done by all KNX professionals. Occasionally it can be done even by KNX interested people, with an ETS.
Logic Maintainability	8	The software is very complex, as different ecosystems have to be combined. This can only be done through the pre-defined HLP. Know-how and orientation in the software itself is necessary for this. Therefore only experts can extend and troubleshoot the system in depth. However, due to the limitation to KNX, the logic only needs to be adapted if the scope of functions needs to be extended. Any KNX component can be added without further effort.

Parameter	Rating	Description
Extendability	5	The model is reduced to KNX components. However, there is a very large catalog of components and the possibility to convert other systems by using KNX gateways.
Manufacturer Dependency	2	The simple frameworks used in the logic must remain supported or be replaced. The most important components that can limit the system are the gateways such as the KNX IP BAOSs gateway. If KNX as an entire system is no longer developed further or some module manufacturers stop their thesis, the system is hardly expandable for new standards and technologies.
Development Effort	7	The development effort is high, as the calculation and execution of the (automated) actions must be implemented.
Disposal	5	Due to the use of KNX, it can be assumed that little can be reused during disposal. The entire KNX cables must also be disposed of. This can also generate additional waste due to chiseling work and similar.

Table 3.11: Communication Model 2 Rating

Communication Model 3

Parameter	Rating	Description
Cost	1	Due to the exclusive use of KNX components, the KNX logic modules, which scale with increasing system size, and the (touch) displays as HMI, the model as a total is very expensive ($$$).
Interoperability & Functions	4	Only KNX based components or such with a KNX gateway converted systems are usable. Due to the logic module, complex automation is possible, but there is still a limit.
Installation Effort	3	By using KNX, cables must be installed throughout the building or the more expensive and less reliable wireless alternative KNF RF must be used. However, as each building is unique, the configuration effort is high. Especially the configuration of the logic modules can be very time-consuming.
Reliability	8	The industry standard KNX is used and known as very reliable.
Security	10	The KNX network itself is known to be secure. The whole system is usually used as local and closed environment. By adding access with phone apps and similar, sufficient authentication options have to be chosen.
Component Maintainability	10	Trained KNX personnel are required to maintain the system. However, this is easy for them and can be done by all KNX professionals. Occasionally, it can be done even by KNX interested people, with an ETS.
Logic Maintainability	10	Trained KNX personnel are required to maintain the system. However, this is easy for them and can be done by all KNX professionals. Occasionally, it can be done even by KNX interested people, with an ETS.
Extendability	5	The model is reduced to KNX components. However, there is a very large catalog of components and the possibility to convert other systems by using KNX gateways.
Manufacturer Dependency	1	If KNX as an entire system is no longer developed further or some module manufacturers stop their thesis, the system is hardly expandable for new standards and technologies.

Parameter	Rating	Description
Development Effort	9	Since KNX is configured with the ETS, no additional development work is required. Only the high configuration effort must be taken into account.
Disposal	5	Due to the use of KNX, it can be assumed that little can be reused during disposal. The entire KNX cables must also be disposed of. This can also generate additional waste due to chiseling work and similar.

Table 3.12: Communication Model 3 Rating

Communication Model 4

Parameter	Rating	Description
Cost	9	A server or microcomputer must be purchased as the basic infrastructure (\$). Based on the inexpensive TCP/IP components and the possibility for further TCP/IP systems the component costs are average (\$).
Interoperability & Functions	5	The fact that nearly any HAS device supports TCP/IP means that there are almost no limits to interoperability. Because of the HLP, almost any functionality can be implemented in the logic.
Installation Effort	8	The installation effort for the system in the building is low. Depending on the components, most TCP/IP components can be controlled by wireless. The different systems must be configured in the logic with low effort.
Reliability	3	Due to the use of many different systems, problems and errors can always occur on the system side. As a self-developed program, the logic can be seen as a reliable component.
Security	4	The TCP/IP stack has a wide range of attacks due to its scope. The system is very suitable for integrating it into the existing LAN and thus making the system accessible on the internet. Various precautions can be taken to secure the system. Compared to other models, however, this one is very vulnerable.
Component Maintainability	3	HAS devices are designed so that they can be maintained by users.
Logic Maintainability	3	Because the model is limited to the use of TCP/IP, no extensions are possible. It may only be necessary to search for errors. Only experts can troubleshoot the system in depth.
Extendability	7	There are many HAS devices on the market that cover almost all applications.
Manufacturer Dependency	9	Because the model is limited to the use of TCP/IP, no extensions are possible. It may only be necessary to search for errors. Only experts can troubleshoot the system in depth.

Parameter	Rating	Description
Development Effort	4	The development effort is high, as the entire functionality must be implemented. Both the inter-connection of the systems and the calculation and execution of the (automated) actions must be implemented.
Disposal	8	As LAN networks are widespread, it may be possible to pass on used goods or even use them for other purposes on site. The components to be disposed of are often very small and produce little waste. Many of the HAS devices are wireless and therefore do not require long communication cables.

Table 3.13: Communication Model 4 Rating

Communication Model 5

Parameter	Rating	Description
Cost	10	A Raspberry PI must be purchased as the basic infrastructure ($). The electric components connected to the serial line are very cheap ($).
Interoperability & Functions	2	Because of the HLP or the microcontroller programming language, almost any functionality can be implemented in the logic. Nevertheless, each component must be implemented individually and existing HASs cannot be used or extended.
Installation Effort	2	A separate cable must be installed throughout the building, which is not only very limited in its extended functional benefits, but must also be routed in separate cable ducts for safety reasons.
Reliability	5	The reliability is entirely dependable on the developed system and consequently highly prone to error.
Security	7	As a self-developed program, the logic can be seen as a reliable component. However, it is hardly subject to any frameworks and is therefore primarily based on in-house programming. This can be an advantage, but can also lead to overlooked errors.
Component Maintainability	3	Additions to the components are automatically accompanied by additions to the logic and require experts.
Logic Maintainability	3	Changes of the functionscope can only be done through the pre-defined HLP. Know-how and orientation in the software itself is necessary for this. Therefore only experts can extend and troubleshoot the system in depth.
Extendability	3	Each additional external component must be integrated and implemented using UART or a gateway (e.g. ESP 32) from the origin protocol to UART .
Manufacturer Dependency	4	Through the usage of LAN, there are barely any dependencies.

Parameter	Rating	Description
Development Effort	4	As most low-level electrical components are currently equipped with a serial interface, the model is independent of manufacturers. Should a change in interface standards take place at some point, the system may no longer be expandable.
Disposal	4	Due to the use of serial lines, it can be assumed that little to nothing can be reused during disposal, as these lines are not as widespread compared to a LAN network. In addition, the components used are on a low-level basis, meaning that they can only be reused with high amount of development work.

Table 3.14: Communication Model 5 Rating

Communication Model Comparison

The evaluated models are compared with each other. Table 3.15 shows the respective evaluation by model and parameter. In table 3.16, the weighting is combined with the rating and totals are calculated for the respective models:

Parameter	Weight	1	2	3	4	5
Cost	0,70	5	3	1	9	10
Interoperability & Functions	1,10	9	4	4	5	2
Installation Effort	0,40	5	3	3	8	2
Reliability	1,30	7	8	8	3	3
Security	1,40	6	8	10	4	7
Component Maintainability	1,30	8	8	10	3	3
Logic Maintainability	1,00	4	8	10	3	3
Extendability	1,50	10	5	5	7	3
Manufacturer Dependency	0,50	7	2	1	9	7
Development Effort	0,50	6	7	9	4	4
Disposal	0,30	7	5	5	8	4

Table 3.15: Communication Model Comparison

Parameter	**1**	**2**	**3**	**4**	**5**
Result	**70,90**	**61,20**	**67,70**	**50,80**	**41,80**
Cost	3,5	2,1	0,7	6,3	7,0
Interoperability & Functions	9,9	4,4	4,4	5,5	2,2
Installation Effort	2,0	1,2	1,2	3,2	0,8
Reliability	9,1	10,4	10,4	3,9	3,9
Security	8,4	11,2	14,0	5,6	9,8
Component Maintainability	10,4	10,4	13,0	3,9	3,9
Logic Maintainability	4,0	8,0	10,0	3,0	3,0
Extendability	15,0	7,5	7,5	10,5	4,5
Manufacturer Dependency	3,5	1,0	0,5	4,5	3,5
Development Effort	3,0	3,5	4,5	2,0	2,0
Disposal	2,1	1,5	1,5	2,4	1,2

Table 3.16: Weighted Communication Model Comparison

It turns out that **Communication model 1** (see section 3.4.1) is the most suitable model to solve the problem. Not only because of the average costs, but above all because of the extensive and almost infinitely expandable range of functions, as well as the possibility of expanding almost any transmission medium and protocol it prevails amongst the models.

3.6.2 HMI Models

Based on the selection of the communication model 1, only HMI model 1 (see section 3.5.1) and HMI model 2 (see section 3.5.2) can be used to control and visualize the data.

As the visualization media in HMI model 2 are more extensive and no additional hardware is required that could cause hardware dependencies, HMI model 2 is used. In addition, the know-how for common front-end frameworks is more widespread than in the layout design suite of HMI model 1. The connection of the HMI to the logic will be made via a REST interface. This type of data can be displayed without further processing. Additional graphical visualizations can be developed by front-end frameworks, but are not within the scope of this thesis. In addition, a microcontroller can also query data based on REST calls and use them to implement the HMI model 1.

CHAPTER 4

Implementation

A common way to confirm or reject a theoretical model is to develop a Proof of Concept. Therefore, based on the selected model from section 3.6, a prototype will be developed. For this purpose, an architecture is designed that combines and controls various communication protocols at a central node. This architecture is implemented using frameworks in a high-level programming language. The workflow of the software and the operation of the PoC are explained in this chapter.

4.1 Software Architecture

As already described in model fundamentals (see section 3.2), the PoC, hereinafter referred to as the prototype, consists of three components: Communication, logic and HMI. When these components are combined, the prototype for the MEMS is created. Figure 4.1 shows the connections between the components. The individual components are described in the following sections. An additional object called "PARAMS" was created in the communication component. This object is interpreted as equivalent to a connected system, but only serves as local data storage. This enables values that do not have an actual system component to be stored using standard datapoints. For example, the target room temperature can be defined MEMS-wide.

4.1.1 Communication Component

The communication component connects the various ESs, HASs and other devices (hereinafter referred to as systems) with the MEMS. Each system is implemented seperatly as "driver". Depending on the system, the driver communicates with the system components via the required transmission medium and protocol. The connected systems consist of individual components (devices). These components consist of individual values that can be read or written. The individual values (states) are referred to as datapoints.

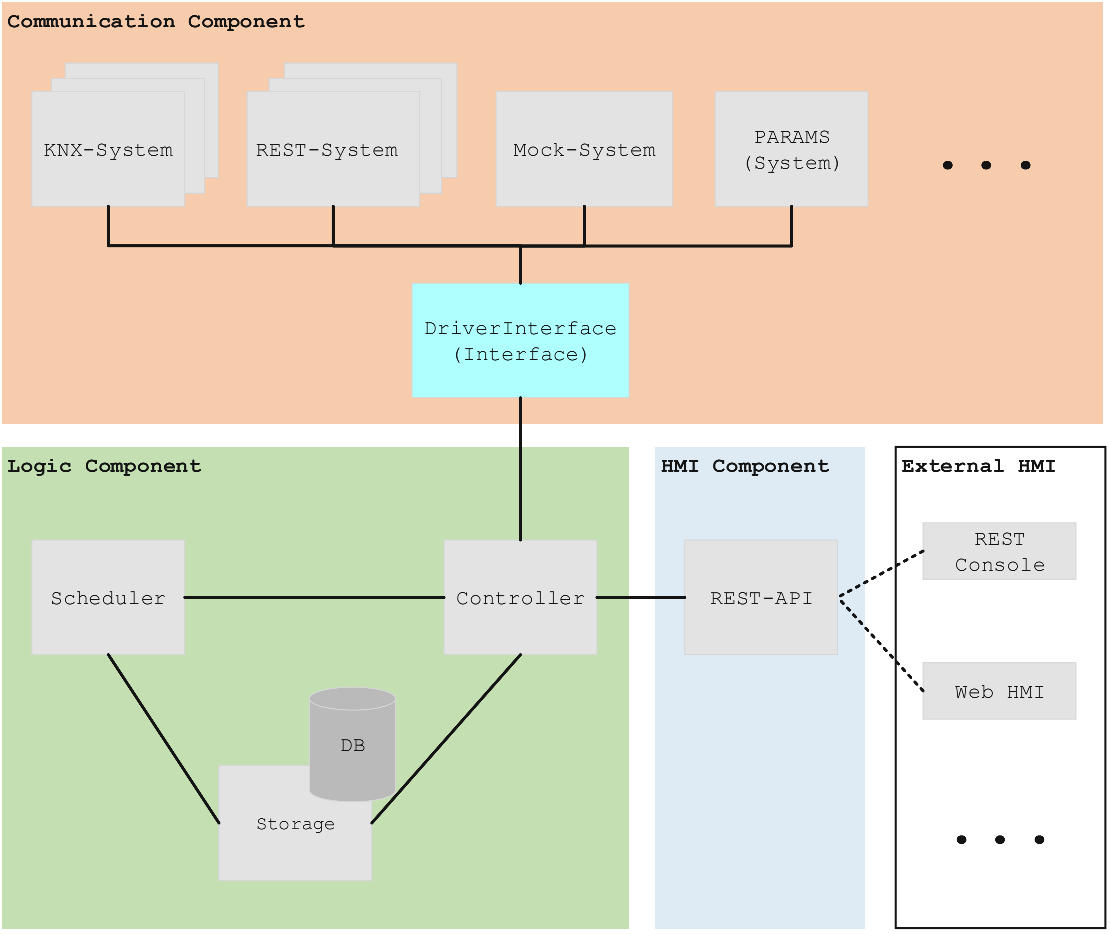


Figure 4.1: MEMS Software Architecture

In order to connect the systems to the MEMS, it is necessary to communicate with the systems (read, write), manage datapoints, detect value changes and convert (normalize) the data set into the MEMS data structure. These operations are implemented with the help of an interface (DriverInterface). The interface has various methods that the MEMS requires in order to communicate with the individual systems. The interface methods are implemented individually for each driver. This allows the MEMS to be expanded with additional systems at a later date. If several systems of the same type are connected to the MEMS, the drivers can be used several times.

4.1.2 Logic Component

The logic component is the core of the MEMS and monitors, controls and stores the states of all datapoints. It consists of three components:

- **Database.**
  The database stores all datapoints, states, automations and configurations with all necessary attributes (see section 4.5).
- **Controller.**
  The controller collects all data from the individual systems via the DriverInterface, reads and updates the database and connects the MEMS to the MEMS.
- **Scheduler.**
  The scheduler is responsible for automating the tasks. At regular intervals, it checks the database for status changes, applies the MEMS functions (see section 4.4) and initiates datapoint changes in the controller.

4.1.3 Human Machine Interface Component

The HMI component is implemented via a REST Application Programming Interfaces (API) and protected using a bearer authentication [Sof]. This allows all necessary information and states to be read from the system, states to be changed, automations to be created and managed and the MEMS to be configured. Based on REST, different forms for controlling the MEMS can subsequently be integrated. For example, a front-end can display the MEMS graphically, an AI can be connected for advanced optimization or a microcontroller with pre-layouted display can be connected. However, this is not the scope of the PoC. Currently the MEMS can be controlled via a REST console.

4.2 Technologies and Frameworks

The Proof of Concept can be developed in different ways and in different High-level Programming Language. The possibilities were narrowed down by the type of the MEMS and the architecture. For this, an object-oriented programming language is suitable. Due to the use of Java [Orab] in many courses (see section 1.6), Java was chosen as the programming language. To connect the database and providing a REST API for the HMI component Spring [Tan] is used. The decision on the type of database was also based on the courses attended. The entire MEMS stores the data in the database. Relatively speaking for a database application, the stored amount of data and database accesses are very small. For this reason and because of the structured design of the database, a relational database model is used (see section 4.5). Also based on past courses, an H2 Database Engine [noaa] was created for this purpose and connected to Spring Boot using Java Database Connectivity (JDBC) [Oraa]. The dependency handling for the remaining frameworks was organized using Apache Maven [Fou].

4.3 Datapoint Types

Each connected system has its own message structure and data types. The values of the datapoints encapsulated in the communication component must be normalized into general data types. This is important in order to be able to use, compare and change the data in further processing. Different data types must therefore be defined and the respective data formats and limits determined. After various considerations and discussions with the supervisor, the data types used in KNX were used as a basis. These were analyzed and filtered to cover the entire functional scope of the MEMS. The data types used in MEMS are shown in table 4.1. They can be subsequently extended using Java programming without affecting existing data.

#	Name	Format	Edges
0	RAW	Undefined	-
1	Boolean	bool	"true"/"false"
2	Control2Bit	{"Control":bool, "Value":bool}	-
3	ControlDimmingBlinds 4Bit	{"Control":bool, "StepCode":int}	StepCode: [0; 7]
4	CharacterSet1Byte	Character	Length: 1 character
5	UnsignedValue1Byte	Number	[0; 255]
6	SignedValue1Byte	Number	[-128; 127]
7	UnsignedValue2Byte	Number	[0; 65535]
8	SignedValue2Byte	Number	[-32768; 32767]
9	Float2Byte	Number	[-671088.64; 670433.28]
10	Time3Byte	hh:mm:ss	-
11	Date3Byte	YYYY-MM-DD	-
12	UnsignedValue4Byte	Number	[0; 4294967295]
13	SignedValue4Byte	Number	[-2147483648; 2147483647]
14	Float4Byte	Number	[-3.4028235*10^{38}; 3.4028235*10^{38}]
15	-	-	-
16	String14Byte	Text	Max. length: 14 character
17	-	-	-
18	SceneCtrl1Byte	{"Control":bool, "Scene":int}	Scene: [0; 63]

Table 4.1: MEMS Datapoint Types

4.4 Functions

Simply controlling the datapoints of various system does not make the MEMS an EMS. Compares, dependencies and processes should be automated in order to enable the desired benefits without regular intervention. A distinction must be made between two operations: Compare and combine. An extension of the operations in the MEMS is possible through Java programming without influencing existing data.

4.4.1 Compare Operators

The compare operations compare values of the same data type with each other and return a Boolean result. Based on this, a decision can be made as to whether and which actions should be executed. Based on the data types, one to three values can be compared with each other.

- **One Data.**
  A data, i.e. a manually defined value, can be compared under certain circumstances. These are usually comparisons with system values of the MEMS like date or time. An overview of which data types can be compared with which operations can be found in table B.1.
- **One Datapoint.**
  Similar to the comparison above, datapoints can also be compared with system values of the MEMS. This is very important for synchronizing the systems with the MEMS. In addition, state changes of the datapoints can be checked. An overview of which data types can be compared with which operations can be found in table B.2.
- **Two Datapoints.**
  The values of two datapoints can be compared using different operations. An overview of which data types can be compared with which operations can be found in table B.3.
- **One Datapoint and One Data.**
  The value of a datapoint can be compared with a data, i.e. a manually defined value, using various operations. An overview of which data types can be compared with which operations can be found in table B.4.
- **Two Datapoints and One Data.**
  The data can be used to define an offset in addition to the comparison of two datapoints. This offset is added to the first datapoint value. If a negative offset is set, it is subtracted from the first datapoint. An overview of which data types can be compared with which operations can be found in table B.5.

4.4.2 Combine Operators

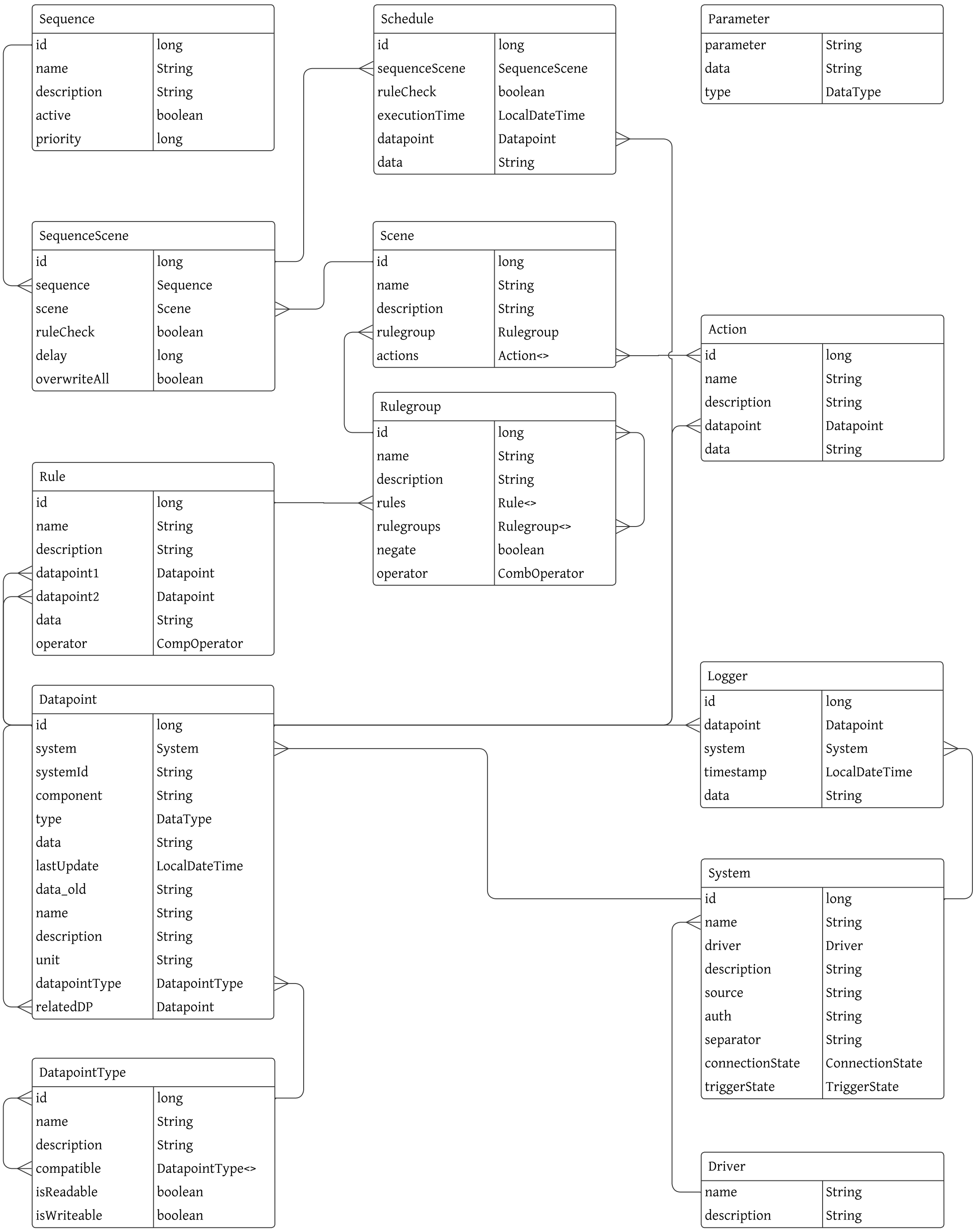
To enable more complex sequences, various compare operations can be combined with each other. The individual compare operations provide Boolean results, which can be combined using combinational logic. Furthermore, nesting is possible. An overview of the implemented operations is shown in table 4.2.

Name	**Combinatoric**
AND	$A \wedge B$
OR	$A \vee B$
XOR	$A \oplus B$
NAND	$\overline{A \wedge B}$
NOR	$\overline{A \vee B}$
NOT	$\overline{A}$

Table 4.2: Combine Operators

4.5 Database and Objects

The database serves as the data storage for the entire MEMS. Operational data of the systems, automations and a history of changes to the individual systems and warnings and errors of the MEMS are stored in the database. In addition, configurations and settings of the MEMS itself are stored in the database and make the MEMS persistent. Different types of storage types could have been used for the MEMS. Instead of a database, data structures of the HLP itself, outsourced files (e.g. XML) or the RAM of the host could have been used. However, due to the requirements and the resulting complexity, it was important to be able to create relations in the data structure. In addition, the data persistence is practical and the access times of the database are sufficient for this application. The relational database H2 Database Engine is used for this. All data used in the MEMS is stored in the database and is therefore retained after a restart. The outdated values of the datapoints are updated when the connection is established. The database may only change data using the methods defined in the MEMS. If data is changed in the database using Data Manipulation Language (DML), inconsistencies may occur. The entity relationship diagram shown in figure 4.2 shows the relationships between the database tables explained in detail in section 4.5.2. When planning the database structure, some separations must be made due to the expandability, flexibility and complexity. The data must be stored in a structure so that it can store and fulfill the necessary functions.



Sequence	
id	long
name	String
description	String
active	boolean
priority	long

Schedule	
id	long
sequenceScene	SequenceScene
ruleCheck	boolean
executionTime	LocalDateTime
datapoint	Datapoint
data	String

Parameter	
parameter	String
data	String
type	DataType

SequenceScene	
id	long
sequence	Sequence
scene	Scene
ruleCheck	boolean
delay	long
overwriteAll	boolean

Scene	
id	long
name	String
description	String
rulegroup	Rulegroup
actions	Action<>

Action	
id	long
name	String
description	String
datapoint	Datapoint
data	String

Rulegroup	
id	long
name	String
description	String
rules	Rule<>
rulegroups	Rulegroup<>
negate	boolean
operator	CombOperator

Rule	
id	long
name	String
description	String
datapoint1	Datapoint
datapoint2	Datapoint
data	String
operator	CompOperator

Datapoint	
id	long
system	System
systemId	String
component	String
type	DataType
data	String
lastUpdate	LocalDateTime
data_old	String
name	String
description	String
unit	String
datapointType	DatapointType
relatedDP	Datapoint

Logger	
id	long
datapoint	Datapoint
system	System
timestamp	LocalDateTime
data	String

System	
id	long
name	String
driver	Driver
description	String
source	String
auth	String
separator	String
connectionState	ConnectionState
triggerState	TriggerState

DatapointType	
id	long
name	String
description	String
compatible	DatapointType<>
isReadable	boolean
isWriteable	boolean

Driver	
name	String
description	String

Figure 4.2: Database Entity Relationship Diagram

The "Parameter" table was created to configure the MEMS and save settings. The "Logger" enables changes in the systems, warnings and errors to be tracked retrospectively. The "System" and "Driver" tables represent the individual HASs and ESs. Each entry in the "Driver" table reflects the implementation of a system, while the "System" table reflects each connected system including the configurations. Using „Rule", „Rulegroup", "Action" and „Scene", state changes can be recognized, evaluated ("Rule" and "Rulegroup") and subsequently executed ("Actions"). The "Rule" and "Rulegroup" tables allow comparisons to be linked and nested in order to create complex relationships. With "Sequence" and "SequenceScene", the rules and tasks linked in "Scene" can be used for different applications and delayed. The "Schedule" table is the cache for scheduled executions.

4.5.1 Objects

Different data types are used in the database to store the values. These data types are either self-created objects that are defined by another database table, Java pre-defined objects or Java enums. All objects that are not defined by the database are explained in table 4.3.

Name	Description
boolean	Java boolean object
CombOperator	Java enum: Combine operators explained in section 4.4.2.
CompOperator	Java enum: Compare operators explained in section 4.4.1.
ConnectionState	Java enum: • STOPPED: Not connected. • NOT_REACHABLE: Connection failed. • NOT_AUTHENTICATED: Authentication failed. • STARTED: Connection established. • LOCAL: Edge case for local system "PARAMS".
DataType	Datapoint types explained in section 4.3.
LocalDateTime	Java LocalDateTime object
long	Java long object.
String	Java String object.
TriggerState	Java enum: • NOT_TRIGGERED: Trigger not started. • TRIGGERED: Trigger started. • NOT_IMPLEMENTED: System does not communicate updates automatically.

Table 4.3: Object Description

4.5.2 Database Tables

Parameter

MEMS settings and properties can be defined in this table. Timeouts, configurations (e.g. see section 4.7.2) and similar can be stored and used by the MEMS.

Name	Type	Description
parameter	String	Primary Key (PK). Name of the setting or property.
data	String	Value.
type	DataType	Data type of field "data".

Table 4.4: Database Table Parameter

DatapointType

The datapoints of the different components of the systems have different properties and functions. Some are used to read out a value, some to set one. In order not to be limited to read and write only, but to implement different possibilities of rights, different datapointTypes can be created, which can be used in the MEMS extensions. A datapointType is assigned to each datapoint.

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
compatible	DatapointType<>	DatapointTypes can be used in a complementary way. For example read <-> write.
isReadable	boolean	The value of this datapoint can be read.
isWriteable	boolean	Can the value of this datapoint be written.

Table 4.5: Database Table DatapointType

Driver

A driver represents the implementation of a system.

Name	Type	Description
name	String	Primary Key. Name of entry.
description	String	Optional: Detailed description of entry.

Table 4.6: Database Table Driver

System

A system is defined by a Driver. Systems and drivers are separated so that several systems of the same type can be connected. Configurations such as addresses, ports and similar are stored in the system table.

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
driver	Driver	Driver of the system.
description	String	Optional: Detailed description of entry.
source	String	Connection address. E.g. ip-adress and port.
auth	String	variable to store the authentication parameters. Depending on the driver, the content of this entry can be processed and encrypted differently.
separator	String	Defined separator that divides the value of "auth" into different parts. This variable was introduced because a defined separator could also occur in a plain text password and cause unintentional splitting of the value.
connectionState	ConnectionState	Current status of the connection to the connected system.
triggerState	TriggerState	Does the system automatically receive changes and is this activated.

Table 4.7: Database Table System

Datapoint

The datapoint represents every single value of every component of every system.

Name	Type	Description
id	long	Primary Key.
system	System	System of the datapoint.
systemId	String	Internal id of the datapoint in its system environment.
component	String	Optional: Name/description of physical component.
type	DataType	Data type of field "data".
data	String	Current stroed data of datapoint
lastUpdate	LocalDateTime	Timestamp of last received data.
data_old	String	Data stroed before last received data.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
unit	String	Optional: Data unit. E.g. Watt (W).
datapointType	DatapointType	-
relatedDP	Datapoint	Complementary datapoint.

Table 4.8: Database Table Datapoint

Logger

Logger saves all changes from datapoints and in the systems. Errors and exceptions are also saved. Each entry must have either an assigned datapoint or an assigned system. Logger is the logging of the MEMS.

Name	Type	Description
id	long	Primary Key.
datapoint	Datapoint	Either: Regarding datapoint.
system	System	Or: Regarding system.
timestamp	LocalDateTime	Timestamp at the time of the event.
data	String	Changed data, message or exception.

Table 4.9: Database Table Logger

Rule

A rule compares one to three values using the compare operator and validates them. The result of the validation is a Boolean. Various inputs can usually be evaluated (see section 4.4.1).

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
datapoint1	Datapoint	-
datapoint2	Datapoint	-
data	String	-
operator	CompOperator	See section 4.4.1.

Table 4.10: Database Table Rule

Rulegroup

A rulegroup links one or more rules or rulegroups with a combine operator. There is only one combine operator per rulegroup. If, for example, three datapoints have to be connected with different combine operators, several rulegroups must be created. At the root, there must always be exactly one rulegroup that connects all underlying rules and rulegroups. The result of a rulegroup can be inverted (negated). Complex verifications of different datapoints can be configured by using rulegroups.

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
rules	Rule<>	Either: Set of rules.
rulegroups	Rulegroup<>	Or: Set of rulegroups to nest different rulegroups with different combine operators (see section 4.4.2).
negate	boolean	Negate the evaluated result of the rulegroup.
operator	CombOperator	See section 4.4.2.

Table 4.11: Database Table Rulegroup

Action

An action is an operation that is executed, i.e. a datapoint change. Each action can only change a single datapoint with a specific value.

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
datapoint	Datapoint	-
data	String	Value to be set.

Table 4.12: Database Table Action

Scene

A scene connects the root rulegroup with one or more actions. The scene must have at least one action or rulegroup assigned to it.

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
rulegroup	Rulegroup	Root rulegroup.
actions	Action<>	Set of actions to be executed.

Table 4.13: Database Table Scene

Sequence

A sequence is the root of each use case and represents it.

Name	Type	Description
id	long	Primary Key.
name	String	Name of entry.
description	String	Optional: Detailed description of entry.
active	boolean	Actives the sequence.
priority	long	Defines priority, if scheduled actions contradict.

Table 4.14: Database Table Sequence

SequenceScene

A sequenceScene connects the sequence with one or more scenes. Exactly one of the assigned scenes must not have a delay and must be subjected to the ruleCheck. A simple sequence is connected to one scene. The scene is assigned a rulegroup and an action and executes this if the rulegroup applies.

More complex sequences contain several scenes. These can be connected to the sequence using a delay and thus create a staircase circuit, for example. To do this, one scene must have a rulegroup (switch pressed) and an action (lights on) assigned without delay. A second scene can be executed with a time delay. This does not need to have a rulegroup assigned, only an action (lights off). For delayed scenes, ruleCheck can be used to specify whether or not the rulegroup should be evaluated. This was introduced in order to be able to use a scene several times both as an undelayed start scene and as a delayed scene. The first scene does not mandatorly need an actions assigned. For example if a light should turn of after a specific amount of time after pressing a button.

Name	Type	Description
id	long	Primary Key.
sequence	Sequence	-
scene	Scene	-
ruleCheck	boolean	The scene's rulegroup has to be evaluated. If the delay is 0, this option must be "true".
delay	long	Delay in the execution of the scene in seconds.
overwriteAll	boolean	When the scene evaluates, overwrites all scheduled scenes.

Table 4.15: Database Table SequenceScene

Schedule

All planned executions are stored in the schedule table. This table is used as a cache and the entry is deleted from the table after execution.

Name	Type	Description
id	long	Primary Key.
sequenceScene	SequenceScene	-
ruleCheck	boolean	The scene's rulegroup has to be evaluated.
executionTime	LocalDateTime	Time of planned execution.
datapoint	Datapoint	-
data	String	Value to be set.

Table 4.16: Database Table Schedule

4.6 Software Flow

By connecting the components from section 4.1 with each other, the basis for the MEMS is created. The connections and processes must now be defined and implemented to enable the automation and controllability of the system. Various situations such as starting the system, data changes and automation sequences are shown in the following sections using a flowchart. A legend can be found in figure 4.3.

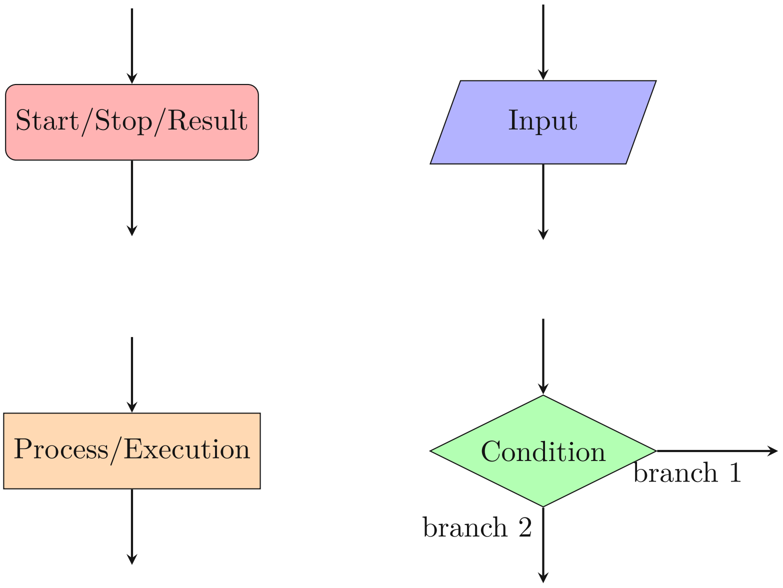


Figure 4.3: Software Flow: Legend

4.6.1 Set Up Modular Energy Management System

Starting the MEMS for a building infrastructure for the first time requires several configuration and process steps. Parameters must be set, the system variables are configured, if necessary, missing parameters of the datapoints are added (unit, description, relation, etc.), sequences are created and the scheduler is started. The steps are shown in figure 4.4. Subsequent editing of datapoints, sequences and the systems is possible at any time.

4.6.2 Start Modular Energy Management System

The MEMS can be shut down or restarted at any time without losing existing data. If it crashes for various reasons, the current database states are also saved. Figure 4.6 shows how to start an existing MEMS.

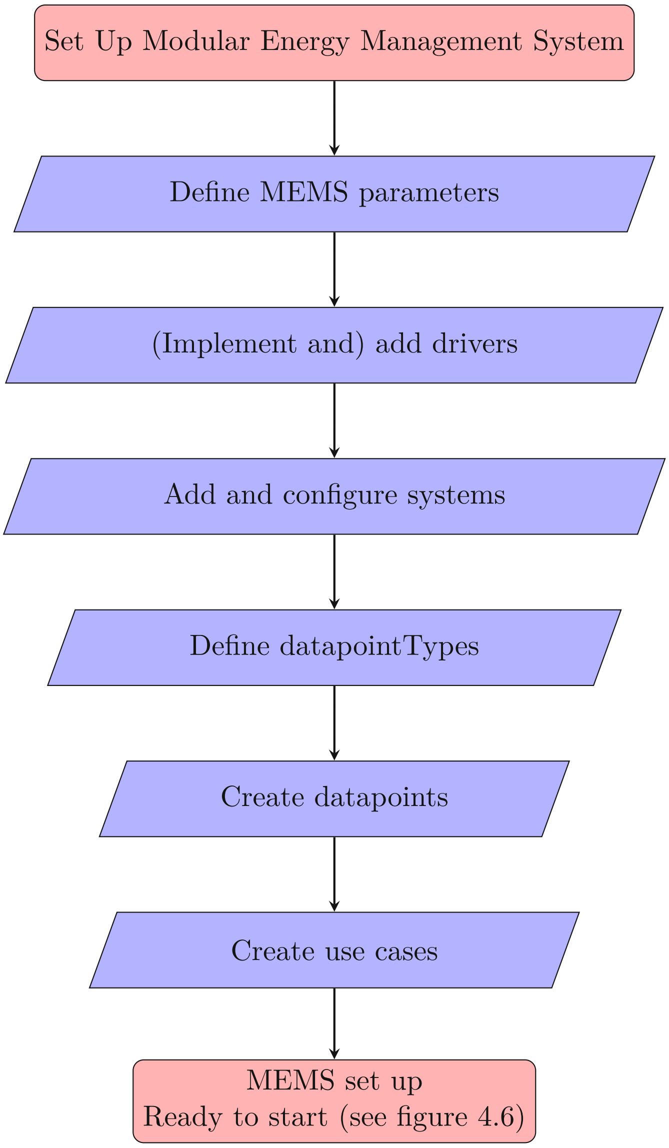


Figure 4.4: Software Flow: Set Up Modular Energy Management System

4.6.3 Read Datapoint

A basic functions of the MEMS is to read the various values of the connected systems. Each value is identified in the MEMS with an unique id. This id is assigned to a system and the internal system id (systemId) in the database and can therefore send a request to the system to receive the current value. The software flow of this operation is shown in figure 4.5.

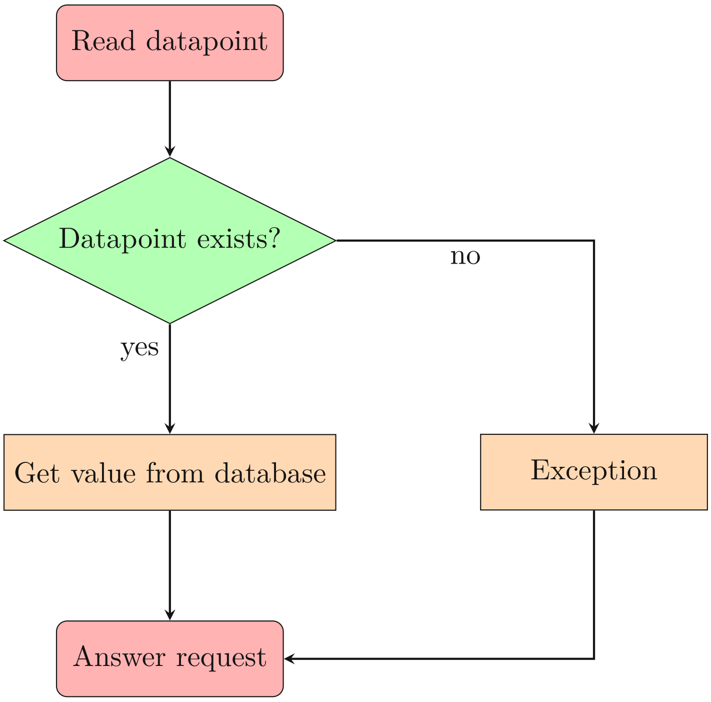


Figure 4.5: Software Flow: Read Datapoint

4.6.4 Update Datapoint

A basic functions of the MEMS is to set or update values of components of the connected systems. Each value is identified in the MEMS with an unique id. This is assigned to a system and the internal system id in the database and can therefore send an command to the system to set a new value. The software flow of this operation is shown in figure 4.7.

4.6.5 Receive Datapoint Update from System

Depending on the type of connected system, a value update is automatically communicated to the MEMS or recognized by explicit queries (further descriptions for the systems implemented in the PoC can be found in sections 5.1 and 5.2). Irrespective of the detection of the value changes, each driver of the systems informs the MEMS of a change in a standardized command sequence. This sequence can be seen in figure 4.8.

4.6.6 Scheduler

The scheduler automates the MEMS and thus the connected systems. It regularly checks the activated sequences and evaluates the associated rulegroups. Figure 4.9 shows the periodic sequence of the scheduler. Some of the operations are split into further parts:

Schedule Evaluated Scenes

The periodically called scheduler checks all activated sequences. For this purpose, the scenes of the respective sequence are read out and the scenes without delay are evaluated. If the undelayed scene is evaluated, this scene and all time-delayed scenes are added to the scheduler database table. In addition, the affecting decision "No Partial Execution of Sequences" (see section 4.7.3) will be verified. A detailed process can be seen in figure 4.10 and figure 4.11.

Delete Everything from Past

The decision discussed in Leap in Time (see section 4.7.2) can be activated or deactivated by a parameter. This parameter is checked with every scheduler execution and, if activated, deletes all outdated scheduled scenes from the database table (see figure 4.12).

Execute Scheduled Scenes

Once all scenes that are to be executed immediately or scheduled for a specific time have been added to the database table "scheduler", the scenes are executed. For this purpose, all scenes with an execution time between the last scheduler execution and the current one are processed. Depending on whether "ruleCheck" is activated, the rulegroup of the scene is evaluated. The process can be seen in figure 4.13.

4.7 Affected Decisions

4.7.1 Time-Scheduled Automation

Depending on the system, changes of datapoints are communicated to the MEMS immediately or must be checked by periodic querying. The data changes must then be processed and stored by the MEMS. The scheduler can then check whether the changed data triggers a sequence call or not. There are two different methods for this: event-based or time-scheduled checking.

With an event-based approach, the associated sequences would be checked every time a datapoint is changed. Depending on the datapoint, its value usually changes every hour, minute, second or even several times within a second. This can lead to unnecessary database accesses and also offers the additional vulnerability that the MEMS can be intentionally or unintentionally overloaded. In addition, not all systems are event-based and therefore generate unpredictable states. Other datapoints could communicate changes again shortly afterwards, forcing a constant change of state.

The time-scheduled approach checks the created sequences at regular intervals and executes actions if necessary. This creates a delay that lasts until the next scheduler check starts. However, these verifications can also be carried out every tenth of a second so that the delays are kept as short as possible.

Both approaches have advantages and disadvantages and must be selected according to the requirements of the MEMS. As stability was one of the requirements for it, the time-scheduled approach was followed for the automation of the MEMS. The periods of verification can be changed in the database table "parameter" and take effect after a restart.

4.7.2 Leap in Time

As can be seen in section 4.6.6, the scheduler stores all upcoming actions that need to be executed in the database table "scheduler". These actions are provided with execution times, which are executed if they are in the past. If the MEMS crashes, the MEMS time is changed or other unexpected circumstances occur, a decision must be made as to whether scheduled actions that are longer in the past should still be executed. For this purpose, an entry has been created in the database table "parameter" (see section 4.5.2) that defines whether these actions should be discarded or subsequently executed.

4.7.3 No Partial Execution of Sequences

If a sequence is evaluated and it turns out that at least one datapoint to be changed already has a planned execution, a decision is made. Based on the priority and the isOverwrite flag it will be decided whether the current sequence is executed and scheduled or not. As partial execution of the actions to be executed can lead to an inconsistent result, the entire sequence is not executed. If the current sequence has a higher priority and the isOverwrite flag is actived, the scheduled scenes will be overwritten with the current.

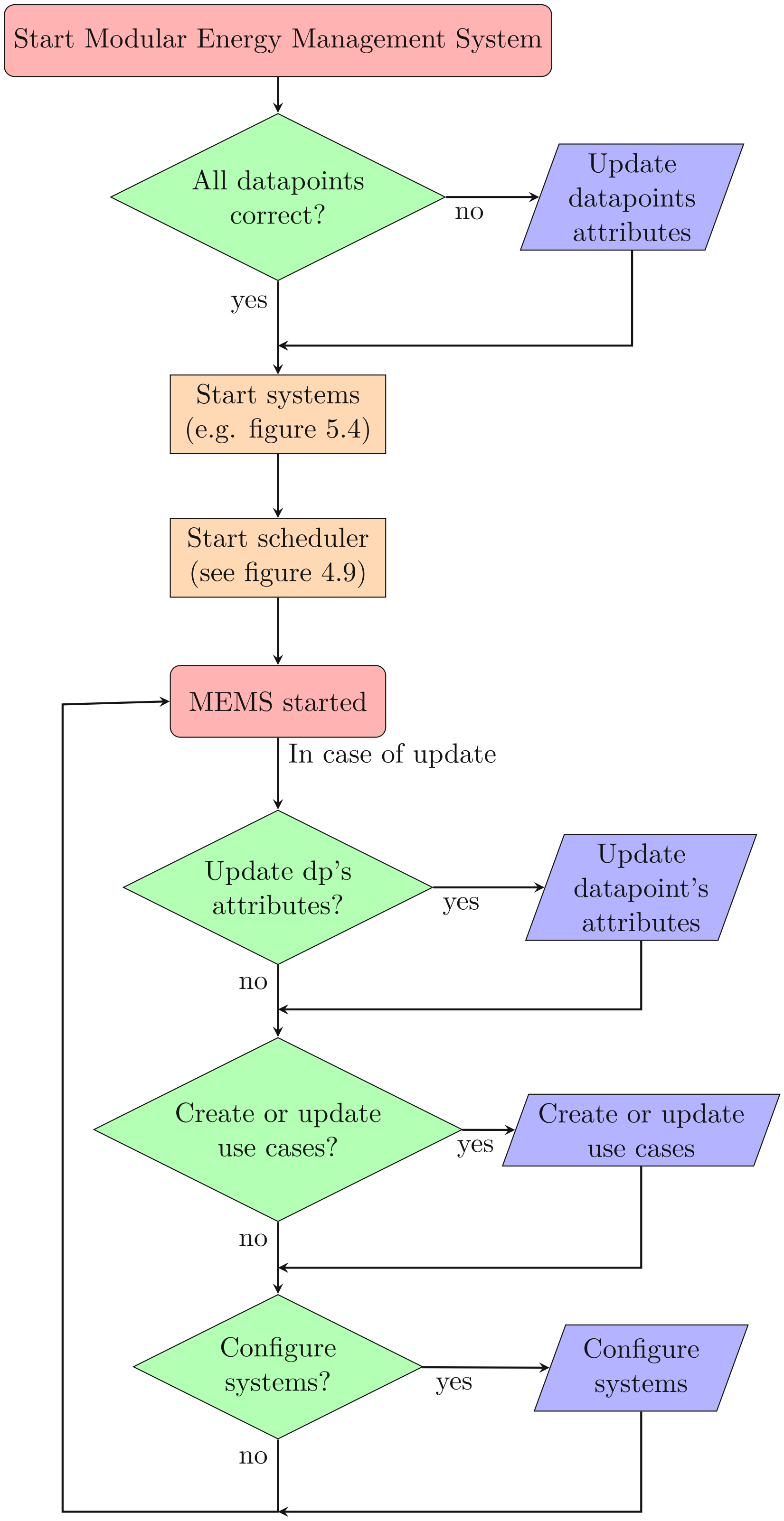


Figure 4.6: Software Flow: Start Modular Energy Management System

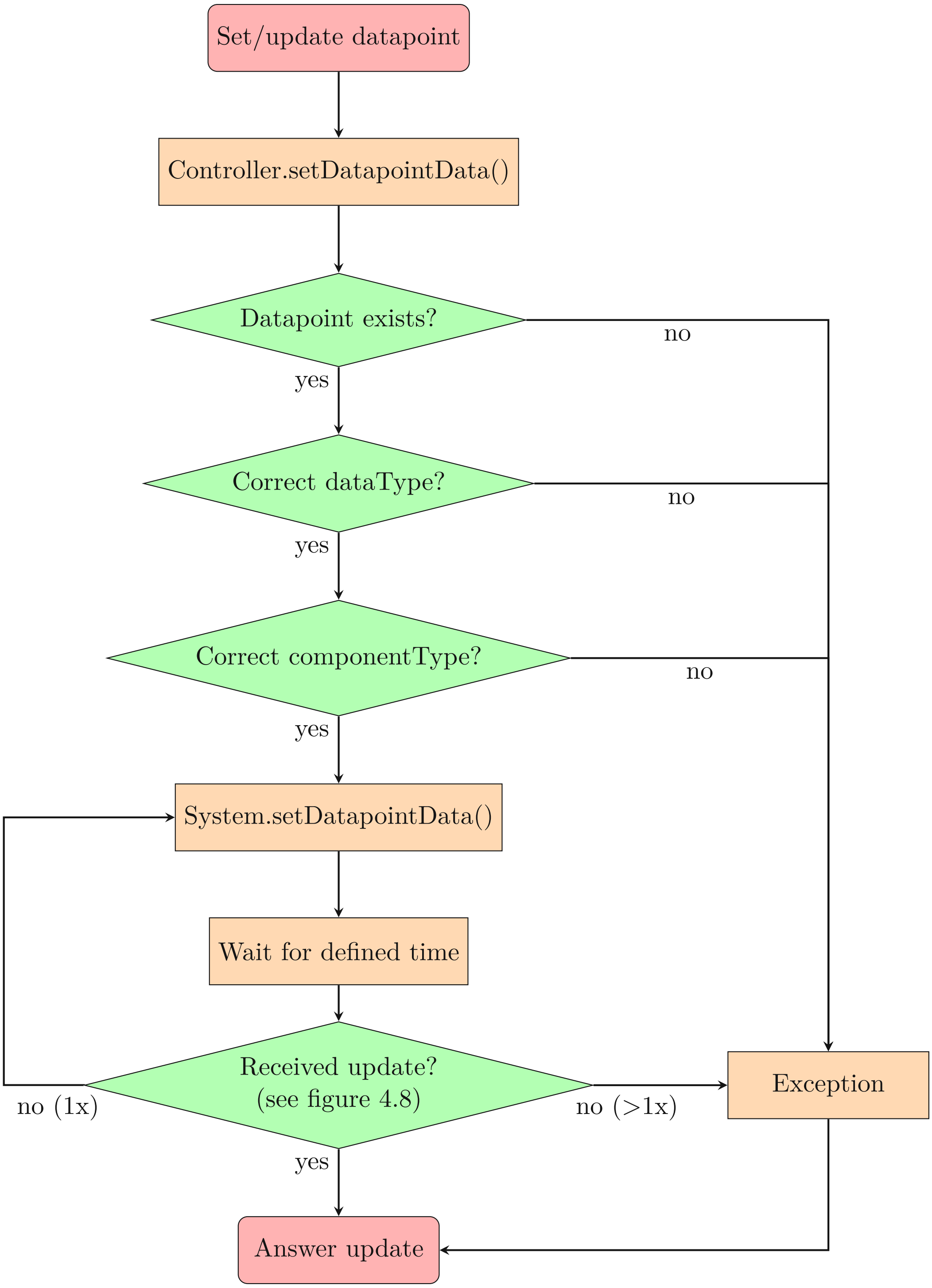


Figure 4.7: Software Flow: Update Datapoint

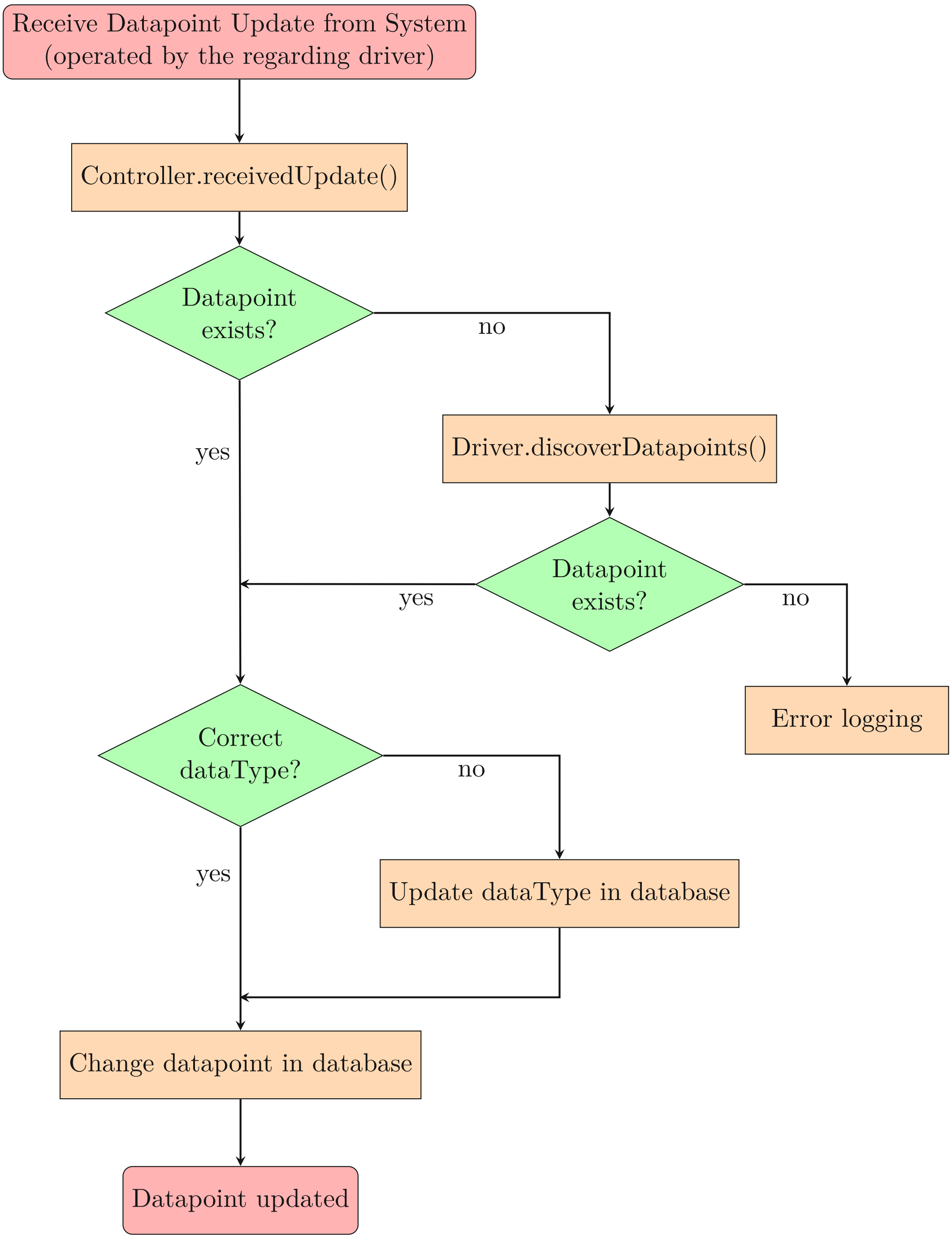


Figure 4.8: Software Flow: Receive Datapoint Update from System

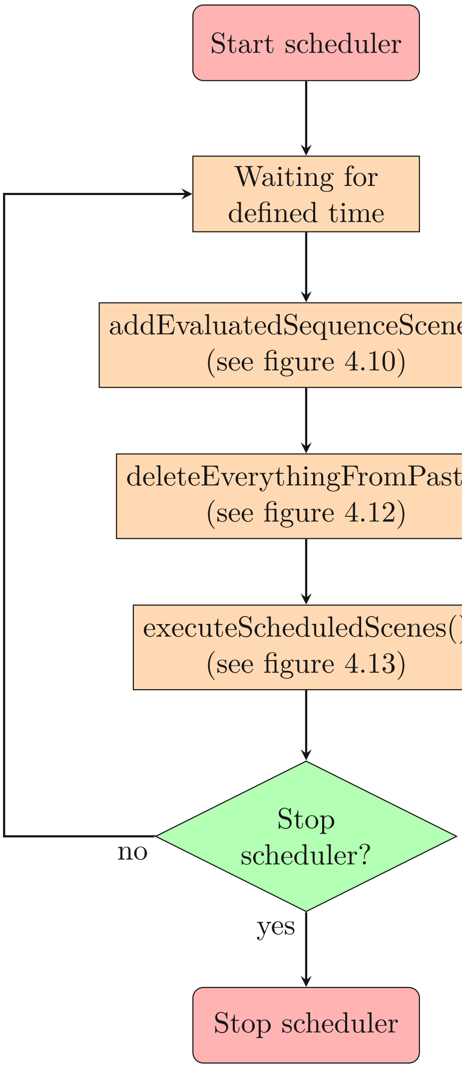


Figure 4.9: Software Flow: Scheduler

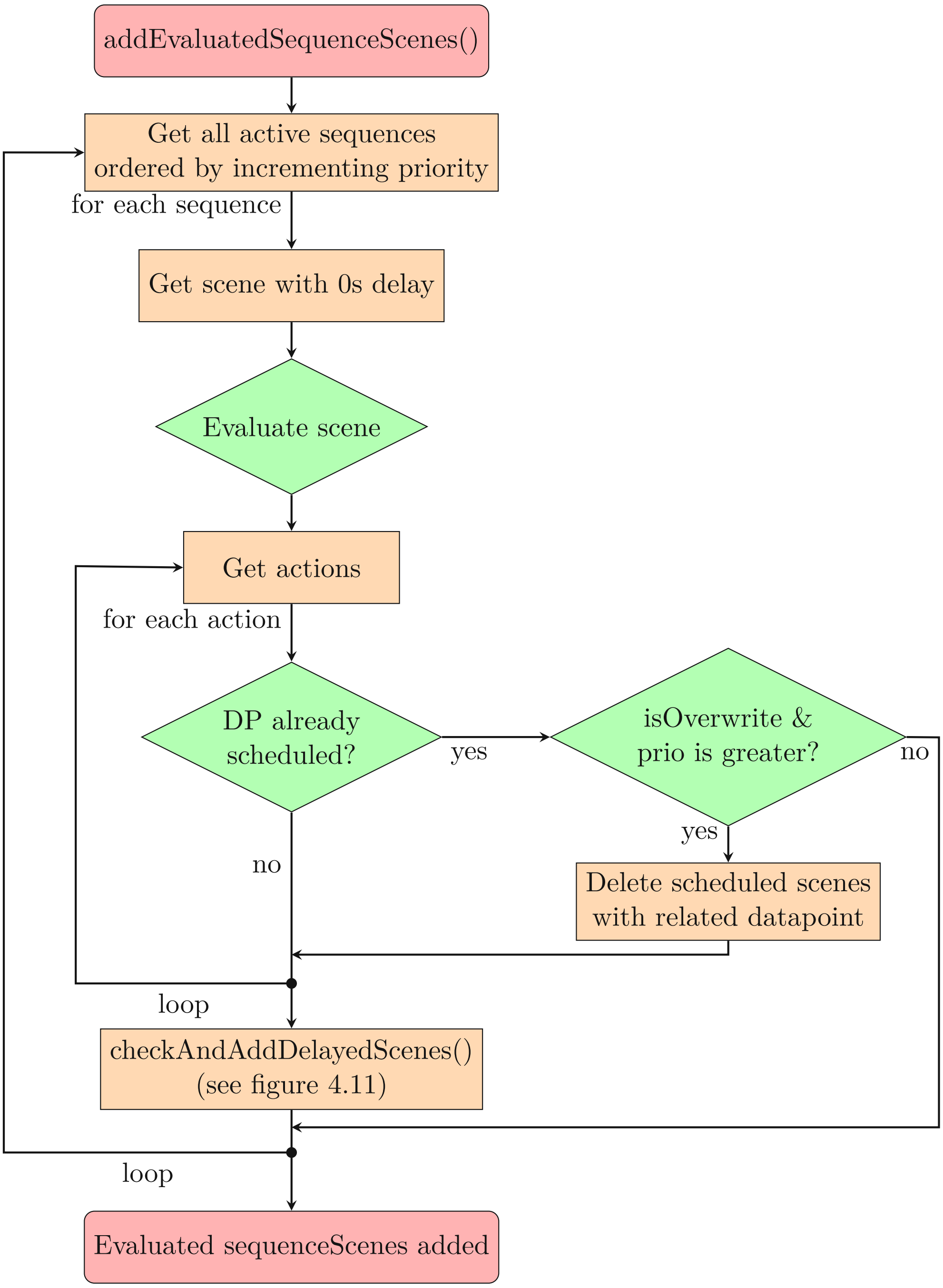


Figure 4.10: Software Flow: Add Evaluated SequenceScenes

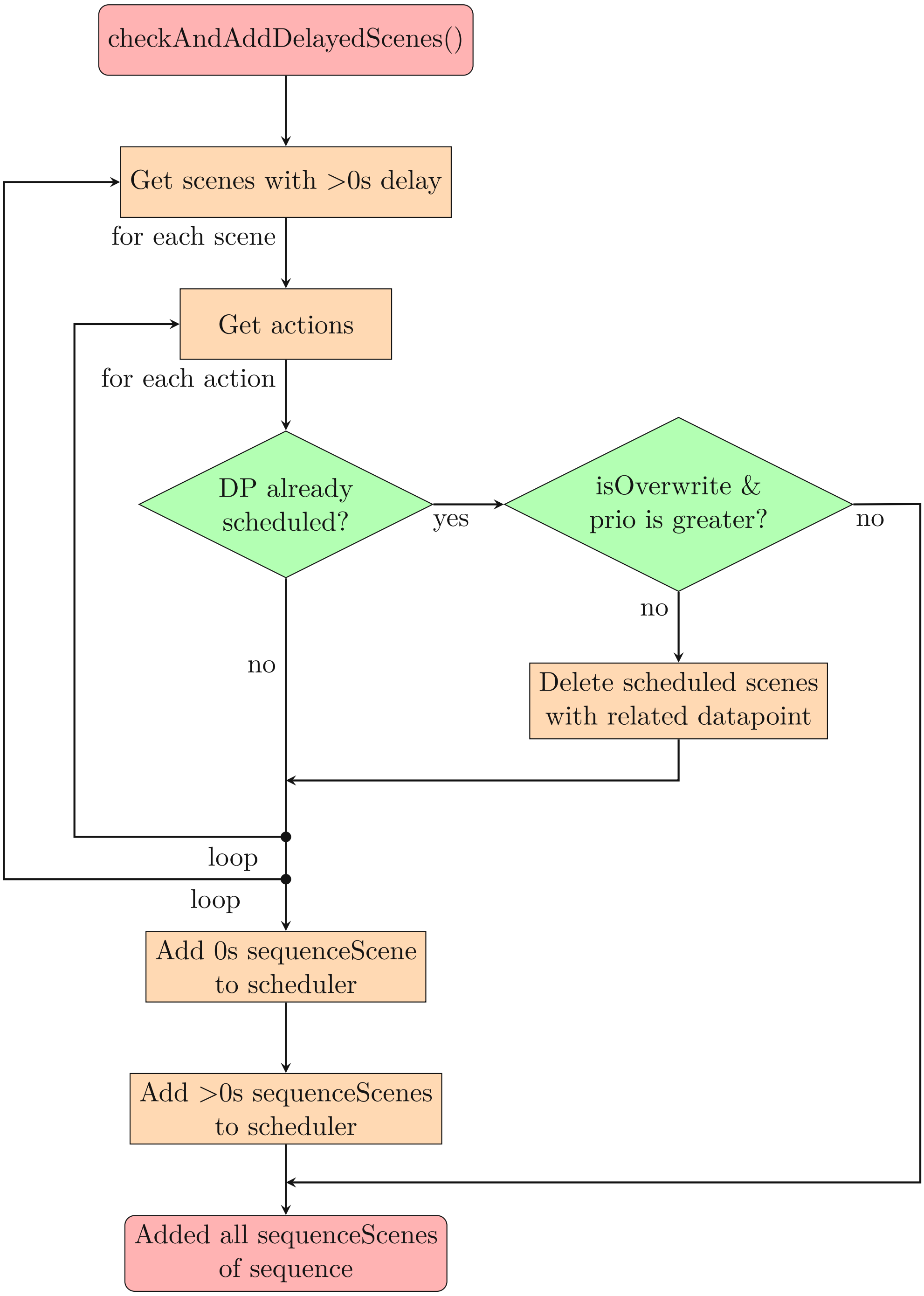


Figure 4.11: Software Flow: Check and Add Delayed Scenes

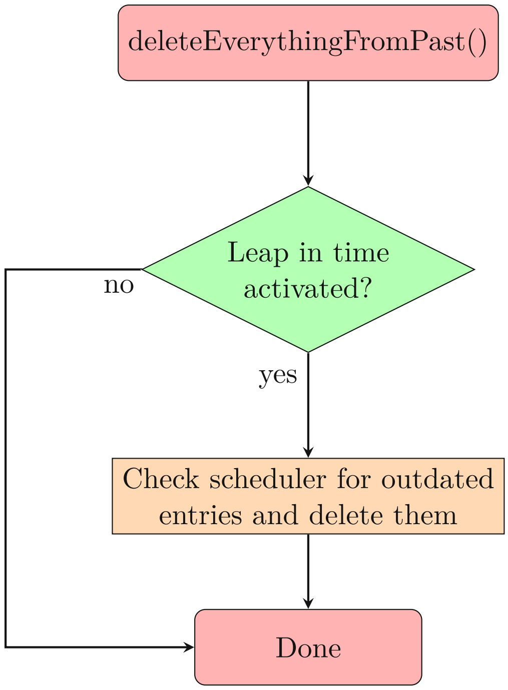


Figure 4.12: Software Flow: Delete Everything from Past

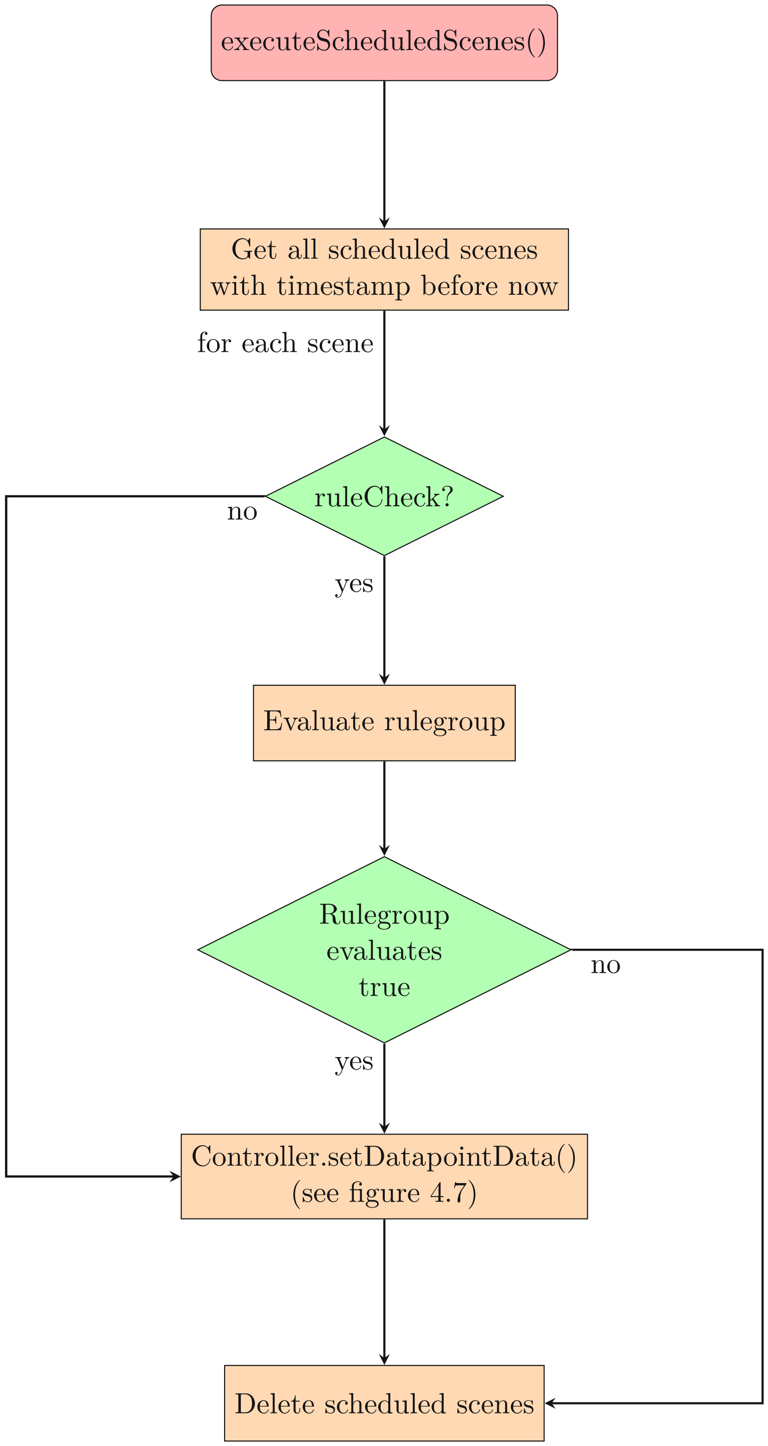


Figure 4.13: Software Flow: Execute Scheduled Scenes

CHAPTER 5

Evaluation

In order to confirm the proof of concept, it must be verified. The general functionality of the necessary requirements of the Proof of Concept are evaluated by partially employing state-of-the-art hardware. In order to check the scalability and range of functions, an environment is developed that simulates various systems. This environment is used to check the compatibility between different communication protocols. Finally, practical and realistic use cases are defined. These are verified with both the test environment and the physical test system.

5.1 Mock System

The first evaluation stage was tested using a mock system. This simulates a system that communicates with the MEMS like a HAS or ES does. Based on the REST API, a Java application was developed that implements a data structure similar to that of the MEMS datapoints. The application simulates individual component values and stores them in a separate structure. Values can be read or set from the application using REST calls. Using time-dependent sequences, changes can be made by the mock system and processed further in the MEMS. In addition, the values can be externally read and set via a REST console.

The mock can be hosted locally, in the LAN or globally. It is addressed via an ip address and port. The use of ports not only prevents conflicts in the ip network, but also allows multiple instances of the mock to be started on different ports. This simulates the use of several individual systems.

The functionality in the MEMS can be verified and the scalability for multiple systems demonstrated both through the time-based command sequences and through manual control via the REST console. The data structure of the objects stored in the mock is shown in figure 5.1. If a value in the mock is changed, this automatically changes the value of the "relatedDP" too. These two value changes are sent to the MEMS as a REST call, as a typ of datapoint change information.

Mock Datapoint	
id	long
name	String
description	String
datatype	int
value	String
relatedDP	long
component	String (devicename)
permissions	String (read/write)

Figure 5.1: Mock Datapoint Object

5.1.1 MEMS Implementation

The mock was created and implemented as a "driver" in the MEMS. Several mock instances can be connected to different ports configured in database table "systems". A flowchart of how the driver of the mock system starts in the MEMS is shown in figure 5.2.

By using the mock, the connection of many systems can be simulated. For the first final evaluation with the mock, 13 instances were created that each had between 10 and 40 datapoints with different data types and relations. Simple use cases were used to implement sequences that verify whether datapoints from different systems can be made dependent. The verification was successful and confirmed the interoperability and functionality of the MEMS.

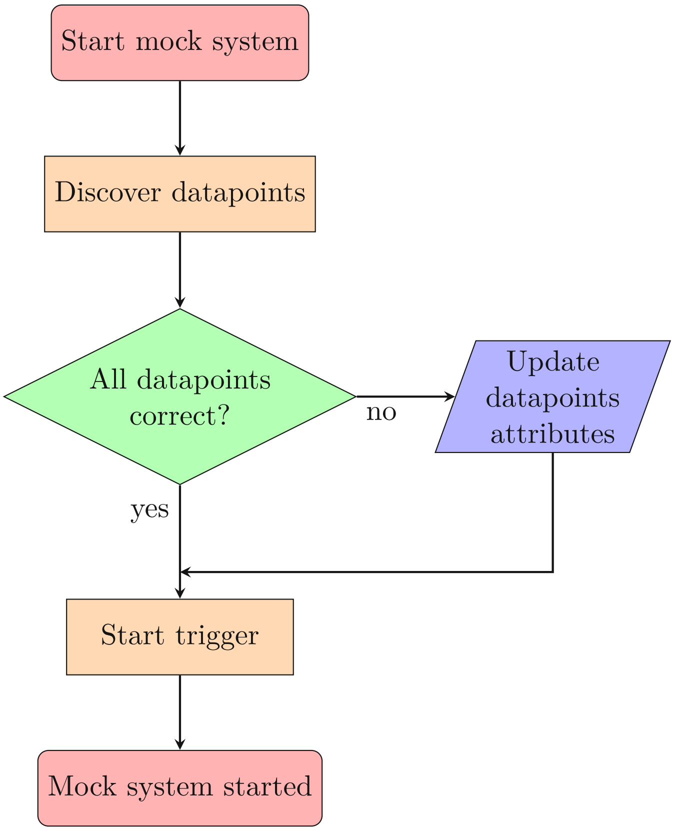


Figure 5.2: Software Flow: Start Mock System

5.2 KNX System

A KNX driver is implemented in order to integrate interoperability, expandability and a system in the MEMS that is independent of own programming. This enables a connection to be established with a KNX IP gateway (Weinzierl KNX IP BAOS 777) using a TCP/IP connection and datapoints (in KNX called group addresses) to be read and set.

For this purpose, a small KNX network (see figure 5.3) consisting of a power supply (1: MDT STV-0160.01), the KNX IP gateway (further called BAOS) (2: Weinzierl KNX IP BAOS 777) and a switch with several input and output options (3: MDT BE-GT2TW.02) was created. The power supply only serves as a simple energy source.

The Bus Access and Object Server offers various options for controlling the KNX network. Before it can access the KNX group addresses, however, datapoints must be created in the BAOS. These can then be controlled by the BAOS. A BAOS datapoint is nothing more than an entry assigned to a group address. A group address can be controlled by teh BAOS by referencing the BAOS datapoint in the ETS.

Unlike the group addresses, the BAOS datapoints are identified by a numerical id. This id can be used to control the BAOS datapoints using a REST API, websocket [Wik24c], binary protocol [Gmbb] or via the BAOS web application. The MEMS can use the REST API or the websocket for this. The advantage of the websocket is that a connection is maintained and changes in the system are communicated immediately.

Several group addresses could be controlled and visualized by the switch. The switch has 6 buttons, 6 color LEDs and a display with various layout options. The buttons and LEDs were used to control and visualize different states in the KNX system. These were then forwarded to the MEMS by the BAOS and processed further. In order not to verify the unlimited use of the systems functions and automations itself, automations were implemented both in the KNX network itself and by the MEMS. The display was used to show numerical values or character strings.

By integrating the KNX system, the MEMS could be connected to a concrete HAS. Interoperability and expandability were confirmed once again. In addition, the KNX system is used in the use cases (see section 5.3) as visualization and control.

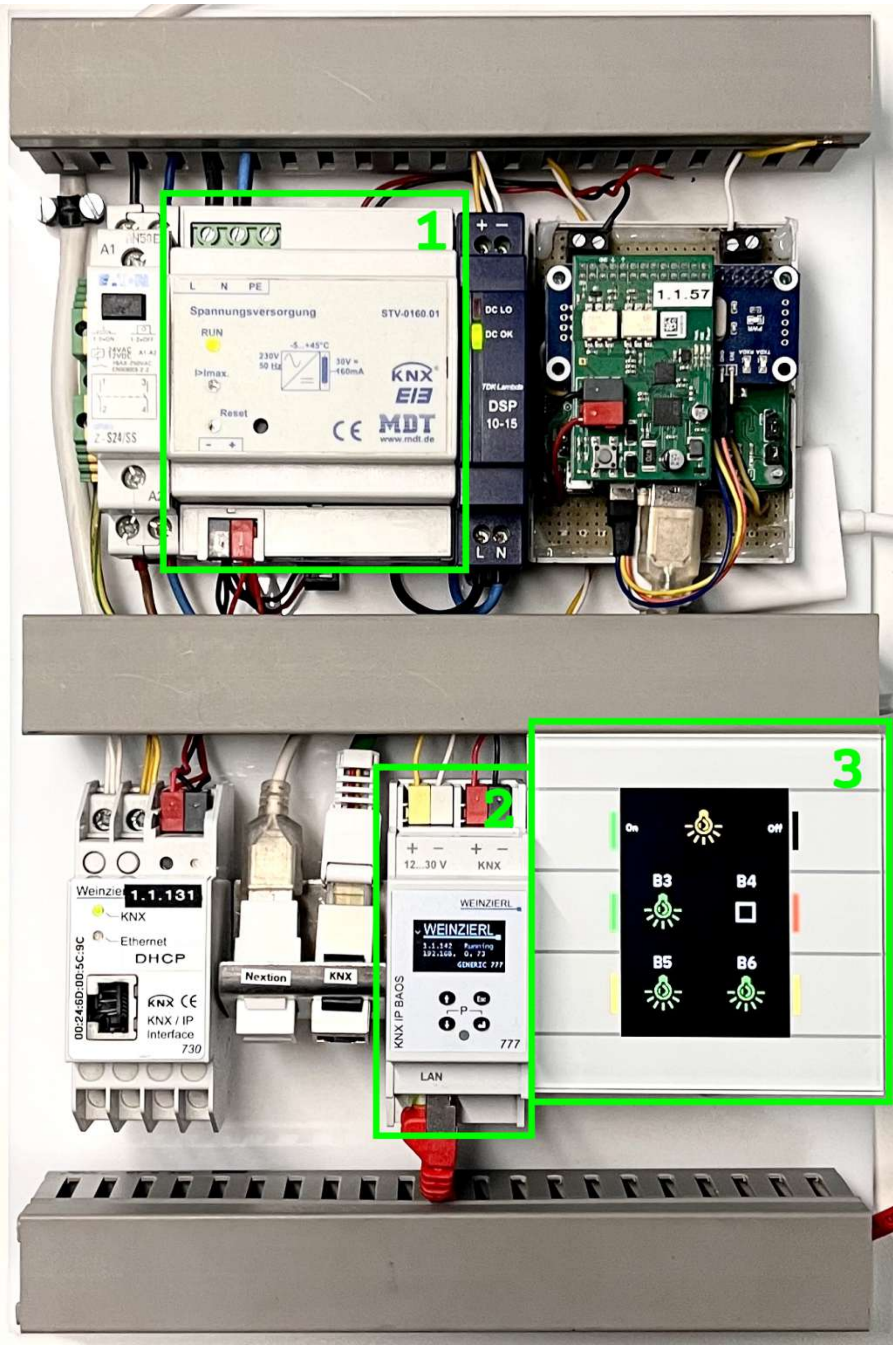


Figure 5.3: KNX System Hardware Kit

5.2.1 MEMS Implementation

The mock was created and implemented as a "driver" and configured as a "system" in the MEMS. As the KNX system is a real HAS, more functionalities could be implemented. For example, a keepalive checks the connection to the system at regular intervals. The MEMS communicates with the BAOS primarily via REST. In addition, a websocket connection is established to recognize updates in the system. Every time there is an update in the system, the BAOS sends a message to the websocket. However, as the BAOS websocket only has a limited set of functions, it is only used to determine that an update has occurred and which datapoint is involved. The (updated) value of the datapoint is then queried by a REST call. A flowchart of how the driver of the KNX system starts in the MEMS is shown in figure 5.4.

5.3 Use Cases

Various use cases were defined, not only to test the MEMS in the form of a meaningful usage and to use as many features as possible, but also to cover various reasons for a HAS. Depending on the needs, the reasoning could be a contribution to self-sufficiency of a building, ensuring or guaranteeing problem-free use or monetary savings.

The use cases employ datapoints from the mock-system (see section 5.1) and therefore test the entire MEMS. In some of the use cases, LEDs are consulted for status visualization. As an extended test and for easier manual verification, all suitable use cases were also tested and visualized with LEDs of the KNX test system (see section 5.2).

5.3.1 General Workflow

Several sequences are required for the following and generally most use cases. They are needed both to revise the initiated state under certain conditions in simple cases and also to be able to use different parameters that perform different actions.

Three use cases were developed, defined and added to the MEMS based on the basic functionalities defined with the supervisor.

Used Workflow

The basic idea, functionalities and fundamental reactions were discussed with the supervisor. Based on this, inputs and outputs were selected from the existing mock datapoints. Considerations were made as to which states could be possible and how they could be achieved. These were then added to the MEMS using sequences (see section 4.5.2). A Finite-State Machine (FSM) was only used for visualization. In some cases, circular relationships were discovered and resolved by creating the FSM. In a subsequent analysis, it was determined that the approach was not the best and the following approach would be an improvement:

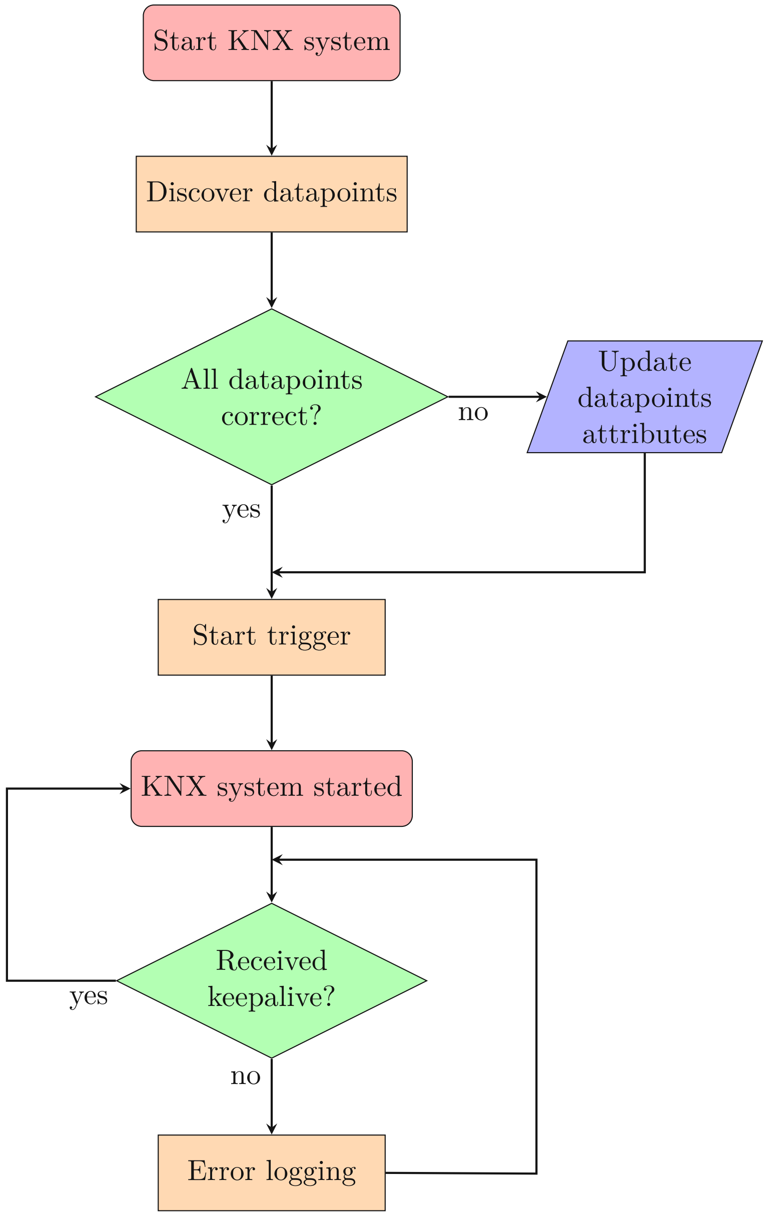


Figure 5.4: Software Flow: Start KNX System

1. Define input and output parameter.
2. Define states of the use case with specific values for the outputs.
3. Define transition conditions for each state.
4. Verify (and visualize) using a Finite-State Machine and check for circular relationships and inconsistencies.

5. Resolve circular relationships and inconsistencies by means of error correction or sequence priority.
6. Add each transition to the MEMS using sequences.

5.3.2 Use Case 1 - General Power Line Protection

Purpose

A modern (private) building in Austria is usually connected to the public power grid with a one- or three-phase power connection from an ESC. As an option, other feed-in sources such as PVS, energy storage systems, electric cars, etc. can also provide electricity. This results in a total amount of electricity available per phase that consumers can consume. However, this consumption is also limited by the load capacity of the power line and the associated protection to ensure that it is not overloaded.

In a normal household, the limitation by available electricity is normally not a restriction, as the connected load is reliably classified by the electric supply company and electricity grid provider when the house connection is installed (note: in the following years, this classification could increasingly prove to be too weak if e-mobility with private charging stations and the use of heat pumps continues to advance). It is primarily the installed (in-house) power lines with their fuses that form the limit of the available electricity. If this limit is exceeded for a certain period of time (depending on the characterization of the fuse), the fuse switches off the overloaded line and no more current flows. This can affect not only the triggering device but also all devices on this phase or line, including the entire residential unit. In some cases, appliances not only lose power as a result, but could also be damaged in the long term by voltage peaks or faulty switch-off procedures. To avoid this, people must be warned or other (automatic) precautions must be taken when the maximum transportable current is approached.

In addition to failure and device protection, there is also the possibility of not wanting to use any or as little (expensive) electricity as possible from the ESC by using alternative forms of energy. To implement such a scenario, a calculation mechanism is needed that compares the available alternatively generated electricity with the electricity currently consumed, similar to the maximum transportable electricity, and warns people in advance or sets precautions. Mathematically and logically, this is the same comparison as mentioned above. This scenario partially covers the category of self-sufficiency, as well as money savings.

A lower connected load can also be required by the ESC and electricity grid provider. This is also associated with lower (running) grid costs. Precautions can also be taken here to protect the private electricity network, which may have a too low connected load from the electric utility company and too much consumption.

Last but not least, this use case can be applied in a conservative private electricity network if, as briefly mentioned at the beginning, one or more electric vehicles charge their batteries in this network. This is a problem that is currently the subject of much discussion [SJL15, EMASA17] and, with the help of this use case, at least protects internal services from overload or may be able to react better in the future. If the electricity grid provider has to provide a variable amount of electricity in the future, this use case can be implemented to regularly check that the upper limit is not reached and precautions can be taken if necessary.

Inputs and Outputs

For this use case, it is assumed that a certain 3-phase power line (L1-state/L2-state/L3-state) is measured and checked regarding its power consumption. Whether this measurement is made at a line point that measures the entire building or just one area does not matter in the use case. The measurement can be done by different devices like an KNX-Ampere-meter or some other HAS devices. Only the limits and subsequent precautions must be selected accordingly. A precaution that switches off a component that is not connected to the same line as the measurement has no effect.

There is an indicator (Line-Protection-state) that shows whether the lines can currently be loaded (ok), caution is required (warning), or an immediate precaution should be taken (alarm). This indicator is stored as numerical values as an internal datapoint. In the KNX visualization mode, a LED lights up in different colors (ok: green, warning: yellow, alarm: red).

The actual use of the use case and the precautions taken can be determined on the basis of the internal datapoint value by adding further sequences.

Figure 5.5 visualizes which values are displayed. The measured lines are compared with limits whose values are saved as internal datapoints. These limits can be set once or variably by the user or can alternatively be replaced by another datapoint. This can, for example, be a datapoint that defines the current power guaranteed by the energy supplier.

To avoid oscillating behavior, the precautions taken subsequently should be well considered and provided with hystereses. For example, a consumer whose consumption is reduced should not be switched on again immediately as soon as the state is ok, but should be reactivated a little later by means of a time delay so that the same state is not reached again. Different hystereses can also be used for the transitions between the states, so that, for example, there must be a difference in power of more than 100 watts.

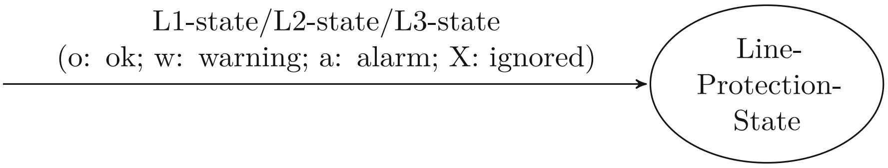


Figure 5.5: Use Case 1: FSM Legend

State Machine

Based on the individual phases measured, the use case switches between three states. The most vulnerable phase is always decisive for the current state. With small changes, each phase could also be individually monitored. To do this, the use case is reduced to one phase and connected three times as three instances to the different datapoints (measurement components). In this case, care must be taken if there is a consumer in a parallel model that is connected to more than one phase. The action must be designed accordingly.

In this specific case, a single FSM represents the three phases. A transition from ok to alarm and vice versa has been explicitly built in to both prevent a delay and to avoid overlapping the precautions taken in such a short time.

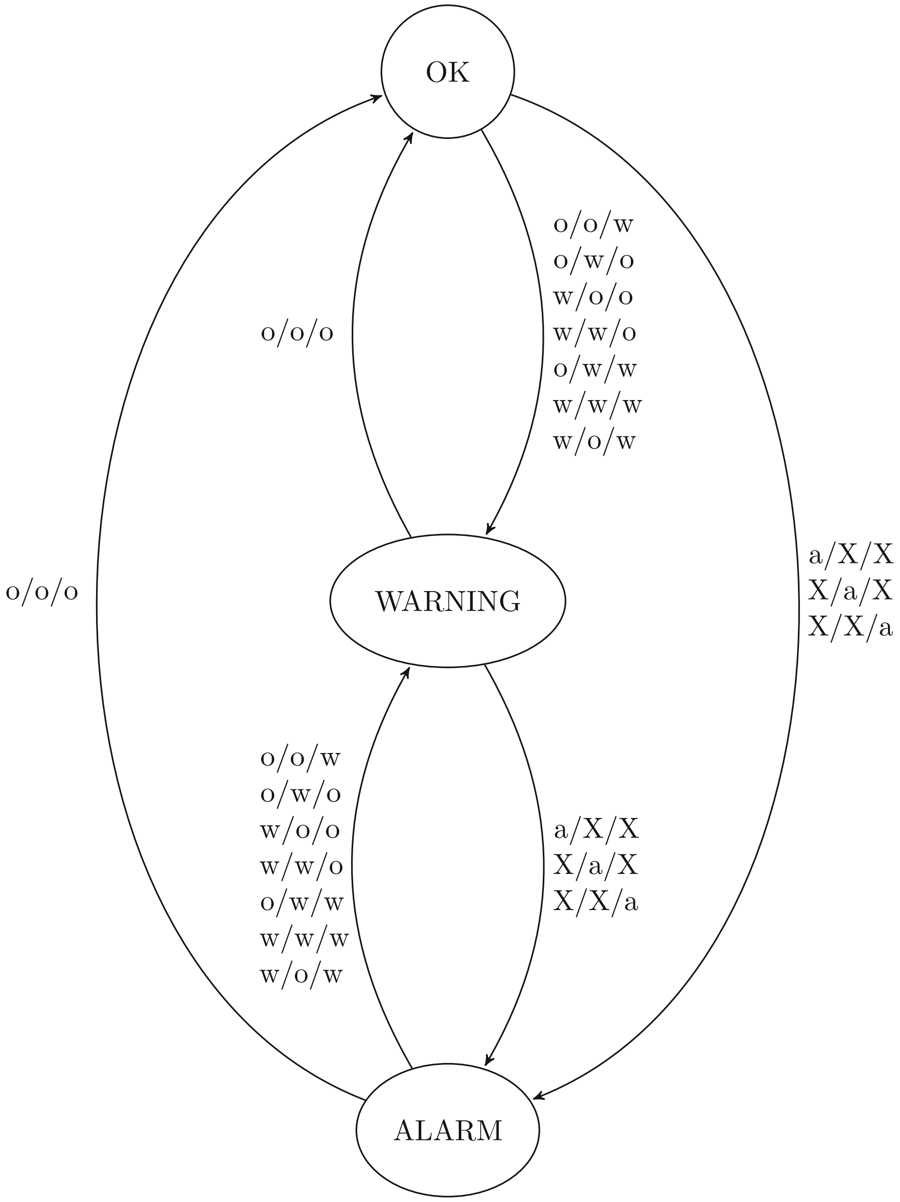


Figure 5.6: Use Case 1: Finite-state Machine

See listing A.1 for all relevant sequences created in MEMS.

5.3.3 Use Case 2 - Washingmachine and Dishwasher Usage

Purpose

Similar to use case 1, this is about electricity consumption. The aim is to prevent a washing machine from being activated at the same time as the dishwasher and also to show the user as simply as possible whether an appliance should be used, and if so, which appliance should be used next.

The benefit is to reduce consumption peaks to avoid an insufficient connected load or overloading a poorly planned or old electrical network.

The use case makes recommendations as to whether or not a device can be activated. However, these are only based on sequences that have been configured once and assume that a minimum power level is available. The overall status of the network from use case 1 is not taken into account. In future stages, the use case recommendations could not only be based on predefined sequences and time delays, but could also use continuous learning pattern recognition to make suitable decisions as to whether or not the other device can be reactivated immediately after the end of use. For more details, see the section 6.2.

The main task of the use case is to ensure that two or more household appliances are never operated at the same time. This is controlled by measurements of an electricity meter, switchable sockets, and LED indicators.

Inputs and Outputs

The use case can be used for two or more consumers with simple additions. This specific use case was designed for two appliances, specifically, a washing machine (WM) and a dishwasher (DW). Both appliances draw power from a socket that is either integrated or connected in between and has a sensor to measure the power currently being consumed (WM-Consumption/DW-Consumption) and an actuator to switch the power on and off (WM-state/DW-state). Power measurements can be used to detect whether a device is switched on or in operation (>0) or switched off (0) (the value 0 may differ slightly depending on the device and can also be defined with a limit close to 0).

In addition to the switched socket, which decides whether the device can be used (Ready/On) or not (Off), there is also an LED (WM-LED/DW-LED) (or a similar type of visualization) near the device, which indicates whether the device can currently be used (green), is being used, or another device is being prioritized (yellow) or cannot be used (red).

Based on the current status and the current power measurements, the status is maintained or changed. A change means that either the sockets are activated or deactivated and/or the visual recommendation of the devices is changed.

The LED colors have the following meaning:

- Green: The appliance has power and is ready for use
- Yellow: The device is either currently in use or is being prioritized
- Red: The device has no power and cannot be used.

This use case can be applied to the components types suggested regardless of the type of device. In this case, the selected devices are simple not-smart devices. Of course, instead of the smart sockets, one or both of the devices could also be measured and controlled directly using a REST interface or similar.

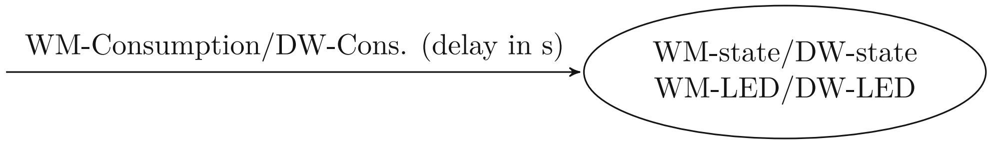


Figure 5.7: Use Case 2: FSM Legend

State Machine

The initial state is that both devices are ready for operation and are supplied with power. If one appliance' socket is turned on, the use case changes to the next state, switches the LED from green to yellow, deactivates the power supply to the other appliances, and switches their LEDs to red. As soon as the washing machine stops drawing power, it switches to the next state in which the previous device is prioritized for a certain period of time. However, the other appliances can still be used and switch to the state described above from their point of view. If no other appliance is activated, the system switches back to the start state after a certain time and all appliances are ready for operation again.

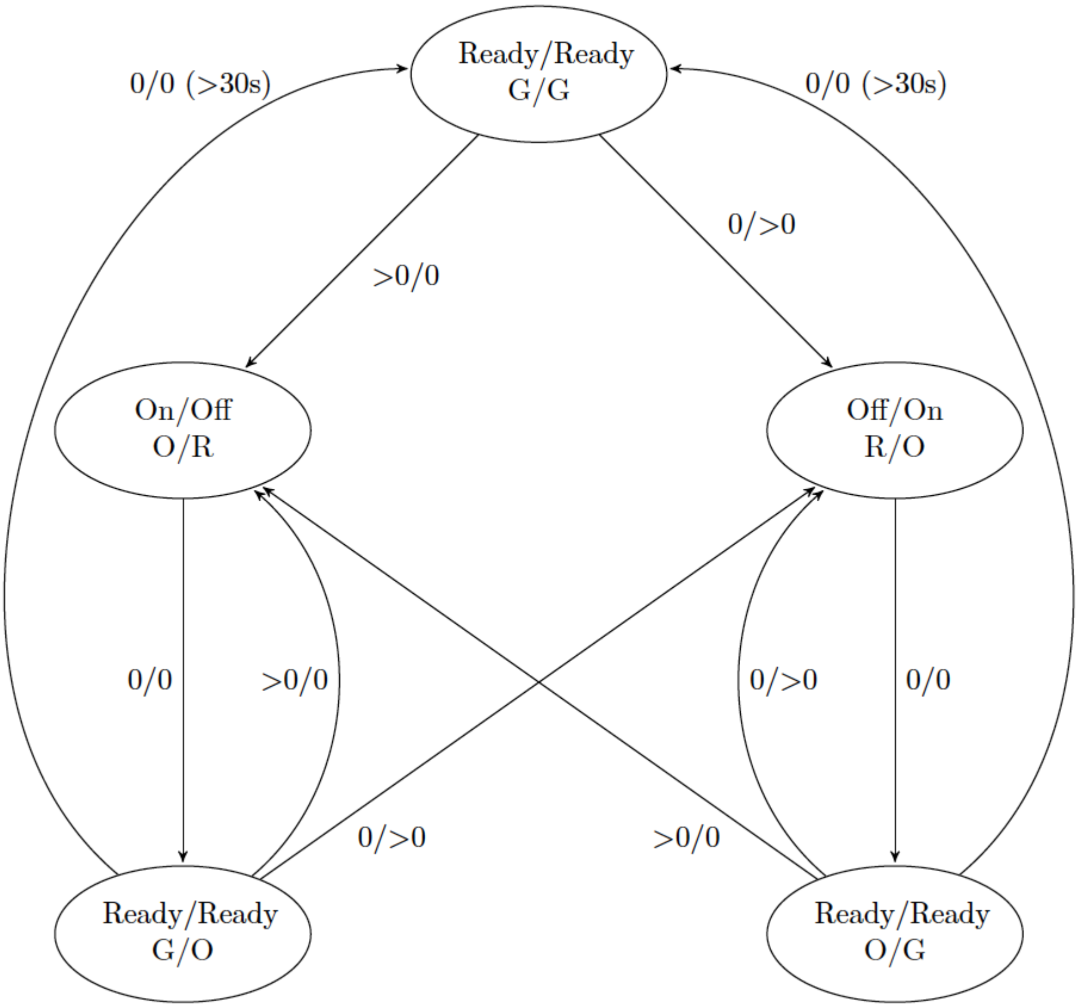


Figure 5.8: Use Case 2: Finite-state Machine

See listing A.2 for all relevant sequences created in MEMS.

5.3.4 Use Case 3 - Photovoltaic Station and Variable Priced Energy

Purpose

A building usually has a supply line from an electric utility company as its main electricity supplier. This usually guarantees the electrical power in the building. In recent decades, however, alternative energy generators have been added more and more frequently. These can be unidirectional systems such as PVSs, small wind turbines or fuel generators, or bidirectional systems such as an energy storage system or an electric vehicle.

The use of multiple energy sources and energy storage systems can have different motivations. Environmental reasons for renewable energy, cost savings, or self-sufficiency are some of them. Some of these motives are compatible with each other, but others are contradictory. Which objective is being pursued must be clarified before a use case is defined. Of course, several use cases can be defined and exchanged if desired. However, applying all use cases at the same time is not feasible.

This use case refers to being as cost-efficient as possible. For this purpose, it is assumed that an ESC supplies electricity from the grid, which has a time-dependent pricing model similar to that shown in section 2.4.5. Additionally, a PVS is installed that provides electricity at irregular intervals. However, the sale of surplus electricity is not economically viable. An energy storage system is also integrated into the building network to store energy.

In order to save costs, the energy storage unit is used as a quick buffer storage unit to avoid having to sell the surplus solar energy but also to consume the time-dependent cheap electricity and use it again at times when electricity is expensive.

Still, the supply of the building is above everything else. For this reason, the indicator from use case 1 is applied in addition and prioritized to support the internal network in critical situations or to interrupt the battery charging process.

Inputs and Outputs

The power storage unit is controlled in this use case. It can be charged (CHG: charging), do nothing (IDLE), or donate electricity (DON: donating). The energy storage can be empty (E), i.e. discharged or at the lower defined limit, have an average charging level (M) or be fully charged (F), i.e. at the upper defined limit. A total of nine different states result from each of the three options. From CHG_E (battery is empty and charging) to DON_F (battery is supplying power and is full).

Depending on the current state, the power mode (day/night) (energy mode), the electricity price (energy price), the line utilization from Use case 1 (line protection state), the energy currently supplied by the PVS compared to the energy consumed (PVS production) and a priority of the sequences (0... low priority, 100...high priority), a decision is made as to whether the current state is maintained or whether a transition into another state is made.

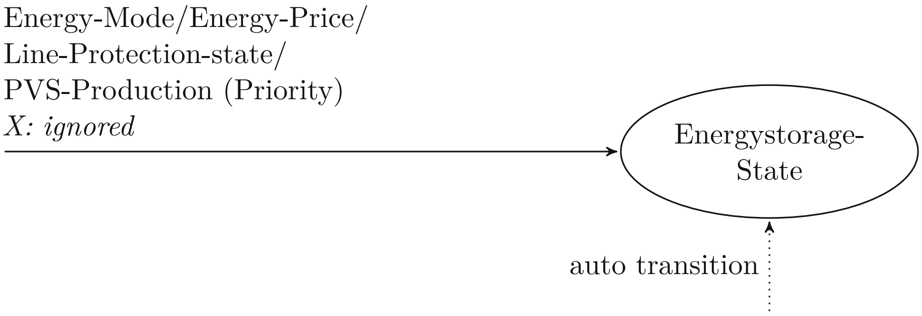


Figure 5.9: Use Case 3: FSM Legend

State Machine

If the energy storage system is discharging energy but is empty or is being charged but is full, it automatically switches to the IDLE state. This also applies if the internal power grid is fully used. If the electricity is very cheap or the PVS produces more electricity than consumed, the energy storage system is charged. If the electricity is too expensive, energy is drawn from the energy storage system.

The dashed lines mark an uninfluenceable transition that takes place directly at the energy storage system. If an energy storage system is empty but is being charged, it automatically switches to the CHG_M state.

For a simpler overview, all rules are shown in the FSM, but the transition is not marked from every state. The transition has been omitted in logical contexts. For example, for the transition from DON_M to CHG_M marked with a star*, it can be assumed that this rule also applies in the DON_F state.

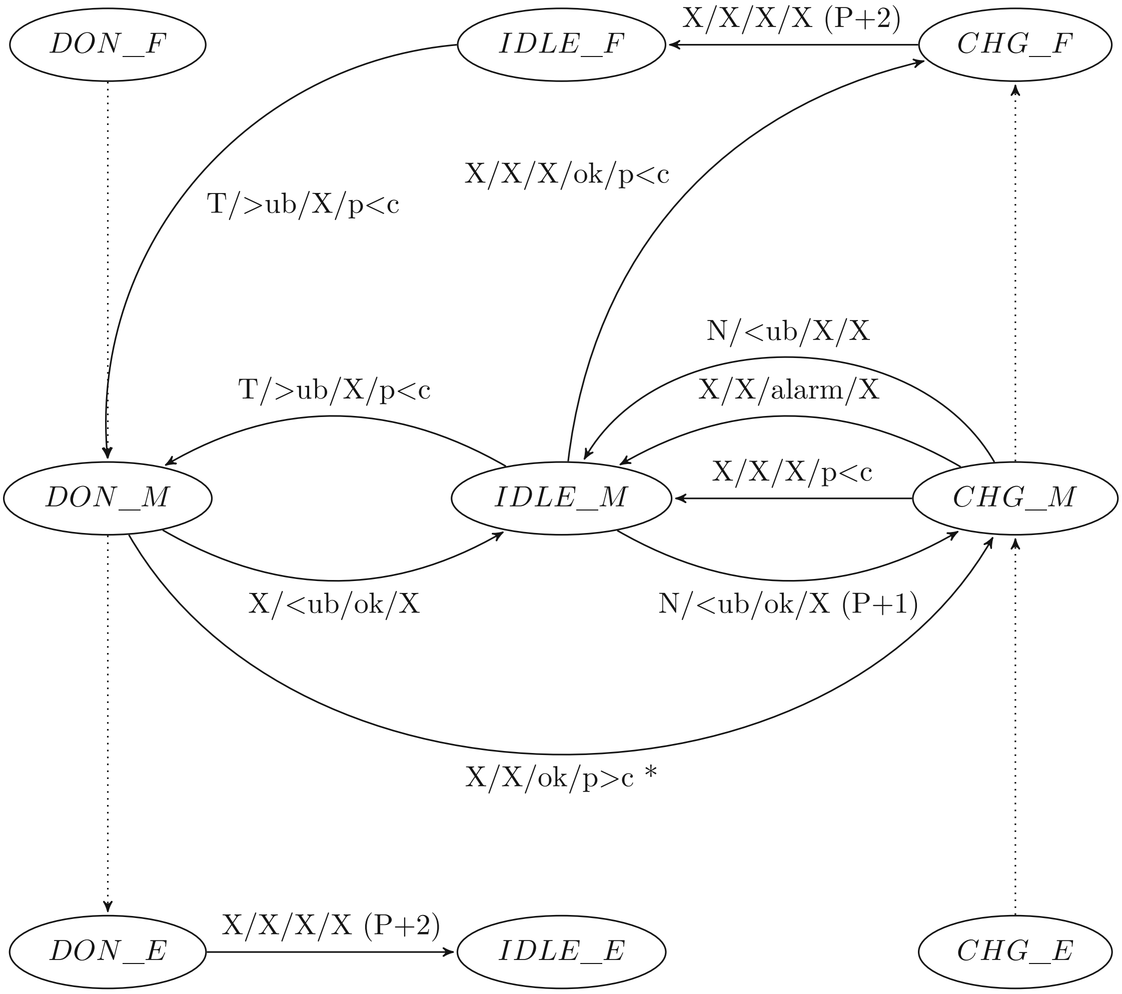


Figure 5.10: Use Case 3: Finite-state Machine

See listing A.3 for all relevant sequences created in MEMS.

5.4 Summary

In reference to section 1.2, this thesis raised different demands of the model design and the PoC that was to be evaluated. Due to the chosen communication model and the central control, implemented with a High-level Programming Language, different transmission media and protocols can be connected to the MEMS and thus fulfill requirement 1 and 3. With the help of the scheduler, the compare operators and the combine operators, datapoints can be compared, automations executed and thus requirement 2 is fulfilled.

Due to the central data storage within a database and the comprehensive design, which saves both operational data and system data, the entire MEMS can be configured via the database. Complex conditions for the automations can thus be nested, stored and checked in the scheduler and executed as required. Based on this, requirement 4 can enable operability for an end user as well as full configurability and expandability for experts. Adding the HMI component with a REST API, which allows datapoints to be controlled, automations to be created and the entire MEMS to be used, requirement 7 is reached.

Due to the implementation in Java, the system can be extended in function and scope at any time by Java-trained personnel and thus fulfills requirement 5. Requirement 6, the security of the MEMS, is attained by the local approach. The MEMS can operate isolatedly from the outside world and thus presents no opportunities for logical attacks. If the MEMS is integrated into the LAN, the HMI interface is secured by means of an authentication token. The database can only be accessed via the local instance and therefore cannot be changed directly. The communication component that connects the systems to the MEMS poses the greatest risk. This risk occurs not only because it can be extended, but also because a man-in-the-middle attack is possible. It can only be prevented by encrypted connections, sufficient authentication and general LAN security. The communication component is consequently not solely responsible for the security between the systems.

Requirement 8 was demonstrated in the evaluation through use cases. The MEMS can execute a wide variety of sequences and thus cover different objectives for each corresponding need. Regardless of whether the focus is on economy, the environment or comfort, the use cases created ensure that the desired goals are achieved.

CHAPTER 6

Conclusio

6.1 Summary

Connecting Home Automation Systems and Energy Systems, and controlling, automating and monitoring them poses many challenges. These include the connection of the systems, the normalization and comparison of the individual data and the further organization, control and processing of the different data. These challenges in combination with the requirements of the Modular Energy Management System necessitated a lot of development and implementation time.

The aim was to evaluate whether it is possible to develop an EMS independent of manufacturer, communication networks, application areas and a connection to the outside world. To this end, five different communication models were designed and evaluated using parameters that cover the entire development and usage cycle. These models are based on different transmission media, communication protocols and logic types. The model selected after the evaluation was implemented as a Proof of Concept. For this purpose, the High-level Programming Language Java and a relational database were used to connect different HASs and ESs in an object-oriented and centralized manner and to control them manually or automatically through defined use cases. The MEMS was successfully evaluated using a physical KNX system, a mock system and various use cases. Depending on the goals of the intended purpose, energy independence, cost efficiency or sustainability can be achieved.

In summary, it can be stated that an Energy Management System can be implemented with the necessary requirements (see section 5.4) and can be extensively expanded by the use of a Modular Energy Management System.

6.2 Future Work

The following section explains possible extensions for the MEMS. These extensions relate both to the range of functions and to improvements to current implementations. The PoC of the MEMS has been prepared for all the tasks listed and offers the necessary interfaces.

Artificial Intelligence Integration

Pattern recognition and AI could improve energy efficiency or cost savings. The goals of an EMS are usually not very different from the general public. Some basic goals have to be defined. Based on this, better use cases can be created through pattern recognition of the own MEMS, but also through patterns of other systems with similar goals. The AI can also adapt its own goals and needs individually to the MEMS. An AI controller could be connected via the HMI component's REST API.

Front-end

The control and configuration of the MEMS can be simplified for the user by means of a graphical front-end, for example a web interface. The current option via the REST console does not limit the control possibility, but is not user-friendly. A web interface, a (touch) display or other types of visualization make the MEMS usable for a professional and an amateur.

Device Monitoring and Statistics

In addition to automate functionality in a building, the MEMS can also be used to reduce the load on devices. If there are several devices with equivalent tasks, load, wear and operating times can be monitored to distribute usage in terms of loads or times to protect the devices. If the components themselves have a fitting measurement, this can already be integrated into the MEMS using use cases. However, it could also be added that the MEMS records the loads and times of the individual components and makes decisions based on this. Supplemented by statistics and wear and maintenance warnings, the components can be prevented from failing.

Analysis board

Currently, you have to rely on the target fulfillment of the use cases or evaluate the efficiency by analyzing the individual non-normalized system data. A comprehensive statistical evaluation from the logger's data and the visualization and analysis of the data can be used to check expected targets and graphically represent the entire MEMS.

Security

One of the requirements for the MEMS was security. An issue that is very important in any implementation. Security is ensured by the possibility of isolation, authentication and encryption. More intensive security mechanisms, exchange of tokens or similar with the connected systems or a more restrictive protection of the REST API can be added. Additional database access controls and the re-implementation of storing the systems authentication attributes can provide better protection for the MEMS. Finally, it should not be forgotten that the host should also be physically protected.

User Management

In addition to securing the HMI component with tokens, user management should be introduced. Not only because token exchange is impractical, but also because different groups of people should have different authorizations. Access via the HMI component can be protected and restricted by adding a user table to the database and HLP implementation.

Interim Datapoint Calculation

The compare operators were explained in section 4.4.1. Different variations of comparison types and comparison data can be selected for this. However, the PoC is limited to comparing only concrete datapoints/data. A calculation (e.g. difference between two datapoints) and further use is currently not implemented. By extending the database and the HLP implementation, it would be possible to create and store differences or perform other mathematical calculations.

Pre-defined Use Cases

In order to make the MEMS operational and, above all, practical for an installation, various use cases should be designed and implemented as templates. These can be selected and chosen in a use case collection and subsequently linked to the datapoints of the specific MEMS. In addition, a graphical use case designer could be implemented, which creates the necessary sequences in the MEMS from the use case.

APPENDIX A

Sequences of the Use Cases

In the following listings, the, in the MEMSs Proof of Concept, defined use cases (see section 5.3) are shown.

A.1 Use Case 1 Sequences

The following listing presents all defined sequences of use case 1:

```
Sequence 1 (Lines OK) [Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 1 (Lines OK):
  IF
    (
      !(
        building-L1 power consumption all (52) GREATEREQU
           building-line protection warning (157)
          OR
        building-L2 power consumption all (53) GREATEREQU
           building-line protection warning (157)
          OR
        building-L3 power consumption all (55) GREATEREQU
           building-line protection warning (157)
      )
    AND
      !(
        LED 4 (159) EQUALS 3
      )
    )
  THEN
    line-state LED green: LED 4 (159) WILL BE SET TO 3
```

```

Sequence 2 (Lines warning) [Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 2 (Lines warning):
  IF
    (
      (
        building-L1 power consumption all (52) GREATEREQU
           building-line protection warning (157)
          OR
        building-L2 power consumption all (53) GREATEREQU
           building-line protection warning (157)
          OR
        building-L3 power consumption all (55) GREATEREQU
           building-line protection warning (157)
      )
    AND
      !(
        LED 4 (159) EQUALS 6
      )
    AND
      (
        building-L1 power consumption all (52) LESS building-
           line protection alarm (158)
          AND
        building-L2 power consumption all (53) LESS building-
           line protection alarm (158)
          AND
        building-L3 power consumption all (55) LESS building-
           line protection alarm (158)
      )
    )
  THEN
    line-state LED orange: LED 4 (159) WILL BE SET TO 6

Sequence 3 (Lines alarm) [Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 3 (Lines alarm):
  IF
    (
      (
        building-L1 power consumption all (52) GREATEREQU
           building-line protection alarm (158)
          OR
        building-L2 power consumption all (53) GREATEREQU
```

```
           building-line protection alarm (158)
          OR
        building-L3 power consumption all (55) GREATEREQU
           building-line protection alarm (158)
      )
    AND
      !(
        LED 4 (159) EQUALS 2
      )
    )
  THEN
    line-state LED red: LED 4 (159) WILL BE SET TO 2
```

Listing A.1: Use Case 1: MEMS-Sequences

A.2 Use Case 2 Sequences

The following listing presents all defined sequences of use case 2:

```
Sequence 1 (Dishwasher<>Washingmachine Edge Case Resolver 1) [
   Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 1 (Dishwasher<>
   Washingmachine Edge Case Resolver 1.1):
  IF
    (
      dishwasher-state (32) EQUALS false
        AND
      washingmachine-state (23) EQUALS false
    )
  THEN
    washingmachine-state LED green: LED 5 (160) WILL BE SET TO
       3
    dishwasher-state LED red: LED 6 (161) WILL BE SET TO 2
    washingmachine-turn power on: washingmachine-ON/OFF (22)
       WILL BE SET TO true
WAIT FOR 2s Scene 2 (Dishwasher<>Washingmachine Edge Case
   Resolver 1.2):
  IF
    (
      dishwasher-state (32) EQUALS false
        AND
      washingmachine-state (23) EQUALS false
    )
  THEN
```

```
    dishwasher-state LED green: LED 6 (161) WILL BE SET TO 3
    dishwasher-turn power on: dishwasher-ON/OFF (30) WILL BE
       SET TO true

Sequence 2 (Dishwasher turns on and lines are saved (WM turned
   off)) [Priority 10]:
WAIT FOR 0s, OVERWRITE ALL DATAPOINTS AND CHECK RULES OF Scene
   3 (Dishwasher turns on and lines are saved (WM turned off)):
  IF
    (
      dishwasher-power consumption (28) GREATER 0.0
        AND
      washingmachine-state (23) EQUALS true
        AND
      dishwasher-state (32) EQUALS true
    )
  THEN
    washingmachine-turn power off: washingmachine-ON/OFF (22)
       WILL BE SET TO false
    washingmachine-state LED red: LED 5 (160) WILL BE SET TO 2
    dishwasher-state LED orange: LED 6 (161) WILL BE SET TO 6

Sequence 3 (Washingmachine turns on and lines are saved (DW
   turned off)) [Priority 10]:
WAIT FOR 0s, OVERWRITE ALL DATAPOINTS AND CHECK RULES OF Scene
   4 (Washingmachine turns on and lines are saved (DW turned
   off)):
  IF
    (
      washingmachine-power consumption (21) GREATER 0
        AND
      dishwasher-state (32) EQUALS true
        AND
      washingmachine-state (23) EQUALS true
    )
  THEN
    dishwasher-turn power off: dishwasher-ON/OFF (30) WILL BE
       SET TO false
    dishwasher-state LED red: LED 6 (161) WILL BE SET TO 2
    washingmachine-state LED orange: LED 5 (160) WILL BE SET TO
        6

Sequence 4 (Dishwasher finished) [Priority 10]:
```

```
WAIT FOR 0s AND CHECK RULES OF Scene 5 (Dishwasher finished 1):
  IF
    (
      dishwasher-power consumption (28) LESSEQU 0
        AND
      dishwasher-state (32) EQUALS true
        AND
      washingmachine-state (23) EQUALS false
    )
  THEN
    washingmachine-turn power on: washingmachine-ON/OFF (22)
       WILL BE SET TO true
    washingmachine-state LED orange: LED 5 (160) WILL BE SET TO
        6
    dishwasher-state LED green: LED 6 (161) WILL BE SET TO 3
WAIT FOR 10s Scene 6 (Dishwasher finished 2):
  THEN
    washingmachine-state LED green: LED 5 (160) WILL BE SET TO
       3
    dishwasher-state LED green: LED 6 (161) WILL BE SET TO 3

Sequence 5 (Washingmachine finished) [Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 7 (Washingmachine finished
    1):
  IF
    (
      washingmachine-power consumption (21) LESSEQU 0
        AND
      washingmachine-state (23) EQUALS true
        AND
      dishwasher-state (32) EQUALS false
    )
  THEN
    dishwasher-turn power on: dishwasher-ON/OFF (30) WILL BE
       SET TO true
    dishwasher-state LED orange: LED 6 (161) WILL BE SET TO 6
    washingmachine-state LED green: LED 5 (160) WILL BE SET TO
       3
WAIT FOR 10s Scene 8 (Washingmachine finished 2):
  THEN
    dishwasher-state LED green: LED 6 (161) WILL BE SET TO 3
    washingmachine-state LED green: LED 5 (160) WILL BE SET TO
       3
```

Listing A.2: Use Case 2: MEMS-Sequences

A.3 Use Case 3 Sequences

The following listing presents all defined sequences of use case 3:

```
Sequence 1 (charge energy storage at night (price <NOG)) [
   Priority 12]:
WAIT FOR 0s AND CHECK RULES OF Scene 1 (charge energy storage
   at night (if electricity is cheap)):
  IF
    (
      ESC power mode (14) EQUALS 2
        AND
      ESC electricity price night (13) LESS electricity price
         night upper limit (164)
        AND
      Energy storage charge level (11) LESSEQU Energy storage 1
          upper protection limit (167)
        AND
      Energy storage mode state (8) LESSEQU 0
        AND
      LED 4 (159) EQUALS 3
    )
  THEN
    Load energy storage: Energy storage mode set (7) WILL BE
       SET TO 8

Sequence 2 (charge energy storage on day (price >TOG)) [
   Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 2 (charge energy storage
   by day):
  IF
    (
      Energy storage level (11) GREATER Energy storage 1
         protection limit below (166)
        AND
      ESC current mode (14) EQUALS 1
        AND
      Energy supplier electricity price day (12) GREATER
         Electricity price day upper limit (162)
        AND
```

```
   Energy storage mode State (8) EQUALS 0
    AND
   PV system power output (9) LESS Object-L1 power
     consumption total (52)
  )
 THEN
  Charge energy storage with -8 kW: Energy storage mode set
    (7) WILL BE SET TO -8

Sequence 3 (Load energy storage: PV produces enough power) [
  Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 3 (charge energy storage
  during the day):
 IF
  (
   Energy Storage Mode State (8) LESSEQU 0
    AND
   LED 4 (159) EQUALS 3
    AND
   PV system power output (9) GREATER Object-L1 power
     consumption total (52)
    AND
   Energy storage charge level (11) LESSEQU Energy storage 1
     upper protection limit (167)
  )
 THEN
  Load energy storage: Energy storage mode set (7) WILL BE
    SET TO 8

Sequence 4 (stop charging energy storage during the day (PV
  does not produce enough)) [Priority 10]:
WAIT FOR 0s AND CHECK RULES OF Scene 4 (no longer charge energy
  storage):
 IF
  (
   Energy Storage Mode State (8) GREATER 0
    AND
   ESC current mode (14) EQUALS 1
    AND
   PV system power output (9) LESS Object-L1 power
     consumption total (52)
  )
 THEN
```

```
    No longer charge/load energy storage: Storage mode set (7)
       WILL BE SET TO 0

Sequence 5 (no longer charge energy storage during day (price <
    TOG)) [Priority 12]:
WAIT FOR 0s AND CHECK RULES OF Scene 5 (no longer charge energy
     storage):
  IF
    (
      RU electricity price day (12) LESSEQU electricity price
         day upper limit (162)
        AND
      Energy supplier electricity mode (14) EQUALS 1
        AND
      Energy storage mode State (8) LESS 0
        AND
      LED 4 (159) EQUALS 3
    )
  THEN
    No longer load/charge energy storage: Energy storage mode
       set (7) WILL BE SET TO 0

Sequence 6 (energy storage full: no more loading) [Priority
    14]:
WAIT FOR 0s AND CHECK RULES OF Scene 6 (no longer charge energy
     storage):
  IF
    (
      Energy storage full (11) GREATEREQU Energy storage 1
         upper protection limit (167)
        AND
      Energy storage mode state (8) GREATER 0
    )
  THEN
    No longer load/charge energy storage: Energy storage mode
       set (7) WILL BE SET TO 0

Sequence 7 (System overloaded: No more loading) [Priority 14]:
WAIT FOR 0s AND CHECK RULES OF Scene 7 (no longer charge energy
     storage):
  IF
    (
      LED 4 (159) EQUALS 6
```

```
        AND
      Energy storage mode State (8) GREATER 0
    )
  THEN
    Energy storage no longer loaded/charged: Storage mode set
       (7) WILL BE SET TO 0

Sequence 8 (Storage empty: No more charging) [Priority 14]:
WAIT FOR 0s AND CHECK RULES OF Scene 8 (no longer charge energy
    storage):
  IF
    (
      Energy storage level (11) LESSEQU Energy storage 1
         protection limit down (166)
        AND
      Energy storage mode state (8) LESS 0
    )
  THEN
    No longer load/charge energy storage: Memory mode set (7)
       WILL BE SET TO 0
```

Listing A.3: Use Case 3: MEMS-Sequences

APPENDIX B

Compare Operator Overview

In the following tables the possible usages of the compare operators depending on the compared data and dataTypes is shown.

	RISING	FALLING	STALLING	NOT_UPDATED	EQUALS	NOT_EQUALS	GREATER	GREATEREQU	LESS	LESSEQU	DAY	TODAY	LAST_CHANGED_BEFORE	LAST_CHANGED_AFTER	LAST_UPDATED_BEFORE	LAST_UPDATED_AFTER	YEAR
BOOLEAN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Control2Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ControlDimmingBlinds4Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CharacterSet1Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UnsignedValue1Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SignedValue1Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UnsignedValue2Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SignedValue2Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Float2Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Time3Byte	-	-	-	-	1	1	1	1	1	1	-	-	-	-	-	-	-
Date3Byte	-	-	-	-	1	1	1	1	1	1	-	1	-	-	-	-	1
UnsignedValue4Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SignedValue4Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Float4Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
String14Byte	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	1
SceneControl1Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.1: Compare Operators for One Data

	RISING	FALLING	STALLING	NOT_UPDATED	EQUALS	NOT_EQUALS	GREATER	GREATEREQU	LESS	LESSEQU	DAY	TODAY	LAST_CHANGED_BEFORE	LAST_CHANGED_AFTER	LAST_UPDATED_BEFORE	LAST_UPDATED_AFTER	YEAR
BOOLEAN	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-
Control2Bit	-	-	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-
ControlDimmingBlinds4Bit	-	-	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-
CharacterSet1Byte	-	-	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-
UnsignedValue1Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
SignedValue1Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
UnsignedValue2Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
SignedValue2Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
Float2Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
Time3Byte	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-	-
Date3Byte	1	1	1	1	1	1	1	1	1	1	-	1	-	-	-	-	1
UnsignedValue4Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
SignedValue4Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
Float4Byte	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1
String14Byte	1	1	1	1	1	-	-	-	-	-	1	-	-	-	-	-	1
SceneCtrl1Byte	-	-	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.2: Compare Operators for One Datapoint

	RISING	FALLING	STALLING	NOT_UPDATED	EQUALS	NOT_EQUALS	GREATER	GREATEREQU	LESS	LESSEQU	DAY	TODAY	LAST_CHANGED_BEFORE	LAST_CHANGED_AFTER	LAST_UPDATED_BEFORE	LAST_UPDATED_AFTER	YEAR
BOOLEAN	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Control2Bit	-	-	-	-	1	1	-	-	-	-	-	-	1	1	1	1	-
ControlDimmingBlinds4Bit	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
CharacterSet1Byte	-	-	-	-	1	1	-	-	-	-	-	-	1	1	1	1	-
UnsignedValue1Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	1
SignedValue1Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	1
UnsignedValue2Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	1
SignedValue2Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	1
Float2Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Time3Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Date3Byte	-	-	-	-	1	1	1	1	1	1	1	-	1	1	1	1	1
UnsignedValue4Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
SignedValue4Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Float4Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
String14Byte	-	-	-	-	1	1	1	1	1	1	1	-	1	1	1	1	1
SceneCtrl1Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-

Table B.3: Compare Operators for Two Datapoints

	RISING	FALLING	STALLING	NOT_UPDATED	EQUALS	NOT_EQUALS	GREATER	GREATEREQU	LESS	LESSEQU	DAY	TODAY	LAST_CHANGED_BEFORE	LAST_CHANGED_AFTER	LAST_UPDATED_BEFORE	LAST_UPDATED_AFTER	YEAR
BOOLEAN	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Control2Bit	-	-	-	-	1	1	-	-	-	-	-	-	1	1	1	1	-
ControlDimmingBlinds4Bit	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
CharacterSet1Byte	-	-	-	-	1	1	-	-	-	-	-	-	1	1	1	1	-
UnsignedValue1Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
SignedValue1Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
UnsignedValue2Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
SignedValue2Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Float2Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Time3Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Date3Byte	-	-	-	-	1	1	1	1	1	1	1	-	1	1	1	1	1
UnsignedValue4Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
SignedValue4Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
Float4Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-
String14Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	1
SceneCtrl1Byte	-	-	-	-	1	1	1	1	1	1	-	-	1	1	1	1	-

Table B.4: Compare Operators for One Datapoint and One Data

	RISING	FALLING	STALLING	NOT_UPDATED	EQUALS	NOT_EQUALS	GREATER	GREATEREQU	LESS	LESSEQU	DAY	TODAY	LAST_CHANGED_BEFORE	LAST_CHANGED_AFTER	LAST_UPDATED_BEFORE	LAST_UPDATED_AFTER	YEAR
BOOLEAN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Control2Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ControlDimmingBlinds4Bit	-	-	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
CharacterSet1Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UnsignedValue1Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
SignedValue1Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
UnsignedValue2Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
SignedValue2Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
Float2Byte	0	0	-	-	1	1	1	1	1	1	-	-	-	-	-	-	-
Time3Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
Date3Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
UnsignedValue4Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
SignedValue4Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
Float4Byte	0	0	-	-	1	1	1	1	1	1	-	-	-	-	-	-	-
String14Byte	0	0	-	-	0	0	0	0	0	0	-	-	-	-	-	-	-
SceneCtrl1Byte	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.5: Compare Operators for Two Datapoints and One Data

List of Figures

List of Tables

Acronyms

AC Alternating Current. 27, 28

AI Artificial Intelligence. 4, 10, 55, 102

API Application Programming Interfaces. 55, 83, 86, 100, 102, 103

BAOS Bus Access and Object Server. 30, 31, 41–44, 85, 86, 88

BLE Bluetooth Low Energy. 10

CAN Controller Area Network. 28, 32

CMS Content Management System. 38

DC Direct Current. 27, 28

DML Data Manipulation Language. 58

DSG Distributed Systems Group. 6

EMS Energy Management System. 1, 2, 5, 14–17, 22–24, 57, 101, 102

ES Energy System. 2–7, 9, 12–14, 16, 17, 25, 27, 30–32, 34, 36, 53, 60, 83, 101

ESC Energy Supply Company. 1–3, 5, 12, 13, 16, 17, 19, 21, 22, 28, 90, 97

ETS Engineering Tool Software. 31, 32, 38, 39, 41, 43, 45, 46, 85

FSM Finite-State Machine. 88, 89, 92, 98

HAS Home Automation System. 2–7, 9–12, 14, 16, 17, 23, 24, 27, 30–32, 34, 36, 41, 47–49, 53, 60, 83, 86, 88, 91, 101

HLP High-level Programming Language. 28, 30–32, 34, 41–43, 47, 49, 55, 58, 99, 101, 103

HMI Human-Machine Interface. 5, 8, 24, 26, 30, 31, 34, 36, 37, 39, 45, 51, 53, 55, 100, 102, 103

IoT Internet of Things. 7

JDBC Java Database Connectivity. 55

JSON JavaScript Object Notation. 30, 38

LAN Local Area Network. 10, 28, 30, 34, 37, 41, 42, 47–50, 83, 100

MEMS Modular Energy Management System. 4, 5, 7, 22, 26, 40, 53–58, 60–62, 64, 69–73, 83–86, 88, 90, 93, 96, 99–103, 105

PK Primary Key. 62–68

PoC Proof of Concept. 4, 6, 25, 26, 40, 53, 55, 71, 83, 99, 101–103, 105

PVS Photovoltaic Station. 1, 5, 12, 19, 20, 22, 90, 96–98

REST Representational State Transfer. 23, 28, 30, 34–36, 38, 51, 55, 83, 84, 86, 88, 95, 100, 102, 103

SHMM Smart House Monitoring and Management. 24

UART Universal Asynchronous Receiver Transmitter. 28, 36, 37, 49

USP Unique Selling Proposition. 2

VPN Virtual Private Network. 5, 38

Bibliography

[Age] International Energy Agency. Electric vehicles - Tracking Electric Vehicles. `https://www.iea.org/energy-system/transport/electric-vehicles`. last visited 13.03.2024.

[AKM22] Nikoleta Andreadou, Evangelos Kotsakis, and Marcelo Masera. Interoperability Testing of a Smart Home Automation System under Explicit Demand Response Schemes. *Energies*, 15(21):7952, January 2022. Number: 21, Publisher: Multidisciplinary Digital Publishing Institute, last visited 13.03.2024.

[Alh04] A. Alheraish. Design and implementation of home automation system. *IEEE Transactions on Consumer Electronics*, 50(4):1087–1092, November 2004. Conference Name: IEEE Transactions on Consumer Electronics, last visited 13.03.2024.

[AMM15] Sereen Althaher, Pierluigi Mancarella, and Joseph Mutale. Automated Demand Response From Home Energy Management System Under Dynamic Pricing and Power and Comfort Constraints. *IEEE Transactions on Smart Grid*, 6(4):1874–1883, July 2015. Conference Name: IEEE Transactions on Smart Grid, last visited 13.03.2024.

[AO15] Marie Aarestrup Aasness and James Odeck. The increase of electric vehicle usage in Norway—incentives and adverse effects. *European Transport Research Review*, 7(4):34, October 2015. last visited 13.03.2024.

[Assa] KNX Association. A brief introduction to KNX – KNX Association [Official website]. `https://www.knx.org/knx-en/for-professionals/What-is-KNX/A-brief-introduction/index.php`. last visited 15.03.2024.

[Assb] KNX Association. Seamlessly integrating alexa with knx smart home technology – KNX Association [Official website]. last visited 15.03.2024.

[AST11] Mahnoosh Alizadeh, Anna Scaglione, and Robert J. Thomas. Direct load management of electric vehicles. In *2011 IEEE International Conference* *on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5964–5967, Prague, Czech Republic, May 2011. IEEE. last visited 13.03.2024.

[AZ19] Zahrah A. Almusaylim and Noor Zaman. A review on smart home present state and challenges: linked to context-awareness internet of things (IoT). *Wireless Networks*, 25(6):3193–3204, August 2019. last visited 13.03.2024.

[BBP23] Dimitri Belli, Paolo Barsocchi, and Filippo Palumbo. Connectivity Standards Alliance Matter: State of the art and opportunities. *Internet of Things*, 25:101005, November 2023. last visited 13.03.2024.

[BfK] Energie Mobilität Innovation und Technologie Bundesministerium für Klimaschutz, Umwelt. Erneuerbaren-Wärme-Gesetz (EWG): ab 2023 keine Gasheizungen in Neubauten. last visited 13.03.2024.

[BS] Inc. Bluetooth SIG. Bluetooth Technologieübersicht | Bluetooth® Technologie Website. `https://www.bluetooth.com/de/learn-about-bluetooth/tech-overview/`. last visited 15.03.2024.

[BZ15] Marc Beaudin and Hamidreza Zareipour. Home energy management systems: A review of modelling and complexity. *Renewable and Sustainable Energy Reviews*, 45:318–335, May 2015. last visited 13.03.2024.

[Com] INNOVATION FUND European Commission. Green the Flex: Achieving significant greenhouse gas emission reductions by intelligent electricity load shifting on the customer level. Technical report. last visited 13.03.2024.

[DPG06] Armando Roy Delgado, Rich Picking, and Vic Grout. Remote-Controlled Home Automation Systems with Different Network Technologies. Glyndŵr University Research Online, July 2006. last visited 13.03.2024.

[DRD16] Shopan Dey, Ayon Roy, and Sandip Das. Home automation using Internet of Thing. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1–6, New York City, NY, USA, October 2016. IEEE. last visited 13.03.2024.

[DSdHR16] Laura Daniele, Monika Solanki, Frank den Hartog, and Jasper Roes. Interoperability for Smart Appliances in the IoT World. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016*, pages 21–29, Cham, 2016. Springer International Publishing. last visited 13.03.2024.

[EA21] Rasha El-Azab. Smart homes: potentials and challenges. *Clean Energy*, 5(2):302–315, June 2021. last visited 13.03.2024.

[EM15] Dalia Eltigani and Syafrudin Masri. Challenges of integrating renewable energy sources to smart grids: A review. *Renewable and Sustainable Energy Reviews*, 52:770–780, December 2015. last visited 13.03.2024.

[EMASA17] Ali M. Eltamaly, Mohamed A. Mohamed, M. S. Al-Saud, and Abdulrahman I. Alolah. Load management as a smart grid concept for sizing and designing of hybrid renewable energy systems. *Engineering Optimization*, 49(10):1813–1828, October 2017. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/0305215X.2016.1261246, last visited 13.03.2024.

[ERDW16] Maximilian Engelken, Benedikt Römer, Marcus Drescher, and Isabell Welpe. Transforming the energy system: Why municipalities strive for energy self-sufficiency. *Energy Policy*, 98:365–377, November 2016. last visited 13.03.2024.

[Exn23] Carmen Exner. flexQgrid: Mit dem Smart Grid den "Haushalt der Zukunft" ermöglichen. *Energiewirtschaftliche Tagesfragen*, 73(Heft 1-2):54–56, 2023. last visited 13.03.2024.

[FHNB19] Zohreh Fotouhi, Massoud Reza Hashemi, Hamed Narimani, and Islam Safak Bayram. A General Model for EV Drivers' Charging Behavior. *IEEE Transactions on Vehicular Technology*, 68(8):7368–7382, August 2019. Conference Name: IEEE Transactions on Vehicular Technology, last visited 13.03.2024.

[Fou] The Apache Software Foundation. Maven – Introduction. `https://maven.apache.org/what-is-maven.html`. last visited 20.03.2024.

[Fou24] GEF Green European Foundation. Energy Atlas 2018 - Facts and Figures about Renewables in Europe, February 2024. last visited 13.03.2024.

[Gmba] Fronius International GmbH. Solarenergie für Zuhause - Fronius Solar Energy. `https://www.fronius.com.ua/de-at/austria/solarenergie/eigenheim`. last visited 16.03.2024.

[Gmbb] Weinzierl Engineering GmbH. KNX IP BAOS 777 - Operating manual. `https://weinzierl.de/images/download/products/777/weinzierl-777-knx-ip-baos-5193-manual-en.pdf`. last visited 21.03.2024.

[GSZW19] Xiao Guo, Zhenjiang Shen, Yajing Zhang, and Teng Wu. Review on the Application of Artificial Intelligence in Smart Homes. *Smart Cities*, 2(3):402–420, September 2019. Number: 3, Publisher: Multidisciplinary Digital Publishing Institute, last visited 13.03.2024.

[IEA] Dr Fatih International Energy Agency. World Energy Outlook 2022. last visited 13.03.2024.

[IFT] IFTTT. IFTTT - Automate business & home. `https://ifttt.com/`. last visited 13.03.2024.

[IVNO20] Semich Impram, Secil Varbak Nese, and Bülent Oral. Challenges of renewable energy penetration on power system flexibility: A survey. *Energy Strategy Reviews*, 31:100539, September 2020. last visited 13.03.2024.

[JK17] Markandeshwar Jerabandi and Mallikarjun M Kodabagi. A review on home automation system. In *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, pages 1411–1415, Bangalore, August 2017. IEEE. last visited 13.03.2024.

[JKR+19] Waheb A. Jabbar, Tee Kok Kian, Roshahliza M. Ramli, Siti Nabila Zubir, Nurthaqifah S. M. Zamrizaman, Mohammed Balfaqih, Vladimir Shepelev, and Soltan Alharbi. Design and Fabrication of Smart Home With Internet of Things Enabled Automation System. *IEEE Access*, 7:144059–144074, 2019. Conference Name: IEEE Access, last visited 13.03.2024.

[Koc] Jan Koch. Solardachpflicht NRW: Ab 2025 Photovoltaik auf jedem Dach. last visited 13.03.2024.

[LA14] Hayet Lamine and Hafedh Abid. Remote control of a domestic equipment from an Android application based on Raspberry pi card. In *2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 903–908, Hammamet, Tunisia, December 2014. IEEE. last visited 13.03.2024.

[LC16] Dasheng Lee and Chin-Chi Cheng. Energy savings by energy management systems: A review. *Renewable and Sustainable Energy Reviews*, 56:760–777, April 2016. last visited 13.03.2024.

[LCLP15] Jean-Nicolas Louis, Antonio Calo, Kauko Leiviskä, and Eva Pongrácz. Environmental Impacts and Benefits of Smart Home Automation: Life Cycle Assessment of Home Energy Management System. *IFAC-PapersOnLine*, 48(1):880–885, January 2015. last visited 13.03.2024.

[LGRC20] Joaquim Leitao, Paulo Gil, Bernardete Ribeiro, and Alberto Cardoso. A Survey on Home Energy Management. *IEEE Access*, 8:5699–5722, 2020. last visited 13.03.2024.

[LM17] Henrik Lund, Poul Alberg Østergaard, David Connolly, and Brian Vad Mathiesen. Smart energy and smart energy systems. *Energy*, 137:556–565, October 2017. last visited 13.03.2024.

[Mat] EVN / Daniela Matejschek. Green the flex: Stromkunden leisten Beitrag zur Systemstabilisierung und sparen CO2 ein. *EVN*. last visited 13.03.2024.

[MBV21] T Maragatham, P Balasubramanie, and M Vivekanandhan. IoT Based Home Automation System using Raspberry Pi 4. *IOP Conference Series: Materials Science and Engineering*, 1055(1):012081, February 2021. last visited 13.03.2024.

[Mei23] Philip Meissner. Alles im gruenen bereich? *Harvard Business manager*, pages 57–60, 2023. last visited 05.03.2024.

[MJKR15] Anzar Mahmood, Nadeem Javaid, Muhammad Asghar Khan, and Sohail Razzaq. An overview of load management techniques in smart grid. *International Journal of Energy Research*, 39(11):1437–1450, 2015. last visited 13.03.2024.

[MS17] David Mohler and Daniel Sowder. Chapter 23 - Energy Storage and the Need for Flexibility on the Grid. In Lawrence E. Jones, editor, *Renewable Energy Integration (Second Edition)*, pages 309–316. Academic Press, Boston, January 2017. last visited 13.03.2024.

[MSO17] Daniel Minoli, Kazem Sohraby, and Benedict Occhiogrosso. IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet of Things Journal*, 4(1):269–283, February 2017. Conference Name: IEEE Internet of Things Journal, last visited 13.03.2024.

[Neu21] J Neuhaus. Die Ladeleistung von Elektroautos einfach erklärt | EnBW. `https://www.enbw.com/blog/elektromobilitaet/laden/die-ladeleistung-eines-elektroautos/`, March 2021. last visited 13.03.2024.

[noaa] H2 Database Engine. `https://www.h2database.com/html/main.html`. last visited 20.03.2024.

[noab] PSI Software SE: Informationen zum Cyberangriff auf PSI | Information on cyber attack on PSI. `https://www.psigridconnect.com/`. last visited 13.03.2024.

[noa23] Übersicht zur Solarpflicht für Gebäude, December 2023. last visited 13.03.2024.

[NS19] Lakshmi Kanthan Narayanan and Suresh Sankaranarayanan. IoT Enabled Smart Water Distribution and Underground Pipe Health Monitoring Architecture for Smart Cities. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–7, Bombay, India, March 2019. IEEE. last visited 13.03.2024.

[OB16] Manasseh Obi and Robert Bass. Trends and challenges of grid-connected photovoltaic systems – A review. *Renewable and Sustainable Energy Reviews*, 58:1082–1094, May 2016. last visited 13.03.2024.

[OFHB16] Andres Ovalle, Julian Fernandez, Ahmad Hably, and Seddik Bacha. An Electric Vehicle Load Management Application of the Mixed Strategist Dynamics and the Maximum Entropy Principle. *IEEE Transactions on Industrial Electronics*, 63(5):3060–3071, May 2016. Conference Name: IEEE Transactions on Industrial Electronics, last visited 13.03.2024.

[Oraa] Oracle. Java JDBC API. `https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/`. last visited 20.03.2024.

[Orab] Oracle. Java-Software. `https://www.oracle.com/java/`. last visited 20.03.2024.

[PGFGB+21] Lisardo Prieto González, Anna Fensel, Juan Miguel Gómez Berbís, Angela Popa, and Antonio de Amescua Seco. A Survey on Energy Efficiency in Smart Homes and Smart Grids. *Energies*, 14(21):7273, January 2021. Number: 21, Publisher: Multidisciplinary Digital Publishing Institute, last visited 13.03.2024.

[PIP+22] Gaetanino Paolone, Danilo Iachetti, Romolo Paesani, Francesco Pilotti, Martina Marinelli, and Paolino Di Felice. A Holistic Overview of the Internet of Things Ecosystem. *IoT*, 3(4):398–434, December 2022. Number: 4, Publisher: Multidisciplinary Digital Publishing Institute, last visited 13.03.2024.

[PVK19] Prakash Pawar and Panduranga Vittal K. Design and development of advanced smart energy management system integrated with IoT framework in smart grid environment. *Journal of Energy Storage*, 25:100846, October 2019. last visited 13.03.2024.

[QTOL15] Jairo Quirós-Tortós, Luis F. Ochoa, and Becky Lees. A statistical analysis of EV charging behavior in the UK. In *2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)*, pages 445–449, October 2015. last visited 13.03.2024.

[Ros] Linda Rosencrance. What is Zigbee? | Definition from TechTarget. last visited 13.03.2024.

[Sch15] Stefan Schneyer. Leistungskurven bei Photovoltaikanlagen in Therie und Praxis. `https://www.edu.sot.tum.de/fileadmin/w00bed/edu/Schuelerkonferenz/Seminararbeiten_2015/schneyer_stefan_photovoltaik.pdf`, 2015. last visited 13.03.2024.

[SFDR20] Benjamin K. Sovacool and Dylan D. Furszyfer Del Rio. Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies. *Renewable and Sustainable Energy Reviews*, 120:109663, March 2020. last visited 13.03.2024.

[SHGS17] O. Schmidt, A. Hawkes, A. Gambhir, and I. Staffell. The future cost of electrical energy storage based on experience rates. *Nature Energy*, 2(8):1–8, July 2017. Publisher: Nature Publishing Group, last visited 13.03.2024.

[SJ19] Tara Salman and Raj Jain. A Survey of Protocols and Standards for Internet of Things, February 2019. last visited 13.03.2024.

[SJL15] Jingshuang Shen, Chuanwen Jiang, and Bosong Li. Controllable Load Management Approaches in Smart Grids. *Energies*, 8(10):11187–11202, October 2015. Number: 10, Publisher: Multidisciplinary Digital Publishing Institute, last visited 13.03.2024.

[Sof] SmartBear Software. Bearer Authentication. `https://swagger.io/docs/specification/authentication/bearer-authentication/`. last visited 21.03.2024.

[Tan] VMWare Tanzu. Why Spring. `https://spring.io/why-spring`. last visited 20.03.2024.

[TC19] Mehmet Y. Toylan and Engin Cetin. Design and application of a KNX-based home automation simulator for smart home system education. *Computer Applications in Engineering Education*, 27(6):1465–1484, 2019. last visited 13.03.2024.

[TLPL14] Shih-Pang Tseng, Bo-Rong Li, Jun-Long Pan, and Chia-Ju Lin. An application of Internet of things with motion sensing on smart house. In *2014 International Conference on Orange Technologies*, pages 65–68, September 2014. last visited 13.03.2024.

[Wik24a] Wikipedia. Controller Area Network. `https://en.wikipedia.org/wiki/CAN_bus`, February 2024. last visited 18.03.2024.

[Wik24b] Wikipedia. Electrical grid. `https://en.wikipedia.org/w/index.php?title=Electrical_grid&oldid=1213303346`, March 2024. last visited 13.03.2024.

[Wik24c] Wikipedia. WebSocket. `https://en.wikipedia.org/w/index.php?title=WebSocket&oldid=1212138654`, March 2024. Page Version ID: 1212138654.

[Wik24d] Wikipedia. Wi-Fi. `https://en.wikipedia.org/w/index.php?title=Wi-Fi&oldid=1213688950`, March 2024. last visited 15.03.2024.

[YBMK18] Yuqing Yang, Stephen Bremner, Chris Menictas, and Merlinde Kay. Battery energy storage system size determination in renewable energy systems:

A review. *Renewable and Sustainable Energy Reviews*, 91:109–125, August 2018. last visited 13.03.2024.

[ZA] Österreichischer Zivilschutzverband, Bundesverband Spiegelgasse 6/13, and 1010 Wien Austria. Blackout - der längerfristige Stromausfall | Aktuelles. *ZIVILSCHUTZVERBAND.AT*. last visited 28.09.2021.

[ZLC+16] Bin Zhou, Wentao Li, Ka Wing Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61:30–40, August 2016. last visited 13.03.2024.

[GW] Österreichischer Zivilschutzverband Grünangergasse 8/5 1010 Wien. Ratgeber blackout - was tun wenn der strom ausgeht? *ZIVILSCHUTZVERBAND.AT*. last visited 05.03.2024.