A hierarchical approach to monitoring SCADA networks

Masterthesis

by **Verena Menzel**

Matriculation number: 426918

Computer Science (Master of Science)

Supervised by:

Prof. Dr. Anne Remke

Westfälische Wilhelms-Universität Münster

Department of Mathematics and Computer Science

Group of Safety-Critical Systems

Münster, March 16, 2021

Eine hierachische Herangehensweise zur Überwachung von SCADA Netzwerken

Masterarbeit

von **Verena Menzel**

Matrikelnummer: 426918

Informatik (Master of Science)

Thema gestellt von:

Prof. Dr. Anne Remke

Westfälische Wilhelms-Universität Münster

Fachbereich Mathematik und Informatik

Arbeitsgruppe Sicherheitskritische Systeme

Münster, 16. März 2021

Abstract

Electrical power has become a crucial part of nearly every aspect of our daily lives. Consequentially it is crucial to secure the availability of electricity against all sorts of cyber-attacks. Since inter alia the use of common communication protocols and the possibility of remote maintenance via the internet made the controlling systems (Supervisory Control and Data Acquisition (SCADA)) of our electrical infrastructure more vulnerable than before, additional security effort is needed.
This thesis proposes a process-aware monitoring system that uses the data generated from the physical processes to check incoming sensor data and commands for consistency and with respect to certain security measures. By doing this, already on a local (substation) level, possible intrusion and manipulation may get detected in a distributed way. Additionally, further information gained from neighbouring field stations is taken into account to increase the system's state knowledge and yet remain distributed. The result of this thesis indicate that by using the neighbourhood data more attacks can be detected compared to previously approaches, which were completely local. This statement is shown not only from a formal perspective but also with a prototype implementation of the proposed distributed monitoring system and a matching testbed simulating an electrical grid.

Contents

List of Figures

List of Tables

List of Listings

Acronyms

CEM Centralized Energy Management.

CSV Comma-separated-values, file format.

CVE Common Vulnerabilities and Exposures.

DCS Distributed Control Systems.

DEM Decentralized Energy Management.

GDPR General Data Protection Regulation.

HMI Human-Machine-Interface.

HV High Voltage.

ICS Industrial Control System.

ICT Information and Communication Technology.

IDS Intrusion Detection System.

IPDS Intrusion Prevention and Detection System.

IPS Intrusion Prevention System.

IT Information Technology.

JSON JavaScript Object Notation, file format.

LV Low Voltage.

MTU Master Terminal Unit.

MV Medium Voltage.

PLC Programmable Logic Controller.

RTU Remote Terminal Unit.

SCADA Supervisory Control and Data Acquisition.

UML Unified Modeling Language.

1 Introduction

A modern life without the use of electricity is unthinkable. We use electricity in every aspect of our lives, for food processing, transport, health, work and of course digital free time activities. As a matter of fact, even the light in our homes needs and relys on electricity. Since a steady supply of energy is quite important for our daily lives, it only makes sense to ensure that it is well-protected. On the other hand, since the lives of many people depend on energy supply and unsteadiness can provoke large catastrophes in e.g. nuclear power plants, the energy infrastructure of a country or region becomes a critical target for (terrorist) attacks. Therefore safety and security of our energy supply, which is often controlled with Supervisory Control and Data Acquisition (SCADA) systems, is a highly relevant topic of public interest. To achieve overall safety and security for the electrical system of a single region is an enormous complex tasks with many different starting points. This thesis proposes a way of achieving better security with a focus on local field station monitoring and information exchange between close, neighbouring field stations.

1.1 Motivation

Historically, many electric power distribution systems were specially designed for their customers and supervised with proprietary protocols. This made it much more sophisticated to attack large parts of the infrastructure since every distribution system needed to be attacked separately. During the past years the products of electrical grid system distributors became more and more compatible with each other, more *off the shelf*-solutions got sold instead of individual solutions. Standard protocols and interfaces became popular for energy systems. One one hand this made the maintenance of field stations i.e. via the internet and collaboration of different corporations a lot easier. On the other hand, the vulnerability of the field stations for intruders increased extremely. In the Ukraine in the year 2015, about 230.000 people were left without electricity for multiple hours. Incidents like this show how critical and lethal attacks against our electric grids can become and how hard it is to fight their impacts[1, 2].

Generally speaking, a modern electrical grid consists of two major component types: The physical components (like power lines, transformers, power plants) and the SCADA network operating the electrical grid. As seen in the past, it is quite likely that in case of an attack, an intruder will abuse the

SCADA network for its attack. This could be done, for example as a *man-in-the-middle* attack which corrupts the communication between two parts of the SCADA network. The SCADA network would receive corrupted traffic which displays a wrong state of the physical system. Subsequent commands issued from the grid operator then may result in unwanted decisions, harming the physical system or, even worse, people.

To prevent such attacks on a SCADA network, an Intrusion Prevention and Detection System (IPDS) is needed to analyse the communication within the network. The IPDS is meant to detect manipulated communication and prevent unwanted commands from being executed by the electrical grid. There are many different approaches of IPDS for SCADA networks using whitelistening, encryption, anomaly detection or other insights gained e.g. from deep learning. However, a lot of these approaches focus on communication that seems wrong from a syntactical point of view rather than from a semantic point of view. This allows correctly formatted and inserted malicious commands to infiltrate the SCADA network and potentially hurt them in long terms, e.g. by slowly surpassing security thresholds for physical components. This thesis sheds light on one possible process-aware monitoring system, which can detect if the local communication implies a correctly issued, but unwanted situation for the SCADA network. Additionally, the proposed monitoring system implements a decentralized and distributed approach at each field station instead of a central entity monitoring the complete electrical grid at a central SCADA server. The proposed system is thought of a proof of concept work, which concentrates on a small part of an electrical grid and does not aim to deliver a complete secure (smart) grid operation but rather an addition to further security measures.

1.2 Objective and Research Questions

During this thesis the main objective is to develop an electrical grid model with appropriate security requirements and a corresponding implementation of a process-aware monitoring system. The core idea of the monitoring system is to detect sensory readings that are correctly send by local devices but are either implausible with respect to physical laws or circumvent defined security measures like thresholds. The model and the implementation are created with a distributed approach at each local field station of the electrical grid in mind and utilize additional data exchanges between neighbouring field stations. The proposed approach should be a proof of concept work, that the exchange of neighbourhood data helps to detect more anomalies compared to truly local approaches. Additionally, a test environment needs to be developed to explore the feasibility and limitations of both the implementation and the model. Therefore, the testbed requires the simulation of all relevant physical processes of the electrical grid.

This thesis bases on the work of Chromik et al. who studied local process-

aware monitoring systems for SCADA networks in electrical grids[3, 4, 5, 6]. As their work focused on truly local evaluations at the field stations, this thesis deepens and expands the research with the following research questions in mind:

- **Research Question 1:**
  How can (local) sensory readings from the underlying physical processes benefit the overall security of the SCADA network operating the electrical grid?
- **Research Question 2:**
  How can a hierarchical, distributed evaluation of safety requirements within an electrical grid be organized?
- **Research Question 3:**
  Which additional insights can be gained from using information from neighbouring field stations?

These three research questions will be addressed throughout all chapters of this thesis and reviewed again in the Conclusion.

1.3 Outline

This master thesis is subdivided into seven chapters. This first chapter presents the motivation and the underlying research questions that guide the thesis. In the second chapter the foundations of this work are presented. As a start a short introduction into Energy Systems and the historical and current challenges to them is given here. Additionally, SCADA systems are presented as well as threats and common corresponding countermeasures. Concluding the background chapter the last section constitutes previous incidents to electrical grids and their SCADA systems, security measures and related work in this field. The third chapter builds the theoretical groundwork for the later implementation. It presents an electrical grid model that focuses on a fragmentation into subgrids. Additionally physical and safety-related requirements will be described that can be checked against an instance of the model. This chapter also decipts how these requirements differ in the information scope needed for their evaluation and how decentralized exchange of information can aid the monitoring in total. The fourth chapter explains the development of a test environment for the proposed approach. It therefore gives insight into the chosen co-simulation framework as well as the scenarios that are used to mock attacks against the simulated electrical grid. Chapter 5 then shows the proposed implementation for a supervising monitoring system itself and its functioning. Further it is explained how the different information scopes will affect the evaluation of physical and safety-related requirements in the different parts of the monitoring system. The sixth chapter evaluates the insights

gained from the testbed in combination with the implementation. This Chapter explains which types of attack scenarios can be detected and the limitations of the proposed monitoring system and the testbed. Finally, the conclusions of this thesis are drawn in Chapter 7. There, the research questions are reviewed and the limitations and possibilities of future work are explored.

2 Background

To create a process-aware monitoring approach, proper knowledge of the basis is needed. In case of this thesis, the knowledge from two different fields are required, the world of electricity on one hand and Industrial Control System (ICS) with a focus on SCADA on the other hand. Hence, this chapter provides an introduction to both of these major dimensions and their most relevant aspects for this thesis. The approach for introducing both topics is similar. First Section 2.1 gives an introduction to Energy Systems. The subsections present the challenges caused by the *Energy Transition* in the past and also in current times and the typical components and structure of current electrical grids. To conclude this section, the threats to electrical grids and the desired characteristics in terms of security are explained. The second part of this background chapter, Section 2.2, focuses on SCADA networks. Starting, SCADA networks are classified as Industrial Control Systems and expounded with their typical components. Additionally, threats and desired security properties are shown for SCADA networks. Finally, in this Chapter, the focus shifts to SCADA networks controlling electrical grids in particular. Historical incidents and well-known counter-measures, including Intrusion Prevention and Detection System (IPDS), of these are illustrated to give an overview over the current state of the art. Closing, recent research and related work in the area of IPDS for SCADA networks in electrical grids is presented in Section 2.3.3.

2.1 Energy Systems

Electricity has always had a huge impact on our lives. From the first burning light bulbs and industrial manufacturing to space exploration and entertainment electronics, the access to consistent and predictable electricity has shaped how we interact with our surroundings. In the following subsections the historical and current *Energy Transition* is explained (Subsection 2.1.1) as well as some insights on how (modern) electrical grids are built (Subsection 2.1.2). Additionally, threats against the electrical infrastructure and the current principles of their counter-measures are pointed out in Subsection 2.1.3.

2.1.1 Energy Transition

The first electrical grids were formed in the 1880s, among them one in lower Manhattan, operated by Thomas Edison. His grid successfully distributed

110 V direct current (DC) to his customers. However, the difficulty of direct current already here became obvious: Because of the relative low voltage the distribution over long distances resulted in great currents and after all great losses. Westinghouse Electronics were the first to circumvent this problem with the usage of alternating current (AC) which could be easily transformed into higher or lower voltages, and therefore having lower losses over large distances. This so-called *war of currents* was eventually won by the alternating current system because of its efficient transport possibility. During the following years smaller grids were combined and finally led to nationwide electrification[7].

The historical development led to large and centralized grids. To the current day these grids produce power in large power plants far away from their actual customers and transform the current to high voltages to overcome large distances and be delivered at their distinct place. The system which underlays the electrical grids is made to allow power flow from the producer to the consumer and not the other way around. This design decision is a result of different technical constraints that have been crucial during past development and operation of electrical grids. One of these constraints has been, and currently is, the problematic storage of energy. As a result a centralized control-paradigm was needed to balance production and demand[8]. Until the mid-nineties this led in most western countries to a vertically integrated electricity supply chain, where state-owned or at least regulated companies owned and controlled everything from power plants to the distribution. Slowly the energy market began to open up and became more diverse, allowing companies to manage only small parts of the supply chain. This development, however, led to a more complicated billing compared to centralized supply management[9].

In current days the centralized control-paradigm is challenged more and more. One of the main reasons is the current global warming and the climate change, which was induced by humanity according to over 90% of climate scientists[10]. This awareness resulted in multiple (inter-)national treaties to minimize or stop the climate change, i.e. in Kyoto 1997 and in Paris 2015[11, 12]. On the one hand the use of fossil fuels and its following emission of carbon-dioxide (CO_2) and other greenhouse gases supported the increase of the global warming. On the other hand, fossil fuels are limited and partly present in political unstable regions. To minimize the effect of energy production on the global climate and not being dependent on finite fossil fuel resources, a shift to renewable energy production is aspired. This shift from fossil fuels to renewable resources is called *Energy Transition*.

When comparing renewable resources to fossil fuels, it can be seen that the power gets generated nearer to the place of usage than before. This does not only include private photovoltaic plants, but industrial solutions, too. Additionally, because of the diverse kinds of connectivity to the grid, the grid management is compounded compared to fossil fuels. Note that now much better metering material and sensory equipment is available to allow more detailed analysis of consumer data[9]. When the renewable energy generation

started, most providers integrated them with a "fit and forget" paradigm since the small amount of generated energy did not cause any major disturbances compared to the large amount of conventional produced energy. However, this gets more and more impractical since renewable energy production is starting to take over the conventional production. One of the biggest problems in this area is, that the original grid along with the centralized control-paradigm was only meant to allow energy flow in one direction. Now, private photovoltaic plants or heat generators start to add energy from the bottom up. Another large problem is, the weather dependency of renewable energy production and that the production cannot be increased like the energy production with fossil fuels. However, this level of control is needed to fulfill the demand of energy in the current system state. The fact that saving the already produced energy in e.g. batteries is remains unprofitable further hinders the growth of renewable energy resources[13].

The *Energy Transition* is expected to continue in the future and decrease the use of fossil fuels even more. At the same time, current and future research will develop methods to transport energy and better connect different regions. The latter, in form of fast in- and export of energy, it can help to aid with local bottlenecks in case of e.g. weather changes. Further, even more in-depth data will be available from local sensor devices and integrated Internet of Things (IoT) technology will lead to a more data-driven balancing approach. This approach can help to better control local micro grids which may be highly stressed by sudden consumption or production peaks. The usage of digital technologies to monitor and manage electricity is called a *smart grid* as defined by the IEA[14]. The combined information from generators, consumers, electricity market stakeholders monitoring devices as well as Information and Communication Technology (ICT) in a smart grid enables a *smart* utilizing of energy resources, to minimize costs and environmental impact without sacrificing the grids stability. Corresponding smart grid paradigms have been developed by the Electric Power Research Institute (EPRI), the Department of Energy (DoE) and the European Commission Task Force for Smart Grids.[15]

2.1.2 Electrical Grid Topologies and Operation

As discussed in the previous subsection the structure how electrical grids are built and operated is currently changing. In this chapter both, the historical and present principles are explained. The achieved knowledge on the grid topology is the foundation to build a reasonable mathematical representation of an electrical grid.

The conventional electrical grid can be split into three parts: production, transmission and distribution. The power is produced at large scale power plants. In the past, these plants were responsible to fulfill the complete de-

mand and were build far away from customers. Since energy storage has been (and still is) rather inefficient the energy needs to be transported to the customers quickly and with a desirable low loss. To achieve this a three level voltage system has been developed, which is graphically explained in figure 2.1. Near the power plants, which produce energy with (considerable) low voltage, the power is transformed to High Voltage (HV) to enable efficient transportation with low current. Because high voltage is dangerous to human life the last mile distribution is done with Medium Voltage (MV) and Low Voltage (LV)[9]. Note, that the distribution part of the system may consist of different combinations of MV and LV grid. The graphical distinction of a MV grid leading to the industry and a LV grid leading to a neighbourhood is meant exemplary.

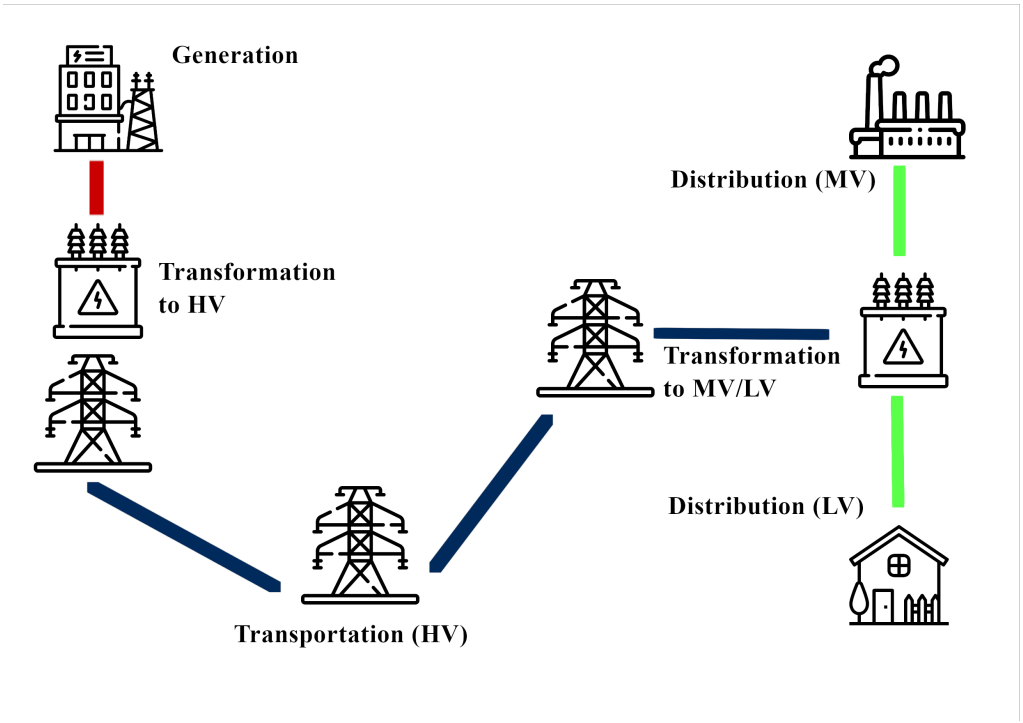

Figure 2.1: Schematic overview over the three different stages within an electrical grid: generation (red), transportation (blue) and distribution (green).

In this conventional grid topology the power was always flowing from the production downward to the distribution. Reverse energy flows were considered as a fault and it was common to check against this principle as a security mechanism[13].

For this sort of vertically organised grids, historically, Centralized Energy Management (CEM) has been used. In CEM, grid optimization is done by a central energy management system for the complete electrical grid. To implement such optimization a matching grid model is needed. The most common

structure for grid models in this case is a tree or acyclic graph. The grid can be split in multiple disjoint tree graphs, each representing e.g. a substation at the root and single customers at the leaves. In general the branches in such a tree model represent power lines and perform transformations between the different voltage levels. Nodes in this model represent different entities like generators, substations and customers[8]. To increase the reliability of the grid, small redundant connections are added to allow multiple alternatives in case of failure in certain branches. This design decisions then resulted in weakly meshed ring structures or loopy trees[16].

As explained above the historical development led to a breakup of the established vertical structure as small renewable energy resources began to feed in additional power at the bottom of the system. This additional power flow, which partly went against the preset direction, caused more and more local imbalances. To handle these imbalances the complete Energy Management shifted away from the centralized approach as the now occurring problems were local and needed quick reaction that did not necessarily affect a complete re-computation for the electrical grid. Following this development, the Decentralized Energy Management (DEM) paradigm was formed. DEM does not consist of a central calculation and optimization anymore but on smaller, more local balancing of small sections, so-called micro grids. The micro grids are LV networks and operate quite independently within themselves and only exchange power with the main grid in highly specified ways to not disturb the main grid[13]. Following this approach, more exchange between neighboured micro grids is desired to decrease local imbalances and therefore strengthening the meshing of the complete grid. It can be expected that the DEM approach will widen in the future to match the growing impact of renewable energy resources and making better use of local real-time data generated from smart grids as it could be done with one central optimization entity. Finally, the computational load of energy managment is decreased when using DEM compared to CEM as not all data needs to be analyzed at once and thus can be done quicker at the decentralized entities[17].

2.1.3 Security Policies and Threats

In the following paragraphs common security policies and general threats to electrical grids are explained. This gives a quick overview about the goals of electrical grid operators with respect to (cyber) security and therefore aids the specification of the monitoring system of this thesis. In delimitation to the section about SCADA security this section concentrates on (mostly) physical properties and characteristics of electrical grids.

Availability is indisputably the crucial property of an electrical grid as most consumers count on a direct and instantaneous connection to power. Therefore, every threat that targets the *availability* of an electrical grid can be considered a major fraud. Additionally the *integrity* and the *confidentiality* of

the electrical grid are of major importance, too. No electrical grid operator would want the data integrity to be breached and data modified by third parties as this might cause major problems while e.g. billing customers. On the other hand, the consumption data of the customers must remain confidential. Therefore, the common CIA triad (*confidentiality*, *integrity* and *availability*, highest importance to lowest) is inversed in order for electrical grids[15]. In general, energy theft has always been a large problem since the dimensions of an electrical grid and the constant surveillance of distributional power line is quite cost-intensive.

As an electrical grid always consists of physical components which are positioned outdoor most of the time, the environmental influences are additional threats alongside with targeted attacks. This includes weather conditions like storms, hurricanes and flooding as well as animals or plants living and growing next to physical components. Note that wearing of every physical components requires constant a maintenance according to the component type to prevent material failure which could lead to e.g. a violation of the *availability* of the grid.

2.2 SCADA networks

Electrical grids are controlled, among others, by SCADA networks. To analyse threats to electrical grids resulting from their SCADA networks and to develop appropriate counter measures, general insights into SCADA networks are required. In this Section an overview about SCADA networks as a part of Industrial Control System (ICS) (Subsection 2.2.1), about the structure and components of a typical SCADA network (Subsection 2.2.2) and about common security threats (Subsection 2.2.3) is given.

2.2.1 SCADA networks in Context of ICS

To manage the control of industrial processes so-called ICS were founded. ICS are meant to collect and monitor data from industrial networks. Generally speaking an industrial network sends two types of information: control-diagnostic and safety information. An ICS collects the control-diagnostic information, which can be sensor data or in- and outputs from control loops to monitor the system. Additionally it uses safety information to secure critical sections and keep the network reliable and real-time capable[18].

When compared to traditional Information Technology (IT) systems, Industrial Control Systems differ in multiple ways: The main goal of a ICS is to maintain the integrity and the availability of the underlying system which

are most likely continuous. For such systems, e.g. downtime for maintenance must be carefully scheduled. Unlike for conventional IT systems, the physical processes and components to which an ICS is connected to are highly complex and specific depending on their field of application. The underlying physical hardware layer often lacks additional resources for e.g. security measures and has special constraints regarding response times, memory space and the used protocols. Consequently, an ICS is used to be adapted matched to the specific purpose of the system[19].

In general, ICSs can be divided into roughly three types of systems: Programmable Logic Controller (PLC), Supervisory Control and Data Acquisition (SCADA) and Distributed Control Systems (DCS). In contrast to PLC, which are often solid state electronic devices, SCADA is a pure software layer above the hardware layer to operate industrial process. One of the main objectives of a SCADA network is it to offer a central Human-Machine-Interface (HMI) for the geographically distributed hardware. When comparing SCADA to DCS it can be viewed as an event-driven approach, where DCS follows a process-driven approach to operate the industrial process. Note, that in general it has to be assumed that a SCADA network may use data which cannot be trusted completely or might have been corrupted within the system[18].

2.2.2 General Structure of SCADA networks

As outlined above, SCADA networks build a software layer to control and target the special needs of industrial processes and geographical distributed physical processes. In the following paragraphs the architecture of SCADA networks in general will be explained, leading to a foundational understanding about how a SCADA network in context of an electrical grid works.

The central entity of a (centralized) SCADA network is the *control room*. The *control room* hosts first of all the HMI which allows the operator to operate the network manually. Most of the time, the SCADA network will be controlled automatically, but, if necessary, commands can be send over the HMI. In case of a SCADA network operating an electrical grid, the *control room* is likely to host additional systems to operate the *Energy Management System* (EMS). Furthermore, there is a *data acquisition server* to collect the data from the *field stations* over different (secured) communication channels. The control room is likely to have diverse security mechanisms, like a firewall for example to prevent malicious command to reach it over communication channels. The field stations host the actual physical process and the matching components. A *Remote Terminal Unit (RTU)* or PLCs operate different types of sensors and actuators to manage the physical process[6].

In case of a centralized SCADA network the control room can be seen as a Master Terminal Unit (MTU) which supervises the subordinated RTUs. Figure 2.2 exemplary represents a SCADA network with one MTU controlling two RTUs. In most instances there will be an additional backup MTU to

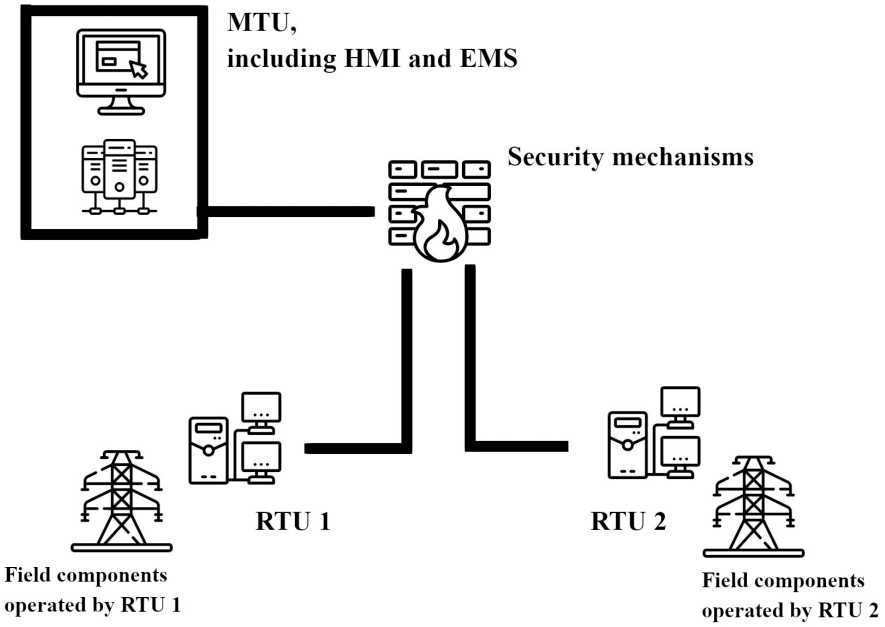

Figure 2.2: Schematic overview over a SCADA network consisting of one MTU and two RTUs.

overtake in case of an emergency or disparity at the normal MTU. In case of a decentralized SCADA network the role of the MTU is less important. Now, each RTU is more independently responsible for a special task or section and exchanges the needed information with its corresponding neighbours directly and not via a MTU which forwards the data to the designated receiver[20].

2.2.3 Threats and Security Measures

SCADA networks, as a part of ICS, can be viewed as a sort of IT system, have certain desired security policies. The following subsection gives an overview about general security principles which play a major role in context of SCADA networks security. Supplementary, common threats and distinct security measurements against those threats are presented. This later aids to classify threats and matching security measurements that target SCADA networks of electrical grids in particular. Overall, current research has shown over 100 000 Common Vulnerabilities and Exposures (CVE), not only for SCADA networks, but for ICS in general[21].

In principle, there are four major types of targeted attacks against SCADA networks which are commonly noted[22, 19, 23, 24]:

- **Injection of malicious code**: Entering malicious code into the SCADA network so that it will get executed.

- **DoS Attack**: Denial of Service attacks by over-flooding the target with requests to cause a temporary overload.
- **Spoofing**: Successfully identifying as a legitimate entity to the system and therefore gaining illegal access to it.
- **Replay of information and modification of data**: Sending historical and/or partly modified data within a legitimate request to e.g. conceal current sensory data.

As SCADA networks often operate critical infrastructure there are certain industry and governmental regulations and standards and additional corporate policies when it comes to SCADA security. One of those standards is the IEEE *Standard for SCADA and Automation Systems*[20]. According to this standard the following aspects need to be achieved for a secure SCADA network:

- **Access control**: Selected devices and information need to be protected against unauthorized access to the devices and information.
- **Use control**: Selected devices and information need to be protected against unauthorized operation of the devices and usage of the information.
- **Data integrity**: Data on selected communication channels needs to be protected against unauthorized changes.
- **Data confidentiality**: Data on selected communications needs to remain confidential and protected against eavesdropping.
- **Restrict data flow**: The data flow needs to be protected against publication of data to unauthorized sources.
- **Timely response to event**: When a security violation happens, the proper authority needs to be notified, the evidence of the violation saved and the corrective action needs to be automatically started in case of critical and safety critical situations.
- **Network resource availability**: To protect against *Denial of Service*-attacks (DoS) in particular, the availability of network resources needs to be ensured.

Similar security policies can be found in most regulations and standards concerning Security in SCADA networks. Depending on the application of a SCADA network however, the priority of this policies might change[18, 22].

Historically speaking, a big security paradigm used for SCADA networks and especially for the used communication protocols was "security through obscurity"[25]. Note, that from the beginnings of SCADA networks in the

1970s up to the early 2000s [21] most SCADA networks used proprietary protocols which were not publicly known. Consequently, searching for faults in those protocols was not practical for most adversaries. With the opening of SCADA networks to the internet and the standardization of protocols, however, this became a major problem. Starting with small attacks from "teenagers" and "mischievous adversaries" against corporate computers, networks and the HMI of the SCADA networks, directly targeted cyber-terror attacks against the SCADA infrastructure became relevant[18]. With the increasing importance of smart grids, inter-connectivity and data exchange, it can be expected that the problem is about to enlarge as soon as faults in commonly used protocols are discovered[22].

An example for such a problematic protocol is the Modbus/TCP protocol. Modbus/TCP is commonly used to connect to remote RTUs in the oil and gas sector as well as for power distribution[26]. An intruder can easily read information which are sent over Modbus and, depending also on, easily modify information. Even tough security extensions have been developed for Modbus, it is often difficult to integrate them into already existing systems[27]. As an additional problem, the matching remote interfaces and gateways to e.g. the Modbus Interface are often poorly secured[21].

Note that not only protocols used in SCADA networks can have security breaches, but also security breaches in e.g. Windows or other more common software used within the SCADA network may lead to critical problems in the infrastructure[21]. Finally, social attacks and attacks against the physical layer are of importance, too. Social attacks mean e.g. attacks via social engineering which target workers of the involved corporations. Security measures against physical layer attacks mean such trivial things as doors, walls and fences guarding the physical components of SCADA networks, especially at the fieldbus stations. Such attacks could not only lead to dramatical physical damage directly (e.g. upon explosion of a field station) or lead to deeper attacks, e.g. manipulating sensors and infiltrating incorrect measurements into the system.

To prevent and detect attacks on SCADA networks there are different sorts of automated software systems available: Intrusion Detection System (IDS), Intrusion Prevention System (IPS) and Intrusion Prevention and Detection System (IPDS). As the names already suggest they follow different approaches to achieve a security while monitoring the (complete) SCADA network. However, even within the same group of software systems they may follow different prevention/detection models. In the following a short overview over such models is given.

- **signature-based (also knowledge-based)**
  Signature- or knowledge-based software uses a database with known incidents that happened and were detected in the past. It is checked whether the current situation matches such an historical attack, and if this is the

case, an alert is triggered. Of course, the underlying database must be updated periodically[28].

- **behaviour-based (also anomaly-based)**
  Behaviour- or anomaly-based software always compares the current state to an ideal, standard state. If deviations to the standard state are detected, an alert is triggered. Therefore, such software is able to find zero-day attacks but it is often hard to train such software in a way that it prevents to many false positives[15, 28].

- **specification-based**
  Specification-based software works with a well tailored set of allowed actions using e.g. a protocol. Deviations from those actions are considered a fault, and an alert gets triggered. Even tough they can easily spot those deviations, most of the time they are only able to detect a small section of attacks according to [28, 29].

In this thesis the proposed monitoring system follows the idea of a sort of behaviour-based approach, where the input data is checked against pre-defined requirements to decide if the data is plausible or not.

2.3 SCADA networks for Electrical Grids

In the previous sections Electrical Grids and SCADA networks were examined separately. This section will go into detail regarding to the combination of both. To highlight the importance of security measures in this context, known and major impacts of the past years are presented (Subsection 2.3.1). Even though there are many different attacks on electrical grids, the focus will be on those who targeted the SCADA network in particular or resulted from vulnerabilities in the SCADA network. Following in this section, proven security measure and known obstacles for SCADA networks in electrical grids are pointed out (Subsection 2.3.2). This gives an overview about the variety of measures that their entirety aid to achieve a secure operation of an electrical grid. Finally, and concluding this chapter, in Subsection 2.3.3 recent research that is in close proximity to the topics considered in this thesis is presented.

2.3.1 Known incidents to Electrical Grids

Before discussing any well-known incidents to electrical grids can be discussed in greater deal, it has to be clear, that in such cases, incidents are often a delicate matter to the targeted corporations. As the security breaches might

have misused internal information or published internal matter and as such an incident in general betrays the trust third parties have in the corporation, the corporation has an interest not to publish all the information related to the incident. Therefore, helpful insights, even to foiled attacks, might remain internal and cannot be investigated by e.g. researchers.

Generally speaking there is multiple statistical research regarding ICS exploits on the one hand and electrical or smart grid exploits on the other[15, 19, 21]. To give an indication about different approaches of successful incidents, five historical incidents who targeted i.a. the SCADA network of electrical grids in particular are presented. These incidents can be seen as stereotypical incidents showing diverse entry points and resulting effects:

- **Slammer worm**
  The *slammer* worm is a good example for an attack that originally targeted proprietary software (in this case Microsoft SQL) that is used within the SCADA network. During the year 2003 a slammer worm infected roughly 75000 machines, one case being a nuclear power plant in Ohio, United States. The worm, which executed DoS attacks caused a Denial of Service at the process computer of power plants for about six hours and the safety-related system for almost five hours.[30]

- **Aurora**
  In 2007 the Idaho National Laboratory exploited the so-called Aurora vulnerability which focuses directly on electric generators. While using the vulnerability, access could be gained to the control network via a diesel generator and multiple commands to open and close circuit breakers could be dispatched, eventually leading to an explosion of the generator itself. Therefore the Aurora vulnerability is a vulnerability that directly leads to physical damage to the components[31, 19].

- **Stuxnet**
  The stuxnet worm is another good example for a malicious software, originally targeting Windows systems and Siemens industrial software, which spread fastly in the energy, oil and gas industry in June 2010. Additionally, the stuxnet worm infected a nuclear power plant in Iran, probably via an infected USB stick. If it had not been stopped, this incident eventually would have led to a nuclear meltdown in that power plant[32, 33].

- **Dragonfly/HAVEX**
  Starting in 2014 a campaign called Dragonfly targeted different ICSs, especially in electric power systems in multiple countries (US, Germany, France, Italy, Spain, Portugal and Turkey). Using the HAVEX malware it spread over phishing e-mails and exploited all devices in the ICS network. Using a weak point in a protocol, a lot of information about

the network structure was gained. The Dragonfly campaign caused no (direct) physical damage itself[34, 35].

- **Cyber attack against Ukrainian substation**
  In December 2015 a large-scale attack targeted several Ukrainian substations and led to a power outages for more than 225,000 customers. The attack used the BlackEnergy 3 malware to gain access to the SCADA networks and the corporate networks of the power companies, manipulated commands, disconnected power lines and delayed the regaining of control by destroying the SCADA servers using the KillDisk malware. It is quite likely that the attackers gained legitimate credentials to the system as a first step via social engineering[1, 2].

2.3.2 Security in the combined context

In general, all security measures that are taken in the setting of SCADA networks in electrical grids, must take both origins into account. Of course, because of the complexity of the systems no single piece of software will be able to guarantee a secure operation of the complete system but can only observe a smaller part. Yet the physical and the cyber features of the various components must be taken into account even when developing security mechanisms for only a (comparatively) small section. White-listening approaches can detect unusual network traffic in the SCADA network really well, but cannot detect well-formatted network packets that could harm the physical components nevertheless. That is why a situational and process awareness is required, to empathize the interaction of the overall system, especially since most critical security requirements focus either on the power substation or on the SCADA network[15, 23]. In the past, SCADA networks were mostly used to operate the HV parts of an electrical grid. However, with the ongoing *Energy Transition* and the upcome of DEM in MV or LV levels, SCADA networks are more and more deployed on this levels, too.

Within this thesis, a process-aware approach is pursued. As already mentioned, a process-aware approach cannot protect against all kinds of attacks, therefore additional and complementing software for e.g. DoS-attacks and brute force attacks is always necessary. Note, that historical security measures to protect the electrical grid against environmental influences (e.g.weather, fire, earthquakes and animals) and analog human interference (e.g. burglary, destruction), which could be walls, fences and distance to other buildings are still relevant and need to remain in place.

As mentioned above, two large security problems in this sector are **A)** the increased interconnectivity in industrial applications, which not only increase the number of possible entry points, but also increase the interest for large attackers into specific parts of the application and **B)** the continuous use of protocols for communication with known security gaps like missing authenti-

cation and access control mechanisms[21, 15]. Matching measures to this two problems are of course the securing of the individual parts of the complete system and the exchange or adaption of currently used protocols. Note, that the latter is often rather cost-intensive as hardware might be constructed only for a specific protocol or unable to adapt to newer mechanisms.

When developing security measures it is important to evaluate which skills and possibilities intruders had in the past and are able to use now. In general, the so-called *intruder model* for SCADA networks in electrical grids includes the following skill set:

- The intruder is able to successfully authenticate as a legitimate communication partner.
- The intruder has knowledge about the used protocols and processes.
- The intruder is able to interrupt communication between two parties.
- The intruder is able to eavesdrop on communication channels, manipulate and to replay historically sent messages.
- The intruder is able to sent new, legitimate messages.

To prevent intruders from manipulating the SCADA network operating the electrical grid matching IPDS can be helpful. Concerning the approach of the IPDS, each of the presented approaches in subsection 2.2.3 has its advantages and disadvantages for electrical grids. Panagiotis et al. developed a good statistical overview which types of IPDS cover which part in a smart grid ecosystem and give a comprehensive overview over current trends[15]. The best choice for a matching IPDS highly depends on the architecture of the operated network and the additional security measures. In most real world scenarios, combinations of e.g. signature-based and specification-based IPDS are used.

Finally, and as a result of historical incidents, there are different national and international regulations and standards for operating SCADA networks with security in mind.

2.3.3 Recent Research and Related Work

In this concluding subsection for the chapter, recent research and recently gained scientific insights on how to achieve security in SCADA networks are presented. The focus is mainly on IDS and process-aware approaches, but does not completely neglect further ones.

Following the awareness gained in Subsection 2.1.1 that the conventional way how grids were planned and optimized is currently changing, recent research has shown a strong link between the risk to the network and its structure. This field has researched by [36]. For example, they found that scale-free networks with a large connectivity between strongly heterogeneous nodes are robust to random failure but more vulnerable to direct attacks while random graphs have been shown to be equally vulnerable to both[36].

Since traffic in SCADA networks is often assumed to be stable and periodic, there are many behaviour-based approaches as they can easily define the "standard" traffic[37, 38]. A big disadvantage of these *whitelisting* approaches is, that they cannot detect legitimate commands and messages that are sent by an intruder. This problem is addressed by further research which targets the protocols and packet inspections[39, 40, 41]. However, these approaches only rarely take the physical dimension of the underlying electrical grid into account.

Note, that even tough attackers often misuse weak points that are grounded in the physical dimension or, at least of semantical nature within the communication, this approach is less researched. One of the first researchers to take the physical dimension into account is [42], which follows a specification-based approach. Further research using sensor data from substation was done by [43, 44, 45]. An issue while developing new security measures for SCADA networks in electrical grids will always be that is it hard to test them directly on real environment and they are mostly tested in testbeds or in simulations only. Further information and a comparision between multiple testbed approaches is given in 4.2.1. More semantically founded are the works of [46, 47] which take the sequence-order of sent packages into account to detect malicious communication.

Recently, Chromik et al. [3, 4, 5, 6] developed a process-aware approach which exploits the knowledge with respect to the physical infrastructure of the control system to check control commands and sensory data against physical requirements (to prove that they are plausible in the first place) and against security requirements (to check that they do not violate desired properties of the system). However, their work mostly considers only one substation and its local commands. In this thesis their approach will be further developed to not only do this on a local basis but check commands between neighbouring substations.

3 Methodology

This chapter focuses on the theoretical groundwork of this thesis. It extends a formal model in Section 3.1 which can be used to describe a complete electrical grid with a hierarchical approach in mind. The hierarchical approach leads to splitting up the grid into smaller subgrids as single entities. The idea of splitting the electrical grid into smaller subgrids later aid the distributed monitoring approach and is introduced in Section 3.1.1. The individual physical components which an electrical grid may be part of are described in Section 3.1.2. To implement such a monitoring system cornerstones are needed to decide whether a system (state) is regarded safe. Such possible cornerstones in form of requirements are discussed in Section 3.2. There are various physical and safety-related requirements presented that can indicate whether the current system state should be considered plausible and safely operated. The third section of the Chapter 3.3 then outlines the effect of the hierarchical structure of the grid regarding to the evaluation of these cornerstones. Primarily, this structure impacts the level of available information from different points of views. Consequently it is explained in this section which requirements can be evaluated in the different data scopes as this largely affects the architecture of the monitoring system. The last section finally gives an outlook on how monitoring can be constructed which supervises the given physical and safety-related requirements regarding to an instance of an electric grid. This algorithm utilizes the insight gained from the previous scope evaluation to ensure an reliable distributed monitoring.

3.1 Formal Model of an electrical grid

3.1.1 The idea behind the hierarchical approach

In this section a formal model is designed to unambiguously describe a complete electrical grid, its structure and its physical components. The model should support the operational development of the grid over time and show how it evolves. This includes for example data sets gained from sensors in a certain time interval. Additionally, it should support the division of the whole electrical grid into smaller subgrids. This division will aid to implement a distributed approach to the monitoring. The subgrids should have (almost) no restrictions regarding their size or content-related structure. The region between two subgrids containing shared components is called a *border region* in the scope of this thesis. Exemplary, figure 3.1 shows an electrical grid topology

containing three subgrids $\Omega_{0\text{-}2}$ and three *border regions* between each subgrid pair. Within this thesis, substations (a RTU and its supervised components)

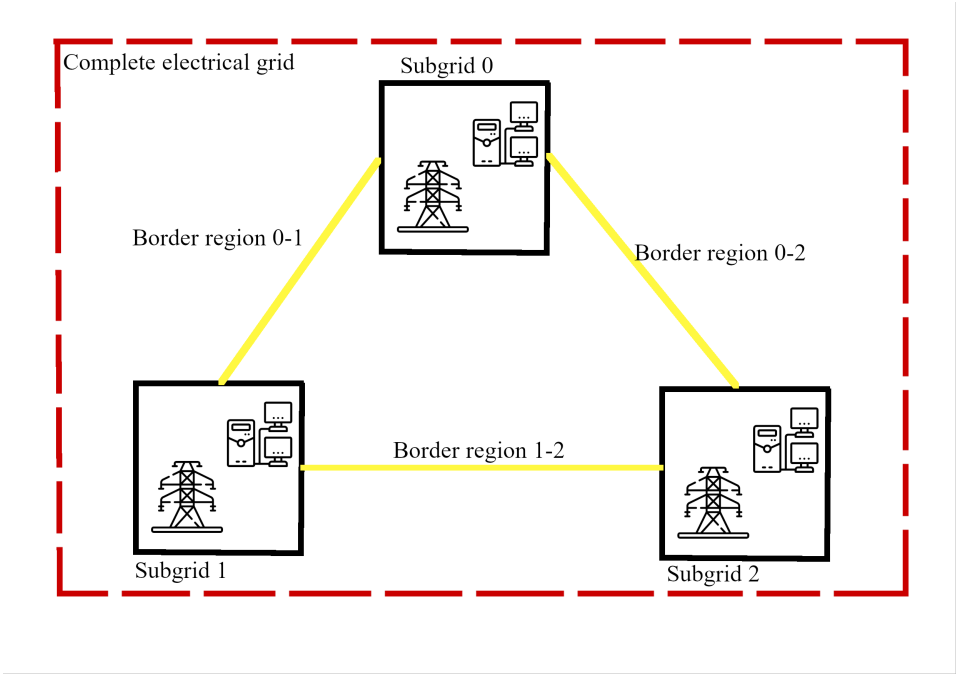

Figure 3.1: An example grid Ω (dashed, red box) with three subgrids $\Omega_{0\text{-}2}$ (black box) and their connecting border regions (yellow).

will be used as subgrids, but the model itself should not be limited to this use. This thesis is using a pilot run using an electrical grid instance with two distinguished subgrids, but again, the model itself is not limited to a specific number of subgrids.

The idea of splitting the complete electrical grid is meant to help analysing and monitoring only small (and mostly independent) parts in a distributed way. Secondly, the interaction between smaller subparts can be evaluated independently from the rest of the system. This can e.g. be done by evaluating each binary relation between two subgrids sharing a connecting edge. Therefore it has to be possible firstly to evaluate each subpart on its own, to evaluate its interaction with its neighbours and lastly to evaluate the complete electricity grid.

The model has to be completely independent from a specific (programming) language or testbed. Both, the following formal model and the in Section 3.2 defined requirements are largely based on the work of Chromik[3, 4, 5, 6].

3.1.2 The hierarchical electrical grid model

The formal model unambiguously describes an electrical grid Ω and how it evolves over time. The model considers both *static* and *variable* parts of the grid. The *static* properties describe the *topology* of the system (its architecture) while the *dynamic* parts describe the *state* of the system it is currently in. An instance of the model Ω consists of different physical components. Just like the system itself, the physical components have *static* properties describing their topology (e.g. how they are connected to other components) and *variable* properties describing their current state (e.g. the measured current).

An electrical grid Ω can be described as the following tuple:

$$\Omega = (\mathcal{P}, \mathcal{B}, \mathcal{T}, \mathcal{L}, \mathcal{S}, \mathcal{M}, \mathcal{F}, \mathcal{R}, \mathcal{K}),$$

with the objects defined as follows:

$\mathcal{P}$ = set of power consumers $\mathcal{P}^C$ and power producers $\mathcal{P}^G$, $\mathcal{P} = \mathcal{P}^C \cup \mathcal{P}^G$
$\mathcal{B}$ = set of buses
$\mathcal{T}$ = set of transformers
$\mathcal{L}$ = set of power lines
$\mathcal{S}$ = set of switches
$\mathcal{M}$ = set of meters
$\mathcal{F}$ = set of fuses
$\mathcal{R}$ = set of protective relays
$\mathcal{K}$ = set of interlocks.

with each of these sets contains all correspondent components that are present in the electrical grid Ω. Note that not every instance of Ω contains each of these component types. For instance, there may be a specially defined electrical grid without transformers.

In the same way, a subgrid Ω_i of Ω can be described as the following tuple:

$$\Omega_i = (\mathcal{P}_i, \mathcal{B}_i, \mathcal{T}_i, \mathcal{L}_i, \mathcal{S}_i, \mathcal{M}_i, \mathcal{F}_i, \mathcal{R}_i, \mathcal{K}_i),$$

with the objects defined as follows:

$\mathcal{P}_i$ = a subset of $\mathcal{P}$, consisting of $\mathcal{P}_i = \mathcal{P}_i^C \cup \mathcal{P}_i^G$
$\mathcal{B}_i$ = a subset of $\mathcal{B}$
$\mathcal{T}_i$ = a subset of $\mathcal{T}$
$\mathcal{L}_i$ = a subset of $\mathcal{L}$
$\mathcal{S}_i$ = a subset of $\mathcal{S}$
$\mathcal{M}_i$ = a subset of $\mathcal{M}$
$\mathcal{F}_i$ = a subset of $\mathcal{F}$
$\mathcal{R}_i$ = a subset of $\mathcal{R}$
$\mathcal{K}_i$ = a subset of $\mathcal{K}$.

So that consequently one could say that $\mathcal{B}$ consists of all subsets of buses present in subgrids, $\mathcal{B} = \bigcup_i \mathcal{B}_i$ and equivalent for the other component types. In the case that an electrical grid consists only of one subgrid this would be trivial. However, as already mentioned in 3.1.1 there are no strict rules on how to architect one subgrid Ω_i. Therefore, the size of the defined subgrids within an electrical grid, may vary. As the previously defined complete electrical grid Ω not necessarily needs to contain elements of every component type, the subgrids likewise do not need to always have e.g. a transformer. In case of a non-trivial splitting of the electrical grid Ω into two or more subgrids it must be ensured that none of the following component types are assigned to more than one of the subgrids.

For two pairwise different subgrids Ω_i and Ω_j, with i $\neq$ j it must hold that:

$$\mathcal{E}_i \cap \mathcal{E}_j = \emptyset \text{ for } \mathcal{E} \in \{\mathcal{P}, \mathcal{B}, \mathcal{T}, \mathcal{S}, \mathcal{M}, \mathcal{F}, \mathcal{R}, \mathcal{K}\}.$$

Noticeably this does not hold for power lines $\mathcal{L}$ as they may be shared as a connecting component between the two subgrids. In addition to the specific properties of every component type which are described later in 3.1.2, every component type has a *static* property called *origin*. This property tracks the belonging of the component to its subgrid Ω_i.

Again, for every component type except for power lines $\mathcal{L}$ the *origin* property is defined as follows:
For an element e of component type $\mathcal{E} \in \{\mathcal{P}, \mathcal{B}, \mathcal{T}, \mathcal{S}, \mathcal{M}, \mathcal{F}, \mathcal{R}, \mathcal{K}\}$ which belongs to the subgrid Ω_i and subset $\mathcal{E}_i$:

$$\text{e.or} = \text{i}.$$

By defining the property *origin* like this, it is possible to define a subgrid Ω_i from a bottom-up approach consisting of every component with e.or = i. The *origin* property of power lines $\mathcal{L}$ will be explained in their distinguished component description below.

The distinct description of the *static* properties of all components is called the *topology* of the electrical grid. Table 3.1 gives an overview of all components and their *static* and *dynamic* properties. The complete *topology* is able to show the architecture of the electrical grid and how all components are connected. This includes e.g. the *origin* property to determine to which subgrid each component belongs to. The tuple of all sets of components together with the *variable* properties and their currents values defines the system *state*

In the following calligraphic capital letters always denote the set of components in the complete electrical grid and a calligraphic capital letter with an index a subset of components in a subgrid. A normal capital letter indicates one element of a (sub-)set of components.

Component	Property (type)	Symbol
Power generators $\mathcal{P}^G$	origin (static)	P^G.or
	location (static)	P^G.pos
	power value (dynamic)	P^G.pv
Power consumers $\mathcal{P}^C$	origin (static)	P^C.or
	location (static)	P^C.pos
	power value (dynamic)	P^C.pv
Buses $\mathcal{B}$	origin (static)	B.or
	incoming power lines (static)	B.in
	outgoing power lines (static)	B.out
Transformers $\mathcal{T}$	origin (static)	T.or
	incoming power line (static)	T.in
	outgoing power line (static)	T.out
	transformer rate (dynamic)	T.r
	tap position (dynamic)	T.p
Power Lines $\mathcal{L}$	origin (static)	L.or
	maximum current (static)	L.I_{max}
	reference voltage (static)	L.V_{ref}
Switches $\mathcal{S}$	origin (static)	S.or
	position (static)	S.pos
	current state (dynamic)	S.st
Meters $\mathcal{M}$	origin (static)	M.or
	location (static)	M.pos
	set point (static)	(M.I_{sp}, M.V_{sp})
	measured current (dynamic)	M.I
	measured voltage (dynamic)	M.V
Fuses $\mathcal{F}$	origin (static)	F.or
	position (static)	F.pos
	state (dynamic)	F.st
Protective Relays $\mathcal{R}$	origin (static)	R.or
	threshold (static)	R.I_{max}
	position (static)	R.pos
Interlocks (static) $\mathcal{K}_{\text{static}}$	origin (static)	I.or
	subset of switches (static)	K.S
	minimum switch count (static)	K.CS_{min}
Interlocks (dynamic) $\mathcal{K}_{\text{dynamic}}$	origin (static)	I.or
	subset of switches (static)	K.S
	minimum current capacity (static)	K.I_{min}

Table 3.1: Physical components of an electrical grid Ω and their *static* and *dynamic* properties

Component descriptions

Power consumers and producers $\mathcal{P} = \mathcal{P}^C \cup \mathcal{P}^G$

Power producers generate the power that is distributed by the electrical grid. A power producers could be e.g. photovoltaic panels or power plants. The power consumers on the other hand are households in a general sense. However, in the context of smart grids there may be consumer households that also produce power.

The power produced/consumed by a power producer or consumer is denoted as $P_i.\text{pv} \geq 0$ (in case of a consumer ≤ 0). In addition to their *dynamic* power value, consumers and producer each have a *static* location in relation to the power line they are connected to. Their position is denoted as $P_i.\text{pos}$ which specifies the power line they are connected to (i.e. $P_i.\text{pos}{=}L_j$ means that the producer or consumer is connected to power line L_j). Note that the power line must belong to the same subgrid as the power producer/consumer.

Power producers and consumer can only be assigned to one subgrid. However, large power consumers, like important data centers, may have to be connected to two or more independent power sources, therefore they are connecting to two or more subgrids. In this case this power consumer, which is always an endpoint in the topology of the electrical grid, will be recorded twice (or more times), one time for each subgrid it is attached to.

Buses $\mathcal{B}$

An electrical bus (or bus bar) connects incoming power lines with outgoing power lines. This can help distributing power between different power lines or overcoming energy levels.
Therefore a bus B_i only has the two *static* properties, the incoming power lines $B_i.\text{in} = \mathcal{L}_{\text{in}}$ and the outgoing power lines $B_j.\text{out} = \mathcal{L}_{\text{out}}$. Both $\mathcal{L}_{\text{in}}$, $\mathcal{L}_{\text{out}}$ are subsets of the power lines $\mathcal{L}_i$ of subgrid Ω_i.

Transformers $\mathcal{T}$

A transformers T_i helps connecting parts of an electrical grid that operate at different voltages. It is connected to the grid by an incoming $T_i.\text{in}$ and a outgoing $T_i.\text{out}$ power line (i.e. $T_i.\text{in}{=}L_j$ and $T_i.\text{out} = L_k$ would mean that T_i is connected to the incoming power line L_j and the outgoing power line L_k). Hence, both power lines L_j, L_k must be from the same subgrid as T_i. In addition to these *static* properties a transformer has a *dynamic* transformer rate and a tap position. The transformer rate $T_i.\text{r}$ defines the voltage ratio in which voltage is transformed and the tap position $T_i.\text{p}$. The tap position allows changing the voltage ratio in distinct steps.

Power Lines $\mathcal{L}$

Power lines connect different components of an electrical grid like power consumer and producer with buses, transformers and with each other. Thereof

power lines $\mathcal{L}$ can be defined as follows:

$$\mathcal{L} \subseteq ((\mathcal{P} \times \mathcal{B}) \cup (\mathcal{T} \times \mathcal{B}) \cup (\mathcal{B} \times \mathcal{B}) \cup (\mathcal{B} \times \mathcal{T}) \cup (\mathcal{B} \times \mathcal{P})).$$

As power lines can connect components that belong to two different subgrids, power lines itself can belong to either one or two subgrids accordingly. However, a power line cannot be assigned to more than two subgrids. Therefore the *static* property *origin* $L_i.or$ of a power line L_i which connects a component E_j with a component E_k is a tuple containing the origin of its two connecting components:

$$L_i.\text{or} = (E_j.\text{or}, E_k.\text{or}).$$

Each power line has the property of a maximum current $L_i.I_{\max}$ it can endure to prevent damage to it. Additionally, each power line has a reference voltage $L_i.V_{\text{ref}}$ which ensures that the power line stays in its voltage boundaries depending on the placement in the electrical grid.

Switches $\mathcal{S}$

A switch in the context of this model is a component to connect or disconnect a power line from a bus. The position of a switch S_i is denoted in regard to the power line L_j and the bus B_k it is attached to. The power line L_j and the bus B_k must belong to the same subgrid as S_i. Therefore the *static* property $S_i.\text{pos}$ is defined as $S_i.\text{pos} := L_j.B_k$. The current state of the switch, meaning the *dynamic* property whether it is currently opened or closed, is denoted as $S_i.\text{st} \in \{0,1\}$. An *open* switch ($S_i.\text{st} = 0$) indicates that the power line is disconnected and a *closed* switch ($S_i.\text{st} = 1$) indicates that it is connected.

Meters $\mathcal{M}$

Meters are basically the sensors connected to power lines measuring the current ($M_i.\text{I}$) on that power line and the voltage ($M_i.\text{V}$) between the power line and the ground. Additionally each meter has a pre-defined set point. This set point should not be exceeded by any measurement and in case it is exceeded an alert needs to be triggered. Consequently, a set point of a meter is defined as a *static* tuple of a current and a voltage value ($M_i.I_{\text{sp}}$, $M_i.V_{\text{sp}}$). The location of a meter is another *static* property, is stored in relation to location on the power line and the next bus, just as for switches. The power line L_j and the bus B_k must belong to the same subgrid as M_i. If a Meter M_i is located on power line L_j and at the side of bus B_k the meter M_i has the position $M_i.\text{pos} = L_j.B_k$. Meters that are attached to a power line, which is connecting two subgrids, must be part of one of these subgrids. Therefore, if a power line connects two subgrids, the meters on that power line must be splitted so that each subgrid has at least one meter assigned.

Fuses $\mathcal{F}$
A fuse can be seen as a one-way switch. Once it is exposed to an overcurrent, it melts and cannot be turned back on, only replaced. A fuse F_i therefore has a *dynamic* state, just like a switch, where $F_i.\text{st} = 0$ describes a disconnected (melted) fuse and $F_i.\text{st} = 1$ a connected one. Again the *static* position of a fuse is described with its location on the power line L_j and the side of the bus B_k, i.e. $F_i.\text{pos} = L_j.B_k$. The power line L_j and the bus B_k must belong to the same subgrid as F_i.

Protective Relays $\mathcal{R}$
Protective relays are controllers for opening and closing switches. Once a certain *static* threshold $R_i.I_{\max}$ is exceeded it will mechanically or digitally open the switch to prevent overcurrent. The *static* position of a relay $R_i.\text{pos}$ is denoted in relation to its switch S_j, i.e. $R_i.\text{S} = S_j$. The switch S_j must belong to the same subgrid as the protective relay R_i.

Interlocks $\mathcal{K}$
Interlocks in this context are a locking mechanism for switches. An interlock K_i attached to a subset of switches $\mathcal{S}_j$ in the same subgrid as K_i, i.e. $\mathcal{K}_i.\text{S} = \mathcal{S}_j$, ensures that either a minimum count $\mathcal{K}_i.CS_{\min}$ of switches is always closed (in case of a static interlock) or a minimal current capacity $\mathcal{K}_i.I_{\min}$ is always ensured (in case of a dynamic interlock). The interlock K_i has to belong to the same subgrid as its switches $\mathcal{S}_j$.

3.2 Physical and Safety Requirements

In the following section a selection of possible physical and safety-related requirements will be presented. This will be done in two different subsection as both sorts of requirements cover different desired properties of a grid.

In general these requirements can be viewed as cornerstones to ensure that an instance of the model presented above is currently in a safe and plausible state. They ensure that the measured data is, from a physical point of view, consistent. While the physical requirements are created from given physical laws, the safety requirements have to ensure that desired properties of the electrical grid are fulfilled.

The following requirements are oriented on the physical constraints and safety requirements defined by Chromik [4, 5, 6]. The requirements are created with a generic electrical grid in mind which consists of the components as defined in Section 3.1. In case of a specific electrical grid or a special subgrid it could be very useful to define additional requirements that target the special needs and circumstances of that subgrid. This could be for instance the case if a subgrid contains a certain crucial power plant or if an electrical vehicle and its docking station is present in the grid. Finally, the requirements could also be extended by newer laws or standards, e.g. from the European Committee for Electronically Standardisation.

3.2.1 Physical Requirements

The physical requirements aim to ensure that the measured state of the electrical grid is feasible from a physical point of view, meaning that it does not violate the physical laws in place. If a physical requirement is violated by one of the measured data sets in the electrical grid, this means that the measurements show a state of the electrical grid that is physically not possible. This would indicate that either one of the measurements is false due to e.g. technical issues, like inaccuracy or a malfunctioning sensor, or that the measurements were manipulated or changed. Note that the requirements are formulated assuming perfect accurate measurement. However this might not always be the case due to measurement inaccuracy. Therefore checking requirements they must be adapted to this inaccuracy to prevent false conclusions. In the following the considered physical requirements that a grid must always fulfill are explained in more detail.

Requirement 1
The first requirement is an adaption of Kirchhoff's current law. The first of Kirchhoff's circuit laws says that at a node in an electrical grid the sum of the incoming currents must equal the sum of the outgoing currents. In the considered electrical grid this law must always hold for every bus. Thus, for a bus it has to be checked if the incoming current on the bus matches the outgoing current:

$$\textbf{REQ 1: } \forall\ B_i \in \mathcal{B}\text{: } \textstyle\sum_{L_j \in B_i.in} L_j.B_i.\text{M.I} = \sum_{L_j \in B_i.out} L_j.B_i.\text{M.I.}$$

Requirement 2
A second, and closely connected requirement, is that all voltages reported at the bus must always be equal:

$$\textbf{REQ 2: } \forall\ B_i \in \mathcal{B}, \forall\ L_j,\ L_k \in B_i.\text{in} \cup B_i.\text{out}\text{: } L_j.B_i.\text{M.V} = L_k.B_i.\text{M.V.}$$

Requirement 3
The third requirement focuses on power lines that have a switch. If the switch is in its state *open*, no current may be measured by the meters on the power line:

$$\textbf{REQ 3: } \forall\ L_i \in \mathcal{L}\ \exists\ S_j \in L_i.\text{S}\text{: } S_j = 0 \rightarrow \forall\ M_k \in L_i.\text{M} : M_k.\text{I} = 0.$$

Requirement 4
The fourth requirement ensures that the measured voltage and the measured current is the same for all measurements on a given power line:

$$\textbf{REQ 4: } \forall\ L_i \in \mathcal{L}\ \forall\ M_j,\ M_k \in L_i.\text{M}\text{: } M_j.\text{I} = M_k.\text{I} \wedge M_j.\text{V} = M_k.\text{V.}$$

Requirement 5
To verify that the data measured at power generators and consumers is feasible, the two requirements REQ 5a (for power generators) and REQ 5b (for power consumers) are placed. These requirements check that the classical power formula $\text{P} = \text{I} \cdot \text{V}$ holds:

$$\begin{aligned}
&\textbf{REQ 5a:}\forall P_i^G \in \mathcal{P}^G \ &:\ & P_i^G.pv = P_i^G.pos.M.I \cdot P_i^G.pos.M.V,\\
&\textbf{REQ 5b:}\forall P_i^C \in \mathcal{P}^C \ &:\ & P_i^C.pv = (-1)P_i^C.pos.M.I \cdot P_i^C.pos.M.V.
\end{aligned}$$

Requirement 6
REQ 6 focuses on transformers and ensures that the measured outgoing values are consistent with the transformation ratio. Since this must be ensured for both current and voltage there are again two requirements, REQ 6a (for voltage) and REQ 6b (for current), formed:

$$\begin{aligned}
&\textbf{REQ 6a:}\forall T_i \in \mathcal{T} \ &:\ & T_i.out.M.V = \frac{T_i.in.M.V}{T_i.r(T_i.p)},\\
&\textbf{REQ 6b:}\forall T_i \in \mathcal{T} \ &:\ & T_i.out.M.V = T_i.r(T_i.p) \cdot T_i.in.M.I.
\end{aligned}$$

3.2.2 Safety Requirements

While the physical requirements ensure that the gained data from the electrical grid is feasible from a physically point of view, the safety requirements check if the grid is in a safe state with respect to desired properties of the grid. Desired properties could be e.g. that certain security thresholds must be met. From a physical point of view it is partly possible to mitigate these thresholds but it could e.g. harm the components or lead to an increased wearing. Therefore, the owner of the electrical grid is probably interested to fulfil these requirements to minimize the risk of components needing to be replaced or customers being not supplied with energy.

Summarizing, these requirement should be viewed more as a proposal of possible safety requirements and could be extended or adapted depending on the special needs of a specific (sub-)grid.

Requirement 7

REQ 7 defines such an appealed security threshold. It ensures that the current on all power lines does not exceed a certain (grid-specific) threshold:

$$\textbf{REQ 7: } \forall\ L_i \in \mathcal{L}\ \forall\ M_j \in L_i.\text{M: } M_j.\text{I} \leq L_i.I_{\max}.$$

Requirement 8

The eighth requirement defines a safety threshold like in REQ 7 but now to target voltage. Since the European Committee for Electrical Standardisation released in its Harmonization Document that the boundary for voltage levels is +- 10% within the reference value of 230V/400V, this should be turned into one requirement. For voltage levels higher than 230V/400V there are slightly different safety thresholds, but in the scope of this thesis always the 10% will be adapted as a requirement:

$$\textbf{REQ 8: } \forall\ L_i \in \mathcal{L}\ \forall\ M_j \in L_i.\text{M: } M_j.\text{V} \in [0{,}9 \cdot L_i.V_{\text{ref}}; 1{,}1 \cdot L_i.V_{\text{ref}}].$$

Requirement 9

As defined in Section 3.1.2 fuses and protective relays cannot be switched on again if they were broken but must be replaced. Therefore REQ 9 checks if all fuses and protective relays are still functional. In addition to its security aspect, this requirement enables the grid operator to be informed directly if a fuse or a protective relay was broken between two measurements:

$$\textbf{REQ 9: } \forall\ F_i \in \mathcal{F}\text{: } F_i.\text{st} = 1 \land \forall\ R_j \in \mathcal{R}\text{: } R_j.\text{S.st} = 1.$$

Requirement 10

Closely related to REQ 9, REQ 10 checks if the cutting current $I_{\max}$ is exceeded at any of the fuses or protective relays:

$$\begin{aligned} \textbf{REQ 10:} \forall F_i \in \mathcal{F} \ &: \ (F_i.pos.M.I < F_i.I_{\max}) \land \\ \forall R_j \in \mathcal{R} \ &: \ (R_j.pos.M.I < R_j.I_{\max}). \end{aligned}$$

Requirement 11

To ensure that the voltage security threshold is met in transformers with respect to the current tap position, the REQ 11a and 11b are formulated. REQ 11a calculates the effect of the transformation rate on the reference input voltage to ensure that the output is within the boundary of +-10% of the reference value (230V/400V). Again, as in REQ 8, different boundaries for higher voltage grids will be neglected here. REQ 11b applies the same idea for the measured input voltage with respect to the output reference voltage:

$$\begin{aligned}\textbf{REQ 11a:}&\forall T_i \in \mathcal{T} : T_i.in.V_{\text{ref}} \cdot T_i.r(T_i.p) \in \quad [0,9 \cdot T_i.out.V_{\text{ref}}; 1,1 \cdot T_i.out.V_{\text{ref}}],\\ \textbf{REQ 11b:}&\forall T_i \in \mathcal{T} : T_i.in.M.V \neq 0 \rightarrow\\ &T_i.in.M.V \cdot T_i.r(T_i.p) \in \quad [0,9 \cdot T_i.out.V_{\text{ref}}; 1,1 \cdot T_i.out.V_{\text{ref}}].\end{aligned}$$

Requirement 12

To ensure that every consumer is always connected to the electrical grid and can receive power, it is checked with REQ 12 that the voltage at the each consumer is positive and the switch is in state *on*:

$$\textbf{REQ 12: } \forall\ P_i^C \in \mathcal{P}^C\text{: } (P_i^C\text{.pos.M.V} < 0 \land \forall\ S_j \in P_i^C\text{.pos: } S_j\text{.st=1}).$$

Requirement 13

Requirement 13 checks that no power is lost within the electrical grid i.e. the power generated is equal to the power consumed:

$$\textbf{REQ 13: } \sum_{P_i^G \in \mathcal{P}^G} P_i^G\text{.pv} = \text{-} \sum_{P_j^C \in \mathcal{P}^C} P_j^C\text{.pv}.$$

Requirement 14

Similarly to security thresholds defined by law or standardized in the previous requirements, there also should be requirements to check whether the measured data exceeds a user-defined threshold. This enables the operator of the electrical grid to set so called set points closely to the wanted current and voltage. By doing this monitoring can check whether all values are still in appropriate proximity. Since choosing such set points is highly dependent on the electrical grid itself here these set points are just indicated by a variable *sp*. REQ 14a checks if the measured current is within these set point boundaries and REQ 14b does this for the measured voltage:

$$\begin{aligned}&\textbf{REQ 14a:}\forall M_i \in \mathcal{M} : M_i.I_{\text{sp}} \in [(1-\text{sp}) \cdot L_j.I_{\max}; 1+\text{sp}) \cdot L_j.I_{\max}], \text{if} M_i \in L_j.M,\\ &\textbf{REQ 14b:}\forall M_i \in \mathcal{M} : M_i.V_{\text{sp}} \in [(1-\text{sp}) \cdot L_j.V_{\text{ref}}; 1+\text{sp}) \cdot L_j.V_{\text{ref}}], \text{if} M_i \in L_k.M.\end{aligned}$$

Requirement 15
To check if the interlocks are respected, REQ 15 was placed. It is splitted in REQ 15a and REQ 15b for both static interlocks (REQ 15a) and dynamic interlocks (REQ 15b). REQ 15a consequently checks if the specified number of switches is connected to interlock K_i. REQ 15b checks that the maximum capacity of the connected lines matches the required current of that interlock K_j:

$$\textbf{REQ 15a} : \forall K_i \in \mathcal{K}_{\text{static}} : \sum_{S_i \in K.S} (S_i.st) \geq K.CS_{\text{min}},$$

$$\textbf{REQ 15b} : \forall K_j \in \mathcal{K}_{\text{dynamic}} : \sum_{S_j \in K.S} (S_j.st \cdot S_j.L_k.I_{\text{max}}) \geq K.I_{\text{min}}, \text{if} S_j \in L_j.S.$$

3.3 Evaluation scopes

In a perfect scenario all measured data sets would be without inaccuracy, securely and *directly* be available at a central SCADA server, ready to be checked against the just defined requirements at once. Shortly, this would require *global* knowledge in order for the monitoring system to function. Unfortunately this is not likely the case. The proposed monitoring approach aims to detect manipulation both in local measurements and in the central SCADA server in a distributed and hierarchical way. Because communication endpoints and protocols are always an additional risk, the following approach aims to minimize the need for completely secure communication (e.g. by minimizing the communication itself when it is not needed or adding superfluous and therefore double-checking communication).

The core idea here is not to evaluate all requirements at once after all the data was communicated to a central entity. Instead the distributed subgrid structure is used and each requirements evaluated as soon and as local as it is possible. This approach reduces the communication needs since as only needed data will be communicated to entities further away from the local field station. Additionally in case of an attack against the monitoring system itself, it isn't a central target but neatly distributed all over the grid. These distributed monitors will only be checking some of the overall requirements and cannot decide if the system itself is in a stable state for themselves.

It can be assumed that the *static* data was safely exchanged before the start of the monitoring system and during operation only the *dynamic* data, the data generated from measurements, needs to be communicated. Furthermore it will be assumed that each subgrid has a entity which receives the *dynamic* data generated in the subgrid without delay. In case that the subgrid consists e.g. of a local substation, the central entity might be a part of the RTU. In this thesis a subgrid and its measured data will be seen as *local* environment and locally achieved data.

With the idea in mind to minimize communication in mind it is only logical to implement a *local* monitor that evaluates all requirements that can be evaluated with the *local* data available at each subgrid. If a fault arrives, an alert can be send directly. If no fault arrives, this part of the overall data can be considered as safe as that it does not violate the locally focused requirements. The *local* monitors are already a big step for achieving a distributed monitoring system, since now every subgrid can check their relevant requirements simultaneously.

One logical step above the requirements that can be checked *locally*, are those that need more information than locally available at one subgrid but less than the complete *global* information. This requirement scope will be called *neighbourhood* scope as it involves exchanging data with the neighbouring subgrids. The *neighbourhood* of a given subgrid Ω_i means every subgrid that Ω_i shares at least one power line L_i with. Since power lines were defined as a connection between two components from two different subgrids at maximum, a subgrid Ω_i may have more than one neighbouring subgrid in total but it always consists of tuples containing Ω_i and one neighbouring subgrid. Unless the complete electrical grid Ω consists of only one subgrid Ω_i, which would be trivial, each subgrid has at least one neighbouring subgrid and therefore always has a *neighbourhood*. Just as the evaluation of all *local* requirements is possible as soon as all the *local* data was available, the evaluation of all *neighbourhood* requirements is possible as soon as the needed *neighbourhood* data is available. Again, there is no need to have *global* data knowledge of the whole system. Especially since power lines were defined to be connecting only two subgrids and the *neighbourhood* requirements focus on these connecting power lines, the data exchange is only needed between every neighbour pair. Still: That is why it currently makes no difference if the neighbouring subgrids are structured dense or rather sparse. But since all neighbours share at least that one power line it is definitely needed to communicate between those two neighbours to checking the border region between the two neighbours. In order to not subordinate one neighbouring subgrid to another the data between both subgrids will be exchanged mutually. After this exchange the monitoring system on both sides will check the *neighbourhood* requirements with the achieved data. This could either be the *local* monitor itself or a distinguished *neighbourhood* monitor located at the same place as the local monitor. Since the two monitors now work on the same data set they should trigger the same

alerts if one or more requirements are violated.

Lastly, a logical step above the *neighbouring* data knowledge scope remains the *global* data knowledge scope. The *global* scope of course focuses on requirements that can only be evaluated with complete data knowledge of the electrical grid. Consequently one central monitor is needed to evaluate these *global* requirements and every local monitor needs to communicate its data to it. Of course here an appropriate communication mechanism needs to be found to send the data securely to the *global* monitor. But all the data only needed for *local* and *neighbourhood* requirement does not need to be communicated again.

Summarising it can be said that there are three different evaluation scopes for evaluating the physical and safety requirements. The three scopes can be distinguished by the fact which rate of information is needed to evaluate it. In general three of these information scopes can be distinguished like this:

1. *local*: requirements that can be evaluated with the information available at a single subgrid
2. *neighbourhood*: requirements that can evaluated after a data exchange between two subgrids containing data of their shared border region
3. *global*: requirements that need global knowledge to be evaluated

In addition to this summary figure 3.2 gives an exemplary overview over a possible electrical grid Ω, its four subgrids $\Omega_{1\text{-}4}$, a matching monitoring system and their data exchange ways. The continuous yellow lines indicate one or more power lines connecting the subgrids. While subgrid $\Omega_{1\text{-}3}$ are all connected to each other, subgrid Ω_2 shares an additional power line connection with subgrid Ω_4 which is not connected to subgrid Ω_1 and subgrid Ω_3. Hence, subgrid Ω_2 has three neighbours ($\Omega_{1,3,4}$), while subgrid Ω_4 only has one neighbour (Ω_2) and subgrid Ω_1 and Ω_3 both have each other and subgrid Ω_2 in their neighbourhood. The green dashed lines indicate that the data generated from each subgrid is send to a *local* monitor ($M_{1\text{-}4}$) which then would evaluate their *local* requirements. The data exchange for the *neighbourhood* scope is then done between each neighbouring subgrids (or between their *local* monitors of the neighbouring subgrids), indicated by a dashed light blue line. This data exchange on subgrid level just focuses on the *locally* generated data. For example subgrid Ω_3 sends the same *local* data to subgrid Ω_1 and subgrid Ω_2. It does not redirect data received from subgrid Ω_1 to subgrid Ω_2. Finally all the *local* monitors send their data to the global monitor, which is indicated by pointed dark blue lines. In the figure, the global Monitor is located next to a global SCADA server. Still: This must not always be the case physically, the location in the figure should emphasise the central nature of the global monitor as a part of a decentralized monitoring system.

From this point on it is reasonable to evaluate each requirement on their lowest

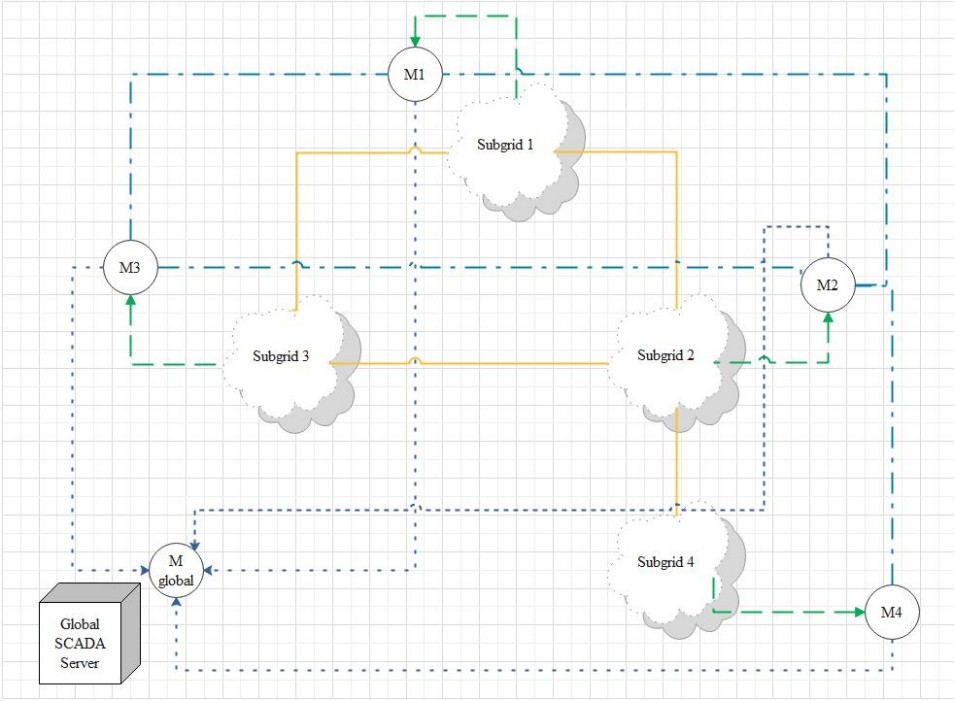

Figure 3.2: An example grid Ω, with four subgrids and its monitoring system.

possible scope to prevent unnecessary information exchange. Additionally it can be assumed that the *local* evaluation can be done faster than the *global* evaluation, because it functions without additional (external) communication. During this thesis a sequential, hierarchical way of evaluating the requirements will be pursued. This means that the data is only communicated to the neighbouring subgrids after the *local* evaluation is done. Consequently the data is only communicated to the *global* monitor after the *neighbourhood* evaluation showed is done. This hierarchical approach should prevent the system from perform large evaluations on data that has already been faulty at a basic level. Additionally this approach clearly can show on which level which type of attacks could be detected. In a real world scenario of course, it would probably make sense to parallelize the evaluation at least partly.

In the following the requirements defined in Section 3.2 will be assigned to their appropriate knowledge scope needed. The individual requirements will not be directly used as presented in Section 3.2 but adapted to the subgrid structure. If e.g. REQ 1 ensures that Kirchhoff's law is hold for every bus, it will be formulated that every bus in that subgrid holds Kirchhoff's law. Logically, if each subgrid ensures that for its (local) buses the requirement is met and does not raise an alert, all buses in the complete electrical grid are checked and meet the requirement. Since power lines are the only components which can be shared between two subgrids one of the main decision points will be if the requirements includes (possibly) shared power lines or not.

3.3.1 Local scope

The *local* evaluation scope will hold every requirement that can be evaluated with the information from one subgrid Ω_a alone. Depending on the general design of a subgrid, the actual requirements that can be checked may differ (as not all subgrids need to contain each type of component), but since a generic subgrid is assumed, all possible requirements will be explained. Further during the scope of this thesis the subgrids will be chosen as substation and their assigned components.
For a subgrid $\Omega_a = (\mathcal{P}_a, \mathcal{B}_a, \mathcal{T}_a, \mathcal{L}_a, \mathcal{S}_a, \mathcal{M}_a, \mathcal{F}_a, \mathcal{R}_a, \mathcal{K}_a)$ the *local* requirements are the following:

- **REQ 1 L:** $\forall\ B_i \in \mathcal{B}_a$: $\sum_{L_j \in B_i.\text{in}} L_j.B_i.\text{M.I} = \sum_{L_j \in B_i.\text{out}} L_j.B_i.\text{M.I}$.
  The first requirement, which ensures Kirchoffs's Law for each bus, can be evaluated locally since all meters which are assigned to a power line and a bus, automatically have to be assigned to the same subgrid as their bus, even if the power line connects two subgrids.

- **REQ 2 L:** $\forall\ B_i \in \mathcal{B}_a, \forall\ L_i, L_k \in B_i.\text{in} \cup B_i.\text{out}$: $L_j.B_i.\text{M.V} = L_k.B_i.\text{M.V}$.
  The same argument applies as for REQ 1. If the bus would have one power line which is shared between two subgrids, the meter in question will still be assigned to the same subgrid as the bus.

- **REQ 3 L:** $\forall\ L_i \in \mathcal{L}_a \wedge L_i.\text{or} = (\text{a,a})\ \exists\ S_j \in L_i.\text{S}$:
  $S_j = 0 \rightarrow \forall\ M_k \in L_i.\text{M} : M_k.\text{I} = 0$.
  This requirement is the first that will be splitted between the *local* and the *neighbourhood* scope. This part now only counts for inner power lines within the subgrid. The power lines assigned to the subgrid Ω_a which connect it to other subgrids will be targeted during the *neighbourhood* scope. But since only completely *local* power lines are evaluated here the evaluation itself is of course *local*, too.

- **REQ 4 L:** $\forall\ L_i \in \mathcal{L}_a \wedge L_i.\text{or} = (\text{a,a})\ \forall\ M_j,\ M_k \in L_i.\text{M}$:
  $M_j.\text{I} = M_k.\text{I} \wedge M_j.\text{V} = M_k.\text{V}$.
  Just like in REQ 3 this version of REQ 4 only evaluates the completely *local* power lines of subgrid Ω_a.

- **REQ 5a L:** $\forall\ P_i^G \in \mathcal{P}_a^G$: $P_i^G = P_i^G.\text{pos.M.I} \cdot P_i^G.\text{pos.M.V}$.
  Power generators can only be part of one subgrid, thereforethe data is *locally* available.

- **REQ 5b L:** $\forall\ P_i^C \in \mathcal{P}_a^C$: $P_i^C = (\text{-1})\ P_i^C.\text{pos.M.I} \cdot P_i^C.\text{pos.M.V}$.
  Power consumers can only be part of one subgrid (or are connected as different entities if they are attached to more than one subgrid), the data is *locally* available.

- **REQ 6a L:** $\forall\ T_i \in \mathcal{T}_a$: T_i.out.M.V $= \frac{T_i.in.M.V}{T_i.r(T_i.p)}$.
  This requirement only relates to transformers and their local information and can therefore be evaluated *locally*.

- **REQ 6b L:** $\forall\ T_i \in \mathcal{T}_a$: T_i.out.M.V $= T_i$.r(T_i.p) $\cdot\ T_i$.in.M.I.
  Same argument applies here as for 6a.

- **REQ 7 L:** $\forall\ L_i \in \mathcal{L}_a\ \forall\ M_j \in L_i$.M $\cap\ \mathcal{M}_a$: M_j.I $\leq L_i.I_{\max}$.
  Unlike the requirements 3 and 4, requirement 7 will be evaluated on all power lines assigned to the subgrid Ω_a, including the ones that are connecting Ω_a to other subgrids. But requirement 7 will not be evaluated for all meters on the connecting lines, but only for those assigned to the subgrid Ω_a. For this requirement it is sufficient to check only the meters on each side of the neighbouring region because the other subgrid will check that the measured current is less equal than the reference value. There is no need to exchange this data between the subgrids. For inner power lines the intersection does not make any difference since all the meters assigned to an inner power line belong to the same subgrid.

- **REQ 8 L:** $\forall\ L_i \in \mathcal{L}\ \forall\ M_j \in L_i$.M $\cap\ \mathcal{M}_a$:
  M_j.V $\in$ [0,9$\cdot L_i.V_{\text{ref}}$;1,1 $\cdot\ L_i.V_{\text{ref}}$].
  With the same argument as presented for requirement 7 it is sufficient here to check only the meters available at the subgrid for every available power line (including the connecting ones).

- **REQ 9 L:** $\forall\ F_i \in \mathcal{F}_a$: F_i.st $= 1 \wedge\ \forall\ R_j \in \mathcal{R}_a$: R_j.S.st $= 1$.
  All information of fuses and protective relays are available at a *local* level.

- **REQ 10 L:** $\forall\ F_i \in \mathcal{F}_a$:
  (F_i.pos.M.I $< F_i.I_{\max}$) $\wedge\ \forall\ R_j \in \mathcal{R}$: ($R_j$.pos.M.I $< R_j.I_{\max}$).
  The meters needed to evaluate requirement 10 always belong to the same subgrid as the fuse or protective relay they are assigned to so this requirement can be evaluated *locally*.

- **REQ 11a L:** $\forall\ T_i \in \mathcal{T}_a$:
  T_i.in.$V_{\text{ref}} \cdot\ T_i$.r($T_i$.p) $\in$ [0,9 $\cdot\ T_i$.out.V_{ref};1,1 $\cdot\ T_i$.out.V_{ref}].
  All transformator-related information is available *locally*.

- **REQ 11b L:** $\forall\ T_i \in \mathcal{T}_a$:
  T_i.in.M.V$\neq 0 \rightarrow T_i$.in.M.V $\cdot\ T_i$.r(T_i.p) $\in$ [0,9 $\cdot\ T_i$.out.V_{ref};1,1 $\cdot\ T_i$.out.V_{ref}].
  Same argument applies as for Requirement 11a.

- **REQ 12 L:** $\forall\ P_i^C \in \mathcal{P}_a^C$: ($P_i^C$.pos.M.V $< 0 \wedge\ \forall\ S_j \in P_i^C$.pos: S_j.st=1).
  Since this requirement only checks if all consumers of a certain subgrid are connected to the subgrid and receive positive voltage, this requirement too, can be checked *locally*.

- **REQ 14a L:** $\forall\ M_i \in \mathcal{M}_a$: $M_i.I_{\text{sp}} \in [(1\text{-sp}) \cdot L_j.I_{\max};1+\text{sp}) \cdot L_j.I_{\max}]$, if $M_i \in L_j.\text{M}$.
  Meters can only be assigned to one subgrid, therefore their measurements are available *locally*. Even if a checked meter is assigned to a power line connecting Ω_a with another subgrid, the reference value is a *static* property and is the same on both sides. No further data exchange is needed.

- **REQ 14b L:** $\forall\ M_i \in \mathcal{M}_a$: $M_i.V_{\text{sp}} \in [(1\text{-sp}) \cdot L_j.V_{\text{ref}};1+\text{sp}) \cdot L_j.V_{\text{ref}}]$, if $M_i \in L_k.\text{M}$.
  Same argument applies as for REQ 14a.

- **REQ 15a L:** $\forall\ K_i \in \mathcal{K}_{a_{\text{static}}}$: $\sum_{S_i \in K.S} (S_i.\text{st}) \geq \text{K}.CS_{\min}$.
  Interlocks and their assigned switches are always part of the same subgrid and therefore their data is *locally* available.

- **REQ 15b L:** $\forall\ K_j \in \mathcal{K}_{a_{\text{dynamic}}}$:
  $\sum_{S_j \in K.S} (S_j.\text{st} \cdot S_j.L_k.I_{\max}) \geq \text{K}.I_{\min}$, if $S_j \in L_k.S$.
  Same arguments applies as for REQ 15a.

3.3.2 Neighbourhood scope

In this section the requirements that need data exchange between two subgrids are explained. As already said in the explanation about how the different scopes are formed, there are currently no requirements that would require data from multiple neighbouring subgrids together e.g. taking advantage of a ring-structure. Currently each of the data exchanges for the *neighbourhood* scope is only done between two neighbouring subgrids which share at least one power line. Of course one subgrid may have multiple of these connections with different subgrids.

As already done for the *local* scope, the requirements are adapted to fit the *neighbourhood* scope. The following requirements are formulated for a subgrid $\Omega_a = (\mathcal{P}_a, \mathcal{B}_a, \mathcal{T}_a, \mathcal{L}_a, \mathcal{S}_a, \mathcal{M}_a, \mathcal{F}_a, \mathcal{R}_a, \mathcal{K}_a\ )$ which shares at least one power line with its neighbour $\Omega_b = (\mathcal{P}_b, \mathcal{B}_b, \mathcal{T}_b, \mathcal{L}_b, \mathcal{S}_b, \mathcal{M}_b, \mathcal{F}_b, \mathcal{R}_b, \mathcal{K}_b\ )$.

- **REQ 3 N:** $\forall\ L_i \in \mathcal{L}_a \wedge L_i.\text{or} = (\text{a,b})\ \exists\ S_j \in L_i.\text{S}$:
  $S_j = 0 \rightarrow \forall\ M_k \in L_i.\text{M} : M_k.\text{I} = 0$
  Unlike presented in the local scope, REQ 3 N now is evaluated only for non-inner, i.e. connecting, power lines between Ω_a and Ω_b. If a switch is present on at least one side of that connecting power line, it must be checked that for all connected meters to that power line (the meters assigned to subgrid Ω_a and Ω_b) that there is no current if the switch is *open*.

- **REQ 4 N:** $\forall\ L_i \in \mathcal{L}_a \wedge L_i.\text{or} = (\text{a,b})\ \forall\ M_j,\ M_k \in L_i.\text{M}$:
  $M_j.\text{I} = M_k.\text{I} \wedge M_j.\text{V} = M_k.\text{V}$
  Just like for REQ 3 N, now REQ 4 N is formulated for the power lines connecting Ω_a and Ω_b. To evaluate this command on connecting power lines the measured data from the meters on both sides of the subgrid must be present.

3.3.3 Global scope

Finally, in this subsection the focus is on requirements that need *global* knowledge from all subgrids. More precisely, the *global* scope includes all requirements that need data from two subgrids or more which are not neighbours so they cannot exchange their data within the neighbourhood scope. This small addition is needed since there might be requirements that do not require data from all subgrids e.g. if one of the subgrids lacks a relevant component type.

- **REQ 13 G:** $\sum_{P_i^G \in \mathcal{P}^G} P_i^G.\text{pv} = - \sum_{P_j^C \in \mathcal{P}^C} P_j^C.\text{pv}$
  As this requirement focuses on every power consumer and producer in the complete electrical grid it can only be evaluated globally. To make this requirement a neighbourhood requirement a special grid *topology* would be needed where all consumers and producers are within one neighbourhood.

3.3.4 Practical considerations

It is clear that in general more data and more precise data leads to a better evaluation of the requirements independent of their scope. If data is missing, some requirements cannot or can only partly be evaluated in a correct way. If one measurement is delivered with high inaccuracy this may lead to false conclusions. However, if during the evaluation process higher inaccuracy is expected, the monitoring system might not detect the actual faults and produce false negatives. Consequently, in a perfect scenario, the sensors of the electrical grid would deliver complete and perfect measurements. This would ensure, that, unless in case of malfunction of a component or a manipulation, the physical requirements would always be fulfilled. The needed data accuracy must be adjusted matching a specific electrical grid and must be depending on the capabilities of the sensor.

Regarding to the scope of this thesis, the focus is on evaluating two subgrids which may not cover the complete power distribution system. Both subgrids will include one RTU and their controlled components. The assumed electrical grid however may include further components which are not part of the two subgrids. Finally, it should be stressed again that one subgrid does not automatically need to correspond to one field station.

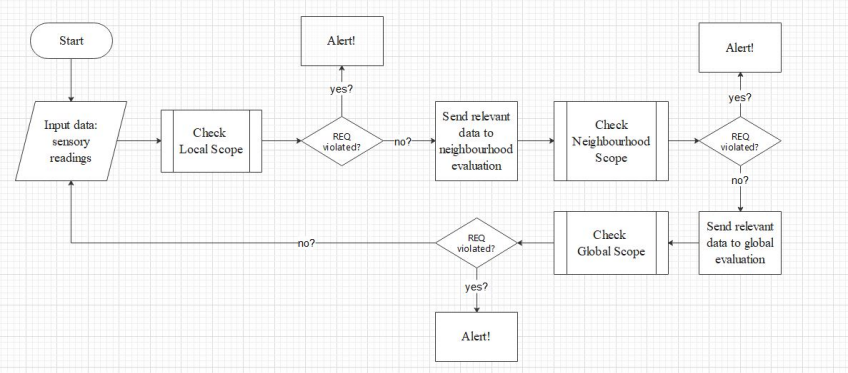

Figure 3.3: The outline of the monitoring algorithm as a flowchart.

3.4 Monitoring algorithm outline

As already mentioned, in this thesis should be regarded as a proof of concept work hat an exchange of relevant neighbourhood information can lead to an increased local and decentralized knowledge and prevent further attacks against the complete electrical grid.

With this in mind, the algorithm to evaluate the physical- and safety-requirements as a part of the monitoring system should be kept simple and easy to understand. At this point there are no particular demands regarding to the time-space complexity of the monitoring algorithm itself. A flow chart describing the basic algorithm is shown in figure 3.3 The algorithm should take sensory readings from each subgrid as a separate input. The main goal and output should be the decision whether the received input data resembles a safe or an unsafe situation regarding the defined physical- and safety-requirements. To make this decision the input data should be analyzed at first *locally* with the requirements from the local evaluation scope. Locally means in this case that the evaluation should be done decentralized by entities located at each subgrid and not at a central entity. If at least one of the requirements is violated by the set of input data, an alert is triggered. If all requirements are evaluated to be fine, a small part of the input data, which is relevant for each neighbouring subgrid, should be exchanged as part of the binary relation between each pair of neighbouring subgrids. With the additional input data gained from all neighbouring subgrids, the physical- and safety-requirements from the neighbourhood scope can be evaluated. Again, if at least one of the requirements is violated, an alert is triggered. If no violation is found, the input data can be considered safe in that context. Additionally, but beyond the scope of this thesis, a further data exchange on global level then could be triggered to evaluate the requirements from the global evaluation scope.

The concrete implementation of this algorithm sketch is further explained in Section 5.2.

4 Testbed

In this chapter a testbed is introduced that can be used to test the prototype implementation of the defined model. The testbed aids to inspect the feasibility, capability and limits of the prototype. First, the idea of testing with fixed scenarios is explained in Section 4.1 and the test objective is presented (Subsection 4.1.1) as well as the content-related requirements to the scenario (Subsection 4.1.2) and the individual scenarios (Subsection 4.1.3). Afterwards, the architecture of the testbed is illustrated in Section 4.2. While Subsection 4.2.1 the chosen framework for the subgrid simulation is explained, and in Subsection 4.2.2 it is explained how this framework is used for the testbed.

4.1 Testing with fixed scenarios

The model presented in the previous chapter targets the SCADA network within the electrical grid. Therefore, testing on a real environment is rather difficult, as a real-time connection for testing purposes could provide difficulties to the grid itself, which is more than undesirable for any provider. It is even difficult to receive real traffic data from an operator of an electrical grid, as normal traffic and appropriate manipulated or attacked traffic would be needed. Furthermore for a feasible approach, every rule should be tested by the testbed and edge cases and limits of the system must be explored. To overcome the lack of a direct connection to a physical system or (historical) test data, a simulation environment is needed.

Such a simulation must be able to portray an appropriate electrical grid with its typical components and it must be possible to start attacks against it. In the following subsection the test objectives, which a matching simulation has to fulfill are explained in more detail. Additionally, the requirements for a feasible virtual grid representation and possible attack scenarios are explained. These attack scenarios indicate which types of attacks the model can detect, as they simulate different manipulations to the traffic.

4.1.1 Test objectives

The overall goal of the testbed is to examine the implementation of the model presented in Section 3.1. The evaluation of the test are used to decide if the model fulfills the criteria specified in the problem statement and to prove the functionality of the implementation. Furthermore, the testbed gives insights

into limitations of the model and points out how it can be adapted and extended. Eventually this can help to move the model and its implementation from a prototype version to a real world application.

Summarising, the testbed is a way evaluate the presented implementation and to make it transparent and traceable from a scientific point of view.

4.1.2 Scenario Requirements

The core of each scenario used in the testbed needs to represent and simulate an electrical grid. This electrical grid should contain precisely the components described in the model formalism. Additionally, to validate the hierarchical approach, it should semantically make sense to split the electrical grid into subgrids. Within the scope of this thesis and the prototype implementation, two smaller parts of the electrical grid are nominated as two subgrids to provide the possibility to test the neighbourhood requirements. The two subgrids do not necessarily need to build the complete electrical grid, as global properties are not directly evaluated. However, it is desirable to be able to extend the scenario at a later point in time to check further model implementations (e.g. an implementation which also checks global requirements, or consists of more than two distinguished subgrids). The scenarios need to be able to support all necessary aspects of the grid topology, like e.g. parallel power lines. Furthermore, is necessary to support different types of topologies of subgrids, e.g. a subgrid which is only connected to one other subgrid and subgrids connected to multiple neighbours.

4.1.3 Scenario Descriptions

In the following, four scenario types are described which are used to evaluate the feasibility and the limitations of the proposed model and the physical- and safety-requirements. Therefore, each scenario type has a different thematic focus, which is stressed in their description. The scenario types may include more than one test case to test, e.g. the violation of different requirements that lead to the same sort of scope violation or result from the same type of attack, but are cumulated to one scenario type.

The four scenario types are described in the following way: first a textual description of the actual grid behaviour and a small example of a typical requirement that is hurt in this scenario is given, followed by the expected reaction of the monitoring system. Last, the thematic priority is stressed, to explain the scenario's background.

Scenario type 1: Normal grid operation

- **Description:** Normal grid operation without any external manipulation.
- **Example:** -
- **Expectation:** The monitoring system should not detect any problems, all checked physical and safety requirements should be "OK".
- **Objective:** This basic scenario type is used to demonstrate the proper working of the monitoring system. No unnecessary alerts should be triggered on normal operation. Furthermore this scenario type can be used to check the selected precision with which the requirements are evaluated.

Scenario type 2: Faults in the *local* scope

- **Description:** The data sets are manipulated to create small faults within a subgrid. The chosen faults all target *inner* components that belong to the same subgrid.
- **Example:** REQ 2 L: A single sensor reading is manipulated, consequently not all meters connected to one bus report the same voltage value.
- **Expectation:** The monitoring system should detect those faults with the correct requirement they violate while evaluating the *local* scope.
- **Objective:** This scenario type should prove that the monitoring system is able to detect faults within the *local* evaluation, prior to the information exchange with other parts of the subgrid.

Scenario type 3: Faults in the *local* and *neighbourhood* scope

- **Description:** The data sets from this scenario type contain faults within one subgrid as well as within the border region between two subgrids.
- **Example:** REQ 4 L/N: A sensor reading is manipulated, consequently not all meters connected to the same power line report the same voltage value.
- **Expectation:** The monitoring system should detect the faults on *inner* and *shared* components. Depending on the violated requirements, they should be detected in both, the *local* scope (at least on one side) and in the *neighbourhood* scope of both subgrids.
- **Objective:** The border region connecting two subgrids is a critical region, therefore faults within this region should always be detected by at least the *neighbourhood* monitors. Additionally, there may be readings that hurt requirements within the *local* scope and within the *neighbourhood* scope at the same time.

Scenario type 4: Faults in the *neighbourhood* scope only

- **Description:** The data sets from this scenario type are faultless when evaluated only locally, but contain faults in the border region.
- **Example:** REQ 3N: A switch is *open* at one side of subgrid 1 and the meters at subgrid 1 reports no current. However, a meter present at the same power line on side of subgrid 2, reports current.
- **Expectation:** The monitoring system should be able to detect these faults in the *neighbourhood* scope although the *local* evaluation found no implausible data sets.
- **Objective:** The border region connecting two subgrids is a critical region, therefore faults within this region should always be detected by at least the *neighbourhood* monitors. Such faults may be undetectable from the *local* scope.

4.2 Architecture of the Testbed

In the following sections the implementation details of the testbed are discussed. This involves the choice for an appropriate simulator framework in Subsection 4.2.1 on the one hand and the actual architecture of the simulation including all content-related simulation components in Subsection 4.2.2 on the other hand.

4.2.1 Simulation framework

To fulfill the requirements stated above a matching simulation framework is required, which can simulate the data available at the field stations (e.g. at a RTU). Since the approach aims a local monitoring approach which also takes information from two neighbouring substations into account, a complete network simulation is not needed. Therefore co-simulation frameworks that only simulate the complete network traffic, like OMNeT++ ([48]) RINSE ([49]) or OPNET ([50]), are not suitable. Furthermore, more advanced testbeds that require a connection to emulate real hardware or use proprietary software would not be practical in this context either.[6]

Independent of the choice of the co-simulation framework, a power simulator is always required, like PowerWorld ([49]), OpenDSS ([48]), MATPOWER-based Matlab/Simulink ([50]) or Mosaik ([51, 52]). Because Mosaik integrates

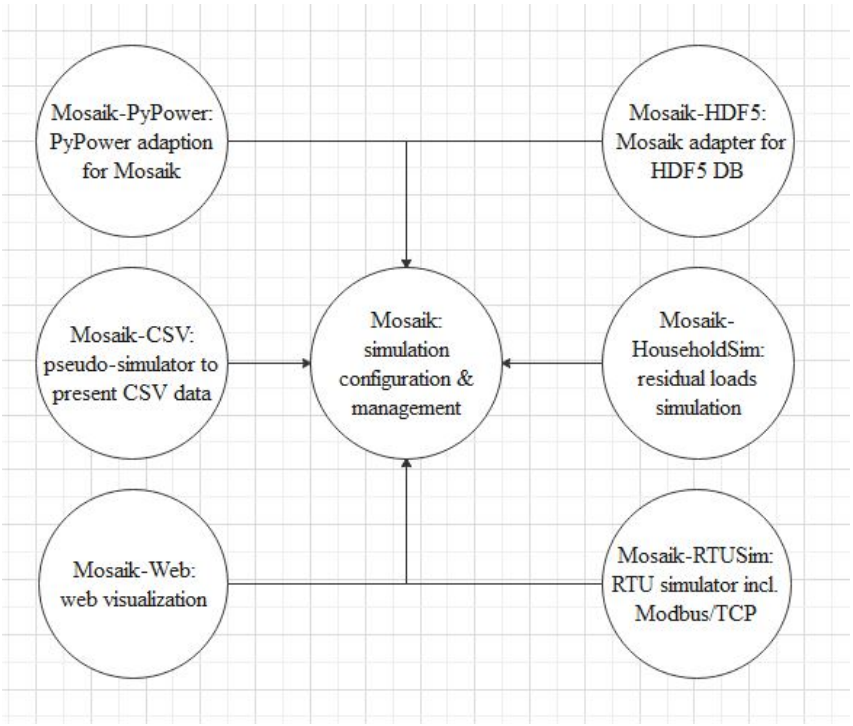

Figure 4.1: The co-simulation framework Mosaik and the six simulators used in the proposed testbed.

additional simulators easily into a smart grid co-simulation framework and can be expanded with self-developed simulators, it has been chosen by Chromik et al.([6, 53]) for their proposed testbed. An additional large advantage of the Open Source framework Mosaik is, that the used simulators do not necessarily need to be written in the same language as the Mosaik Simulator API is language agnostic[54]. Therefore further development and additions in the future are quite easy. Since this thesis is strongly based on the work of Chromik et al. Mosaik is used as a co-simulation framework as well in this thesis.

4.2.2 Architecture of the simulation

As visualized in figure 4.1, the proposed testbed for this thesis combines six different simulators using the co-simulation framework Mosaik: a **database**, a **CSV data pseudo simulator**, a **Mosaik-PyPower implementation**, a **RTU simulator**, a **household simulation**, and a **web visualization**.

An overview over the used technology stack of this components can be found in Table 4.1. Note, that some of the component are using further dependencies, like e.g. PyPower, which are not mentioned here. Using this six simulators and Mosaik the proposed testbed is developed to form a single simulation of an electrical grid, which serves the demands of the testbed.

Name	Developer	Version	License
Mosaik [55]	Steffen Schütte, Stefan Scherfke, Okko Nannen, Florian Schloegl André El-Ama	2.6.0	GNU LGPL v2.1
Mosaik-CSV [56]	Stefan Scherfke	1.0.4	GNU LGPL v2.1
Mosaik-HDF5 [57]	Stefan Scherfke	0.3	GNU LGPL v2.1
Mosaik-HouseholdSim [58]	Stefan Scherfke, Ontje Lünsdorf	2.0.3	GNU LGPL v2.1
Mosaik-PyPower [59]	Stefan Scherfke, André El-Ama	0.7.2	GNU LGPL v2.1
Mosaik-RTU [53]	Justina Chromik	0.1	GNU LGPL v2.1
Mosaik-Web[60]	Stefan Scherfke, Gunnar Jeddeloh	0.2.2	GNU LGPL v2.1

Table 4.1: Technology stack of Mosaik components used in the proposed testbed.

The **database simulator** stores the data generated from the simulation in a HDF5 database, including a relation-graph, time-series of the connected entities and additional metadata[57]. The **CSV data pseudo simulator** is used to present CSV data sets to Mosaik as a simulator[56]. The pseudo data simulator is used to simulate power generation by photovoltaic panels where the load is loaded from a CSV file. As mentioned in the last subsection a power flow solver is needed. In this case this is done by **Mosaik-PyPower**, an implementation of the power flow solver PyPower[61] for Mosaik[59]. The **RTU simulator** is started with an .xml file, which configures all components of the electrical grid that are operated by an instance of the RTU simulator. The respective RTU simulator then serves as a Modbus TCP Server during the simulation which exposes the data measured at its components in matching registers[53]. To resemble a small virtual neighbourhood, the **household simulation** creates residual loads based on load profiles.[58] Both, the household simulation and the photovoltaic panel simulation work with historical data stored in CSV files which consists of data in 15 minute intervals. Finally, the **web visualization** exposes the simulation to the localhost and gives an easy visual access to the simulated electrical grid as well as to the graphs which monitor, e.g. the residual loads or the power generated by the photo-voltaic panels[60]. Figure 4.2 shows an exemplary view of the visualization. The graph in the middle of the figure represents the electrical grid, the grey nodes buses within the grid, the green ones photo voltaic panels and the blue ones houses. The red node, currently selected, is House no. 15, for which its power consumption is shown in the graph below.

The database simulator, the CSV data pseudo simulator, the PyPower implementation, the household simulation and the web visualization all have been developed by the developers and maintainers of Mosaik[54, 62]. The RTU simulator was developed by Chromik [53]. During the development of the proposed testbed of this thesis, only small adaptions were done to the web visualization and the RTU simulator. In general the proposed testbed follows the structure of the Mosaik example scenario. Additionally, the demo grid file was adapted to create two distinguished subgrids within the complete electrical grid that can be operated by the two RTUs. Note, that the name of the simulator *RTU*

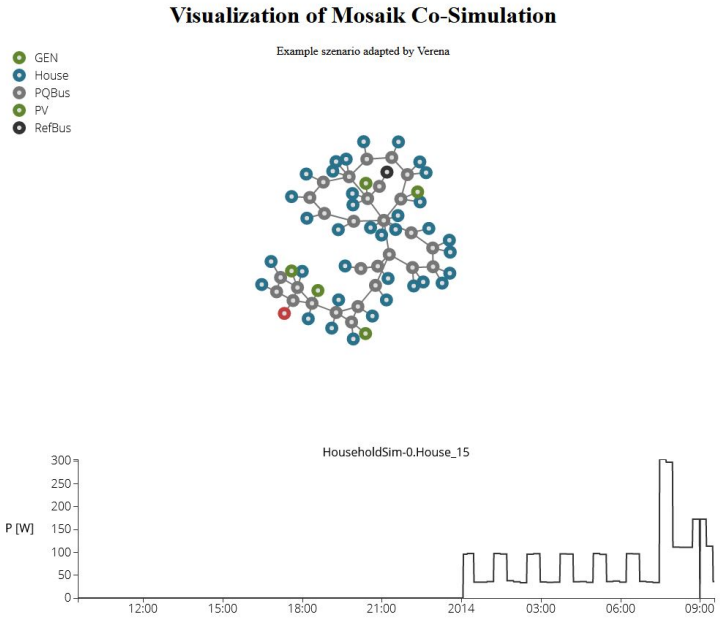

Figure 4.2: Exemplary state of the Mosaik web visualization.

has been kept from the work done by Chromik. During this thesis the RTU simulator is seen as the interface which gives centralized access to a configured subgrid (i.e. the components operated by that RTU).

The topology of the two subgrids/RTUs is configured by two xml-files. These two subgrids can be seen within the web visualization as marked in Figure 4.3 The realized topology itself is shown more schematically in Figure 4.4.

The first subgrid, called *Subgrid 0* (Ω_0), includes 7 power lines (4 inner and 3 shared ones), 4 buses, 1 switch and 12 meters and is only connected to on other subgrid, *Subgrid 1* (Ω_1), the second defined subgrid. *Subgrid 1* (Ω_1) contains 8 power lines (4 inner and 4 shared power lines), 4 buses and of course connected to *Subgrid 0* (Ω_0) but as well via power line "branch 35" to the rest of the simulated electrical grid. Additionally, *Subgrid 1* (Ω_1) consists of 12 meters and 2 switches.

The current attack possibilities of the simulation contain either a direct attack overwriting Modbus/TCP coils and registers or exchanging small snippets that simulate a PLC with malfunctioning version. Furthermore the testbed consists of a **helper program** which connects to both Modbus/TCP Servers of the two servers, reads their coils and registers and saves them to two output files in CSV format. These output files later will be used as (historical) input for regression testing of the proposed monitoring system.

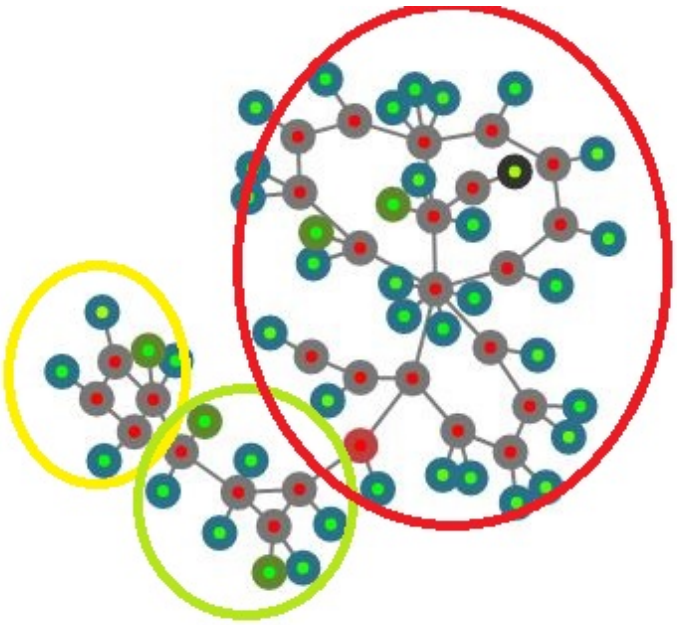

Figure 4.3: Subgrid 0 (yellow), Sugrid 1 (green) and the rest of the simulated electrical grid (red) within the web visualization.

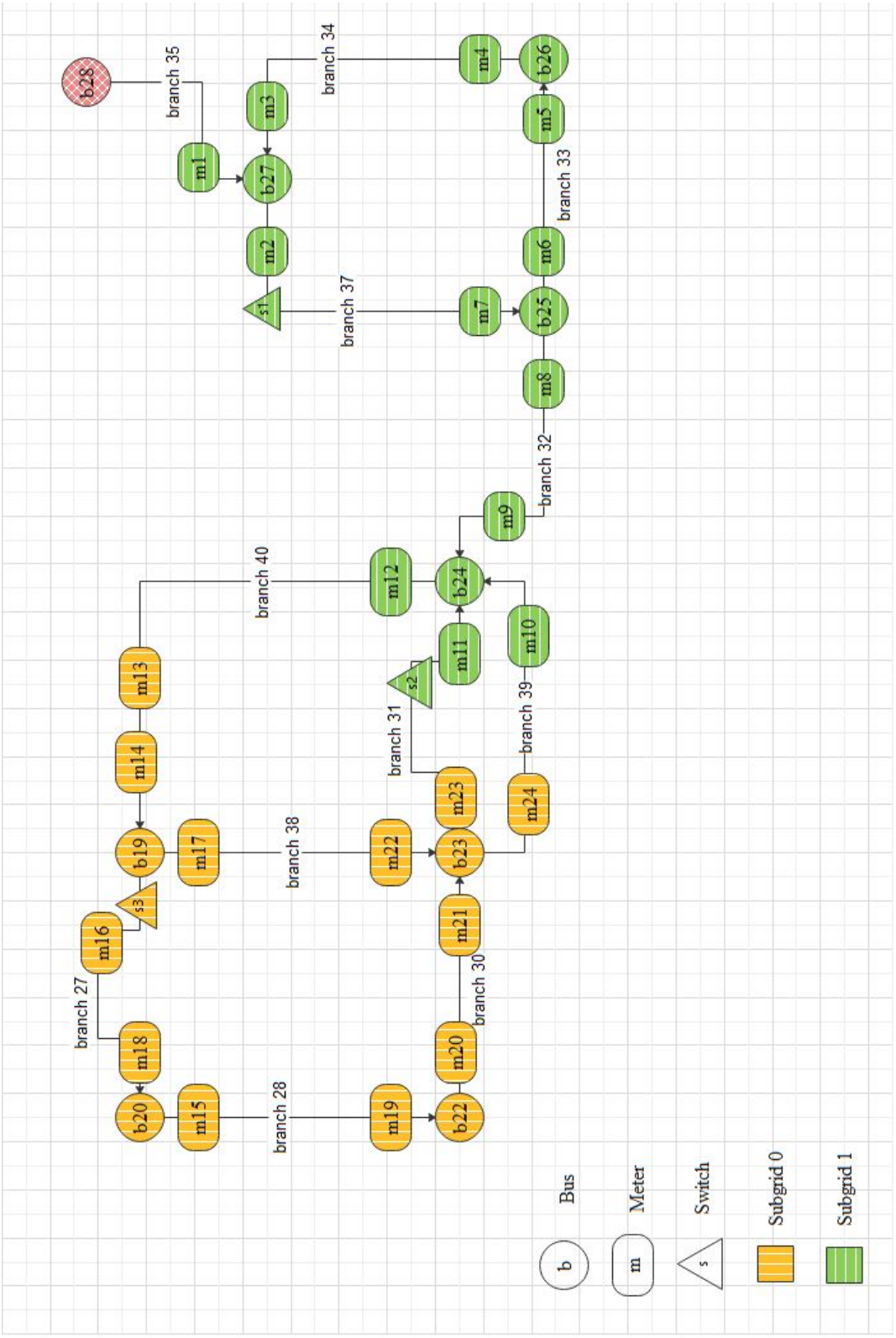

Figure 4.4: The grid topology for both subgrids used in the proposed testbed. b28 indicates the connection to the rest of the simulated grid.

5 Implementation

This chapter focuses on the implementation of the model presented in Chapter 3. To ensure a good implementation of the model, Section 5.1 discusses the functional and non-functional requirements for a solution. Thereafter in Section 5.2 the architecture of the implementation is explained to assist the understanding of how the different parts of the Intrusion Detection System work together. This includes a detailed view into the organization of the monitoring system itself (Subsection 5.2.1) and the virtual representation of the electrical grid that evaluates the physical- and safety-requirements (Subsection 5.2.2). Additionally it is explained how the implementation can be connected to the testbed (Subsection 5.2.3).

5.1 Solution requirements

To determine whether the created implementation fulfills its purpose correctly, it is crucial to define which requirements should be met. In general these requirements for software fall in two main categories: the functional and the non-functional requirements. While the functional requirements focus on the way the solution behaves the non-functional requirements focus on the quality attributes of the solution. In the following the functional and non-functional requirements for the implementation derived in this thesis are described.

5.1.1 Functional requirements

Model coverage

The implementation has to cover all components and physical- and safety-requirements presented in Chapter 3, that are testable by the proposed testbed. Since the implementation remains as a proof of concept work, components and physical- and safety-requirements do not need to be implemented, if they are not covered by the testbed. However, it should be easily possible to extend the implementation if new components or new physical- and safety-requirements would be added in later versions.

Input data

The implementation has to be able to work both with live data generated from the testbed and historical data read from a convenient input file (i.e. from a correctly formatted CSV-file). Additionally the implementation should come

with relevant test data in a persistent file that can be selected as input upon start.

Fault tolerance

Ideally the miss-rate of manipulations within the data should be zero. Of course it is not wanted that the monitoring system produces a lot of false-negatives to minimize the effort checking false alarms, but it is more important to not miss any right alarm, i.e. have a miss rate of zero.

5.1.2 Non-functional requirements

Usability

A decent graphical user interface is not used for this implementation as it will not be used by users without knowledge of the command line. However, the implementation should provide helpful output to the command line to enable the user to follow the taken steps of the monitoring. Additionally the command line interface should be helpful for debugging issues.

Documentation

There have to be README files for every directory to help the user to navigate through these directories. Furthermore the README files should contain the commands needed to start and operate the implementation. Every source code file should contain meaningful comments which help the understanding of the code. Every class and its functions should be commented.

System requirements

The implementation has to work on a normal office computer. Since the implementation will not be deployed on real hardware it is not mandatory to limit e.g. hard drive space. Additionally there are no hard limits regarding the computing time. However, if this proof of concept work would be migrated to a real world scenario these limits certainly will arise.

Code language and external dependencies

The implementation has to use the programming language Python 3.7[63]. It would be desirable to work with similar Python versions, too. This choice enables the simulation to be used with further software components for the Mosaik Simulation framework[62], most written simulators for it and depending software.

License

The implementation should be available as open source software. No commercial nor proprietary software should be used within the implementation.

Operational safety, installation, availability, GDPR
As this implementation is a proof of concept generated in the context of a master thesis and will not be installed or used outside of the academic context, it is not needed to take extra expenses to these topics.

5.2 Architecture

The following paragraphs give an insight on how the model is realized from a software development perspective. Subsection 5.2.1 explains the general approach of the monitoring system, the data-flow within it and the different communication ways. Afterwards, Subsection 5.2.2 shows how the physical- and safety-related requirements are evaluated on a virtual grid representation and feedback is given back to the monitoring system.

These two sections, the *general monitoring* and the *virtual grid*, represent the two thematic parts of the monitoring system: The distributed data management and alert system on the one hand and the actual evaluation of requirements on the other hand. To simplify the approach to the proposed implementation, two UML class diagrams are created. The first one, figure 5.1 focuses on the *general monitoring*, the second one, figure 5.2 on the *virtual grid components*. Both diagrams share the classes `subgrid` and `border_region` as combining points.

Note, that the proposed implementation is written in Python, a dynamically typed language, not all types are realized in the implementation as mentioned in the class diagrams. Furthermore, the `monitor_managment.py` is not a class, but the main program of the proposed implementation. However, both UML diagrams give good insight on the architecture of the implementation.

Lastly, the two different input ways of how the implementation can be connected to the testbed are described in Subsection 5.2.3.

5.2.1 General monitoring

In the following an overview over the general communication and data-flow of the implemented monitoring systems is given. In addition to the UML diagram in figure 5.1, the flow chart in figure 5.3, gives an overview about how the monitoring approach is implemented.

The core element of the proposed implementation is a file called `monitor_managment.py`. This file manages the complete setup of the monitoring systems, configures the different distributed, local monitors and starts and stops the actual monitoring. To preserve the distributed approach, the `monitor_managment.py` does not do the actual monitoring itself, but only, as

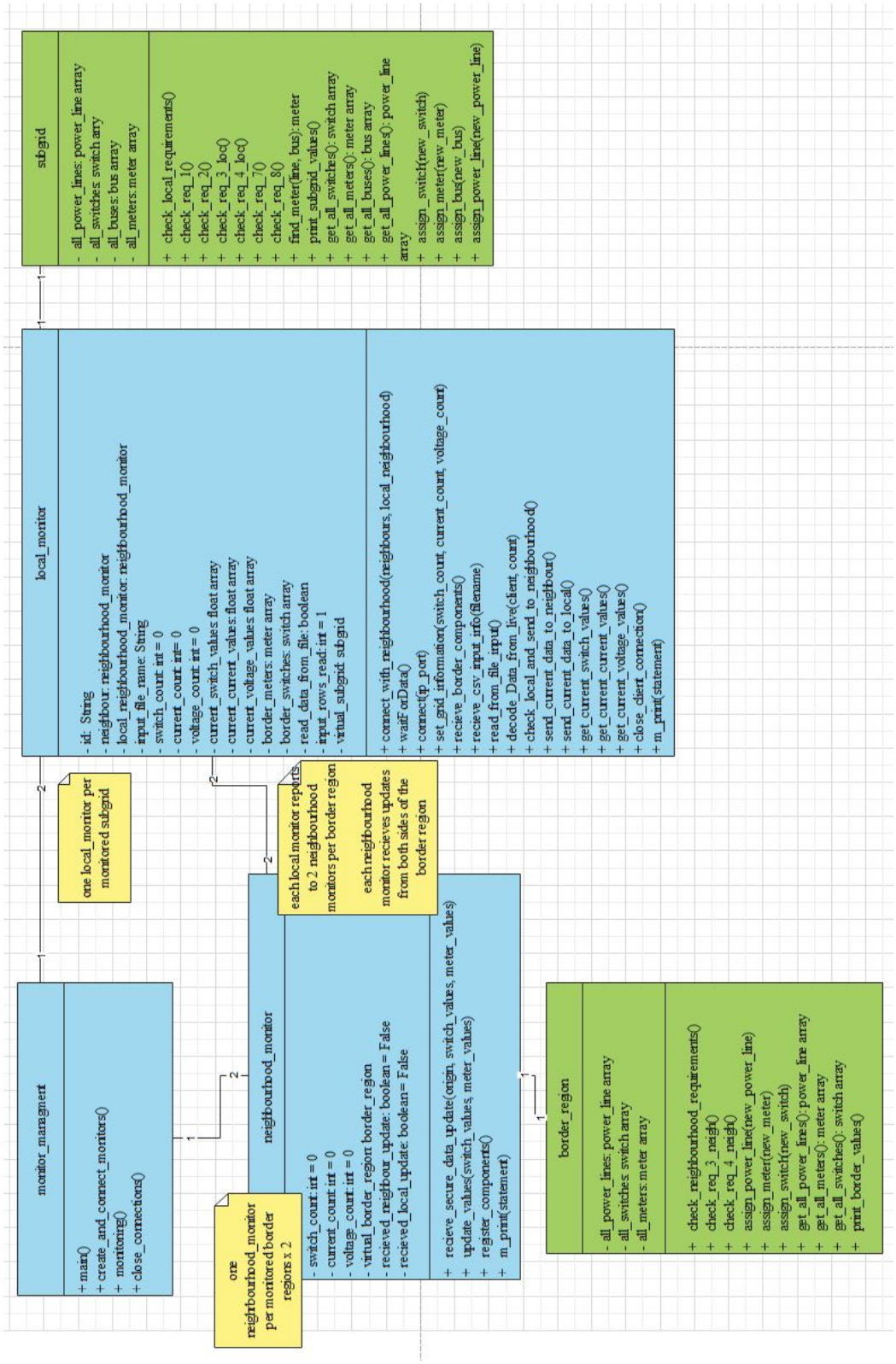

Figure 5.1: UML class diagram representing the collaboration between the general monitoring files (blue) and the two implemented virtual grid regions (green).

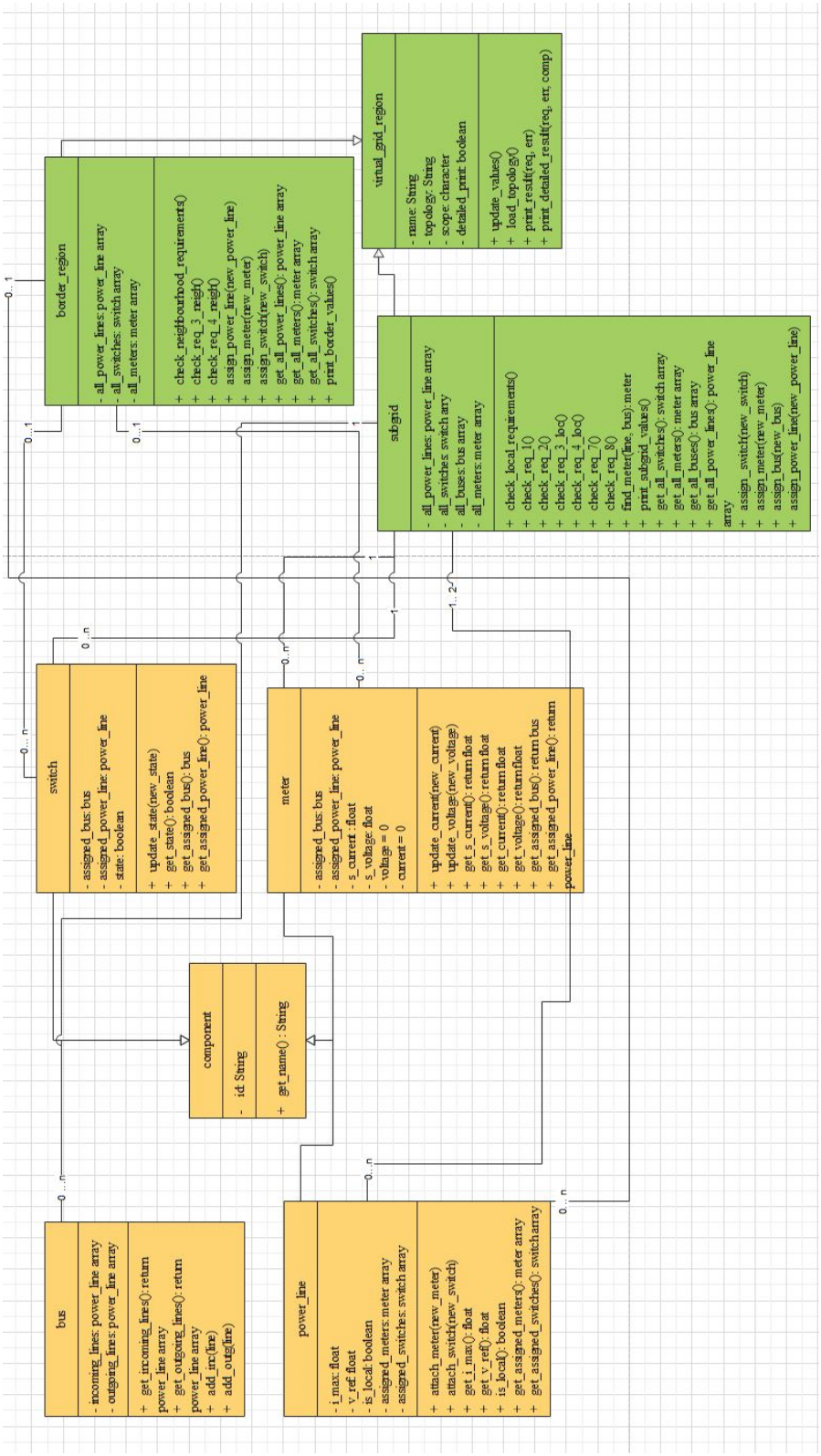

Figure 5.2: UML class diagram representing virtual grid implementation: the virtual grid regions (green) and the virtual grid components (yellow).

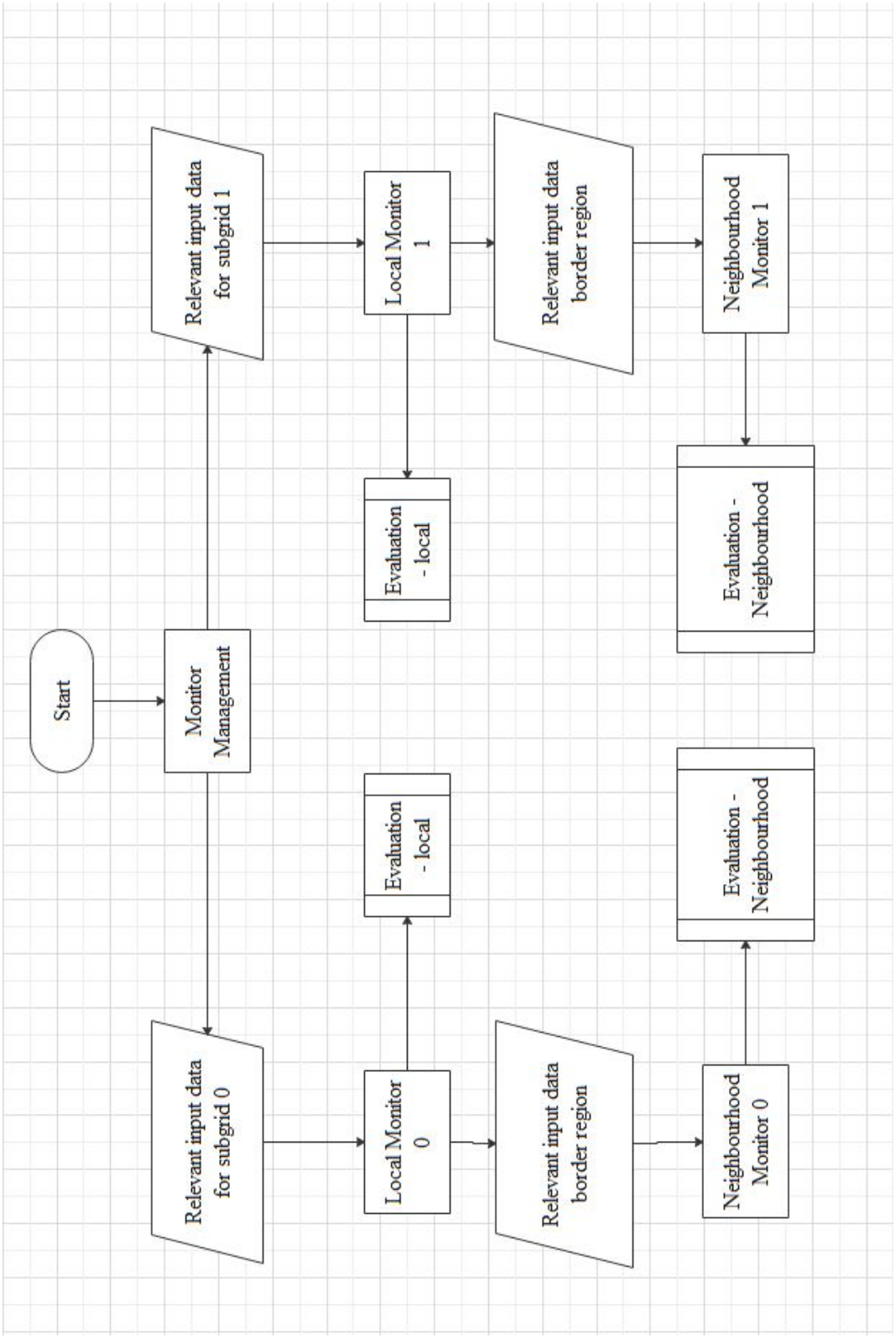

Figure 5.3: General communication and data-flow within the proposed implementation.

the name might suggest, manages it. This includes loading the correct topologies, starting the different local monitors and introducing them to each other and their respective input streams. In case of a real world application such a file would probably be located near the central SCADA server operating the SCADA network.

In the proposed implementation the monitoring system only monitors two neighboured subgrids (subgrid 0 and subgrid 1) and not a complete electrical grid. These two subgrids are sufficient for the scope of this thesis and serve as a proof of concept work for the monitoring system. However, an expansion to more subgrids up to a complete electrical grid, should be possible in future research with this approach.

In the proposed implementation, both subgrids are supervised by one local monitor (`local_monitor.py`) each. Both local monitors receive a configuration file in JSON format which encodes the topology of the subgrid they are supervising. Additionally, for both supervised subgrids, a matching neighbourhood monitor (`neighbourhood_monitor.py`) is created. These neighbourhood monitors are meant to supervise the border region between subgrid 0 and subgrid 1. In case of a third subgrid, each neighboured subgrid would have an additional neighbourhood monitor at their border with the third subgrid. The decision to have one locally available neighbourhood monitor at each supervised subgrid follows the proposal of the monitoring scopes from figure 3.2. This way, a border region between two subgrids is supervised by both sides from neighbourhood monitors. In case a subgrid gets attacked and the monitoring system on that side gets fed with faulty data sets, the neighbourhood monitor is not a single point of failure and the other neighbourhood monitor should be able to detect the attack.

Furthermore, the monitor management (`monitor_managment.py`) connects the local monitors to their input sources (see 5.2.3) and the neighbourhood monitors each local monitor has to keep updated. Then the monitor management starts the monitoring for a distinguished count of data sets. In the scope of this thesis, a data sets means one senor reading of every sensor (i.e. meters and switches) available within the proposed testbed from the same simulation step. After this count of data sets has been evaluated, the monitor management shuts everything down. Of course, in a real-world application, a shutdown of the monitoring system after a specific number of evaluation is not used.

Upon creation, each local monitor (`local_monitor.py`) and each neighbourhood monitor (`neighbourhood_monitor.py`) create a virtual representation of the part of the electrical grid they are supervising. This results in parts of a virtual grid, which functionality is further described in Section 5.2.2. As graphically described in figure 5.3, when the actual monitoring is started, each local monitor (`local_monitor.py`) reads on a data set of sensory updates from its input source. After the new row of input data is parsed, it is passed to the virtual grid as an update. Following the data update, a re-evaluation of all

local requirements is issued. When the local requirement check is finished successfully, the new input data is sent to the neighbourhood monitor which is present at the same subgrid as the local monitor and to each neighboured neighbourhood monitor.

When a neighbourhood monitor received a completely new data set (one part from its own subgrid's local monitor and one part form the local monitor of the neighboured subgrid), an update of the virtual border region is issued. After the update, a re-evaluation of the neighbourhood requirements is triggered.

Lastly, after all border regions were evaluated, the data is sent to a global monitor, evaluating the global requirements. As the global evaluation is not part of this thesis, this step is skipped.

Each of the two monitor types (the local monitor and the neighbourhood monitor) report the results of their virtual grid requirements evaluation back to the monitor management. Currently, this is realized by simple print statements to the terminal which started the monitor management. Note, that in further versions of the monitoring system a more detailed alert and error handling system is needed.

Communication of data sets between the different monitors is assumed to be handled over a secure network protocol. Currently the chosen network protocol is not defined any further, as this is not within the focus of this thesis. However, in future versions of implementation, an appropriate protocol must be chosen to achieve secure communication within the monitoring system.

5.2.2 Virtual Grid

The virtual grid is the part of the monitoring system that actually evaluates the predefined physical- and safety-requirements with the given data sets. Figure 5.2 shows the virtual grid components, which are explained in this section, in an UML class diagram. There are two types of different virtual grid regions in the proposed implementation: the subgrid (`subgrid`) and the border region (`border_region`).

Both regions inherit from the super class `virtual_grid_region`. The subgrid represents a local subgrid Ω_i, while the border region represents the shared power lines between two subgrids and their attached components. In the proposed implementation there are four different types of components: power lines (`power_line.py`), switches (`switch.py`), buses (`bus.py`) and meters (`meter.py`) which are representing the corresponding grid components (see Section 3.1.2) and their properties. All components share with `component` the same super class. A subgrid can consist of all four component types, a border region may only have switches, power lines and meters.

When a local monitor or a neighbourhood monitor is started by the monitor management, the transferred configuration files are used to create a virtual grid representation as the monitored electrical grid. The power lines and the buses of the virtual grid serve to create the topology (e.g. at which buses side

of a power line, a switch or a meter is attached) and the switches and meters save the measured data. Upon receiving a new data set, the values are updated at all meters and switches present in the virtual subgrid or the virtual border region.

Upon receiving a "check requirements" order, the virtual subgrid and the virtual border region start checking each defined requirement on their virtual components. In the proposed implementation, the functions checking the requirements are kept close to the mathematical definition of each requirement as shown in Section 3.2. As an example the requirements check for Requirement 3 N is given in pseudo code in Listing 12. The short function checks the following: for all power lines of the border region it must be ensured that if the line has at least one switch which is currently "open", no meter on that power line may receive any current.

```
1  for line in all_lines:
2      for switch in line.get_switch:
3          if switch is "open"
4              for meter in line.get_meter:
5                  if meter.get_current > 0:
6                      ALERT!
7                  end if
8              end for
9          end if
10     end for
11 end for
```

Listing 5.1: Function to evaluate REQ 3N on all power lines of a border region, written in pseudocode.

A flow chart characterising the program flow within a monitor sending the updates to the virtual region and the following evaluation is shown in figure 5.4.

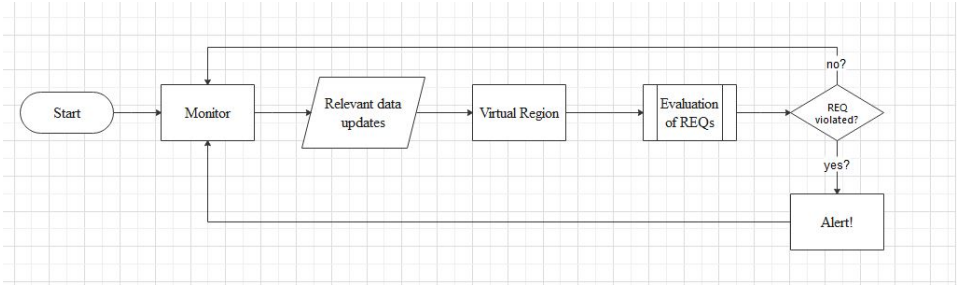

Figure 5.4: Idea of the procedure between a monitor and a virtual region.

Note, that both virtual regions contain additional helper functions to aid a better data handling within the virtual regions. Furthermore the proposed implementation is designed to easily support the addition of further requirements as well as further component types.

5.2.3 Connection to the Testbed

The proposed implementation of the monitoring systems has the capabilities to deal with two different types of input streams: **A)** direct reading from Modbus/TCP servers and **B)** reading from CSV files. Since the proposed implementation covers two subgrids, two Modbus/TCP clients or two CSV files are needed. In case of more subgrids, the number of clients or input files will change, accordingly. In both cases the same information is expected content-wise: an input stream of sensor reading updates matching the corresponding configured subgrid.

A) Since the Mosaik RTU simulator offers the reading from its managed sensors via a Modbus/TCP Server, the first type of input connection was established in the implementation. In this case, the proposed implementation starts two Modbus/TCP Clients that are connected to the Modbus/TCP Server from the Mosaik co-simulation. Since the clients are fitted with the port to their subgrid in the simulation, they are part of each local monitor in the simulation. When the monitoring algorithm starts, both clients read the respective coils from their servers and parse them into update arrays for the virtual subgrid representation. This way, an update array for all virtual switches, the measured voltages for all virtual meters and the measured currents for all virtual meters are created and then their values are updated respectively. Afterwards, when all virtual sensory readings were updates, the next sequence of requirement checks can be started. Upon a new reading request, the two Modbus/TCP Clients will again contact the server and retrieve new updated values.

B) The second option for the input stream, via CSV files, is mostly used to preserve the data once generated by the simulation and makes the regression testing accessible. For this, it is necessary to create two CSV files which contain sensory data readings from the simulation on beforehand. This can be done e.g. by the **helper program** mentioned in 4.2.2. Basically the **helper program** does the same as the two clients indecently do in the first case. Here, it creates two clients which connect to the running simulation and then saves their sensory readings to two CSV files. The implementation opens these pre-generated CSV files and reads them line by line. Each line contains the sensory data updates which, in the other case, would have been directly read from the Modbus registers. The read input line is parsed to matching update arrays, and the virtual components are updated respectively. This way, no direct real-time connection between the Mosaik simulation and the implementation is needed anymore and technical problems like short delays in the simulation due to the scheduling of the used machine can be mitigated. Additionally, the files can always be re-used for regression testing, since live-manipulation or real-time attacks against the simulation are not needed anymore. Note, that both sorts of input streams need specific knowledge about the underlying topology of the simulation to correctly parse the sensory update values.

6 Evaluation

The Evaluation chapter is organized as follows: In a first section the presented scenarios types from Section 4.1.3 are revisited and it is evaluated how the implemented monitoring system could handle them. This continues in Section 6.2, where considerations about which types of attacks against the (simulated) grid can be detected at which stage of the monitoring are given. In Section 6.3 the insights gained from the scenario evaluation and the attack types are brought to a broader perspective, evaluating the hierarchical approach. In the last sections of this chapter, both the testbed (Section 6.4) and the proposed implementation (Section 6.5), are reviewed from a technical point of view regarding of their limitations and restrictions.

6.1 Scenario evaluation

This section will revisit the scenario types that are defined in Section 4.1.3 and the results from their evaluation are presented. To do this systematically, first the scenario type and the expectation is repeated. Then, the (typical) manipulation that happened in this scenario type is shortly summarized. At last, the results and insights gained from the monitoring of this scenario type are discussed in more detail.

For each scenario type, two input files (one for each subgrid) are preserved for regression testing, each containing 20 sets of data readings. Further, the README file in the matching directory contains information on which manipulations were done in the scenario types 2-4.

Scenario type 1: Normal grid operation

Expectation:
The monitoring system should not detect any problems, all checked physical and safety requirements should be "OK".

Manipulation:
No manipulation happens in this scenario type.

Results:
In general, the monitoring of the scenario type 1 input files, leads to no unexpected alerts. Additionally, during the development phase, this scenario helps to find bugs within the monitoring system, especially in the communication

```
PS V:\Uni\Informatik\Masterarbeit\Code\Implementation> python .\monitor_managment.py
Creation and initialization of monitors:
[Local Monitor 0]: Creation successfull.
[Neighbourhood Monitor 0]: Creation successfull.
[Local Monitor 1]: Creation successfull.
[Neighbourhood Monitor 1]: Creation successfull.
---------------
Starting monitoring from historical files.
---------------
Connected locals to significant other neighbourhood monitor.
---------------
Starting the monitoring...

Data set 0
REQ 2 L OK.
REQ 3 L OK.
REQ 7 L OK.
REQ 8 L OK.
[Neighbourhood Monitor 0]: Recieved Updates from the local field station
[Neighbourhood Monitor 1]: Recieved Updates from the neighbouring field station
REQ 2 L OK.
REQ 3 L OK.
REQ 7 L OK.
REQ 8 L OK.
[Neighbourhood Monitor 1]: Recieved Updates from the local field station
[Neighbourhood Monitor 1]: Recieved both parts of the information set.
 Starting to check the neighbourhood requirements.
REQ 3 N OK.
REQ 4 N OK.
[Neighbourhood Monitor 0]: Recieved Updates from the neighbouring field station
[Neighbourhood Monitor 0]: Recieved both parts of the information set.
 Starting to check the neighbourhood requirements.
REQ 3 N OK.
REQ 4 N OK.
```

Figure 6.1: Excerpt terminal output of the evaluation of the first data set from Scenario type 1.

between the different monitors, as the virtual grid has already been tested with additional tests before. Exemplary, Figure 6.1 shows the successful start up of the monitoring system with the data of scenario type 1 and the evaluation of the first data sets.

However, during the process it becomes obvious that the simulation and delays within the testbed largely affect the quality of the generated data. The used grid topology plays an important role as the RTU simulator reads the sensors in ascending orders. Therefore, two sensors, e.g. sensor 4 and sensor 5 which are present at the same bus show the same voltage, as expected, but sensor 7 which is present on the other side of the power line to which sensor 4 is attached, might show a slightly different voltage. Especially with a large gap (numerical) between the naming of two sensors (e.g. sensor 1 to sensor 42) this becomes obvious and leads to alerts when no manipulation has happened.

This effect mostly is a problem while reading the current as the voltage remains more stable.

To prevent such *false positives* or false alerts, additional rounding is introduced to the monitoring system which wasn't used within previous prototypes of the proposed implementation. Of course, rounding in this context is not desirable as this might disguise actual attacks, therefore it is used as minimal as possible. Additionally, the REQ 1 and 4 have been deactivated for the monitoring of the local scope. Due to the used gird architecture, the simulation delay does not affect the voltage evaluation of REQ 4N fortunately, therefore this is kept active. The awareness of this problem now can be used in further versions of the proposed implementation and testbed, to achieve a better simulation or investigate how other grid architectures can minimize the effect, that the simulation itself causes alerts due to the simulation order of the sensors. As REQ 1 and 4 can be successfully tested also with manual test data instead of simulated test data, this seems to remain a limitation of the testbed rather then the proposed implementation.

Scenario type 2: Faults in the local scope

Expectation:
The monitoring system should detect these faults together with the correct requirement they violate while evaluating the *local* scope.

Manipulation:
Individual readings from sensors within both subgrids are manipulated separately, so that they exceed certain set points and do not match the other readings anymore. The manipulation is only contextual and no malformed input is injected. The test data for this scenario type consists of single manipulation rows alternating with non-manipulated rows to prevent intersecting effects.

Results:
The monitoring system is able to detect of all the undertaken manipulations. While checking the one row of input data, the matching local monitor detects the deviation and triggers an alert. Respectively, the other local monitor detects no alert on non-manipulated data. Exemplary, Figure 6.2 shows one alert that is issued by the monitoring system during the evaluation of scenario type 2 using the *detailed print* option of the implementation, which reports the exact component that caused the violation. As the manipulations only targets inner sensor of both subgrids, no alerts are triggered by the neighbourhood monitors. Each requirement that is currently available and reasonable to check without to large rounding, is triggered at least once within the test data for both subgrids.

```
Data set 3
Alert! REQ 2 L violated by b19
REQ 2 L  OK for b20
REQ 2 L  OK for b22
REQ 2 L  OK for b23
REQ 3 L  OK for branch_38
REQ 3 L  OK for branch_30
REQ 3 L  OK for branch_28
REQ 3 L  OK for branch_27
REQ 7 L  OK for sensor_13
REQ 7 L  OK for sensor_14
REQ 7 L  OK for sensor_16
REQ 7 L  OK for sensor_17
REQ 7 L  OK for sensor_18
REQ 7 L  OK for sensor_15
REQ 7 L  OK for sensor_19
REQ 7 L  OK for sensor_20
REQ 7 L  OK for sensor_21
REQ 7 L  OK for sensor_22
REQ 7 L  OK for sensor_23
REQ 7 L  OK for sensor_24
REQ 8 L  OK for sensor_13
REQ 8 L  OK for sensor_14
REQ 8 L  OK for sensor_16
REQ 8 L  OK for sensor_17
REQ 8 L  OK for sensor_18
REQ 8 L  OK for sensor_15
REQ 8 L  OK for sensor_19
REQ 8 L  OK for sensor_20
REQ 8 L  OK for sensor_21
REQ 8 L  OK for sensor_22
REQ 8 L  OK for sensor_23
REQ 8 L  OK for sensor_24
[Neighbourhood Monitor 0]: Recieved Updates from the local field station
[Neighbourhood Monitor 1]: Recieved Updates from the neighbouring field station
REQ 2 L  OK for b27
REQ 2 L  OK for b26
REQ 2 L  OK for b25
REQ 2 L  OK for b24
REQ 3 L  OK for branch_34
REQ 3 L  OK for branch_37
REQ 3 L  OK for branch_33
REQ 3 L  OK for branch_32
REQ 7 L  OK for sensor_1
```

Figure 6.2: Excerpt terminal output of the evaluation of the manipulated data set from Scenario type 2 using the detailed print output option.

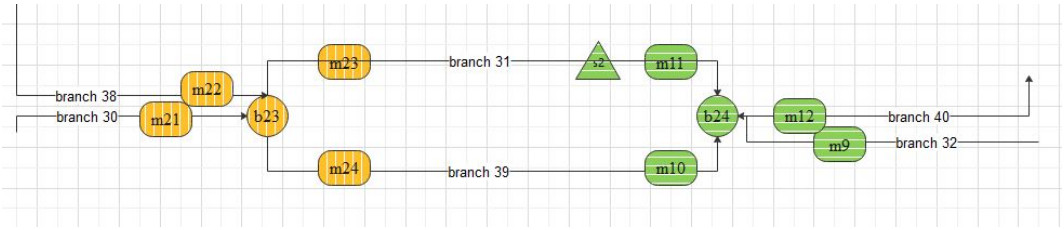

Figure 6.3: Topology of two of the connecting power lines between subgrid 0 and subgrid 1.

Scenario type 3: Faults in the local and neighbourhood scope

Expectation:
The monitoring system should detect the faults on *inner* and *shared* components. Depending on the violated requirements, they should be detected in both, the *local* scope (at least on one side) and in the *neighbourhood* scope of both subgrids.

Manipulation:
The manipulations for this scenario targets components within the border region between the two subgrids. As for scenario type 2 only single data cells are manipulated at once to prevent intersecting effects. The manipulation is only contextual and no malformed input is injected.

Results:
The monitoring system was is to detect all manipulated data sets. As each component belonging to the border region also belongs to one of the subgrids, some of the manipulations are detected by both, the local monitor of the subgrid of the component, and the neighbourhood monitors. The two neighbourhood monitors always deliver the same evaluation of the data. Each requirement that is currently available and reasonable to check without to large rounding, is triggered at least once within the test data for both subgrids.

To further analyse the scenario type, Figure 6.3 illustrates the manipulated part of the border region topology. In one data set of this scenario type, Switch s2, belonging to Subgrid 1, is manipulated from *closed* to *open*. However, the current reported from Meter m_{11} and Meter m_{23} is not manipulated i.e. both meters attached to Branch 31 still reported current. The violation of REQ 3, is only detected by the neighbourhood monitors and not by the local monitor of Subgrid 1.

This evaluation of the local monitor 1 is correct, as is does only check REQ 3 for inner power lines and not for power lines connecting two grids. Therefore, Branch 31 is only checked by the neighbourhood monitor. However, the violation could have already been detected on local level as the Meter m_{11}, present at Subgrid 1 and attached next to Switch s_2 still reported current. An

alternative evaluation, especially the usage of preliminary evaluation is further explored in Section 6.5.3.

Scenario type 4: Faults in the neighbourhood scope only

Expectation:
The monitoring system should be able to detect faults in the *neighbourhood* scope even when the *local* evaluation finds no implausible data sets.

Manipulation:
For this scenario type two kinds of manipulation are preformed: At first, again, single data cells are exchanged. This included e.g. the manipulation of Switch s_2 to *open*. The grid topology is still the same as in Figure 6.3. Different from Scenario type 3, all other components on the same grid are manipulated to match the manipulated switch, i.e. the current reported from the meter m_{11}, next to the switch is set to 0. However, the data on the other side of the border region is not adapted. For both tested requirements, REQ 3N and REQ 4N, the data is manipulated at one side of the subgrid to not cause any local alerts, without adapting the other side. The second manipulation is a complete replacement of a data row from Subgrid 0 with another row from Subgrid 0 that previously evaluated without any alerts. This historical data then exchanges three other rows within the data set. Therefore, instead of the actual input data, Subgrid 0 receives three times a replayed data row.

The manipulations are done separately to prevent intersecting effects.

Results:
The monitoring system is able to detect each manipulation and the two neighbourhood monitors agree on their evaluation. The local monitors can not detect any of the performed manipulations. The different types on manipulations represent two attacks on the electrical grid that cannot be detected locally. The first only affects a small part of the grid (e.g. a bus) where the attached meters deliver manipulated results, the second kind represent the situation that a complete subgrid is replayed. However, the integration into the complete grid via the border regions helps to detect this attacks, since the values reported at that side of the subgrid do not match the values reported at the other side of the border region. Of course, the monitoring system cannot tell which side is manipulated, as it can only realize that a requirement is violated. This evaluation still has do be done by adding further tests or a by passing this task to a human.

6.2 Detectable attack types

The following section focuses on the attack types that can be detected by the monitoring system. First of all, from a technical perspective, all attacks that are pictured in the scenario types above relayed on the manipulation of at least one data cell up to a complete input row. In a real world scenario a manipulated input may origin from two different aspects: The first one being a (physical) manipulation of the component directly, causing a unintended reaction from the components e.g. by manipulation the windings, hard ware settings or similar. This of course can also be an unintended result when the component is broken or worn off. The monitoring system however, can detect this too and spare further damage when the broken component gets repaired or exchanged quickly. The other way of manipulation is an exchange of data within the communication. In the scope of this thesis it is assumed that local communication is possible in a secure way, even though this might not always be true. As seen in the background chapter, Modbus/TCP or frequently used protocols are not always secure and lack security mechanisms. An intruder, who knows about the used protocols and can eavesdrop on previously send messages could re-send an earlier message, or manipulate some data, interrupt a normal message and exchange it with his manipulated one.

No matter which way an intruder uses, the effect is always that the manipulated data cell or row, deviates from its original, reaches the monitoring system, like simulated in the four scenario types. The proposed monitoring approach is able to detect all manipulations present in the four scenario types for the used grid topology. Nevertheless this does not automatically mean that the proposed monitoring system can detect every manipulation. To achieve more completeness in this regard, further requirements need to be implemented and a good balance for the accuracy needs to be found. This also includes an evaluation of the global scope, too.

In accordance with the four scenario types, the proposed hierarchical monitoring approach is able to detect three types of attack scenarios. An illustration outlining the attacked components for the three attack scenarios (A, B and C) is shown in Figure 6.4:

- **A Manipulation of components within a subgrid**
  This sort of attack restricts to inner components of a subgrid, as they can already be detected by the local monitoring approach.

  It is crucial to detect these manipulations already at the base and at a local level, as they might not be detectable by further evaluations since no shared components are involved. Note, that such manipulation may lead to wrongly issued commands and lastly, result in physical damage to the components.

- **B Manipulation within the border region**
  Without the exchange of information between two field stations a ma-

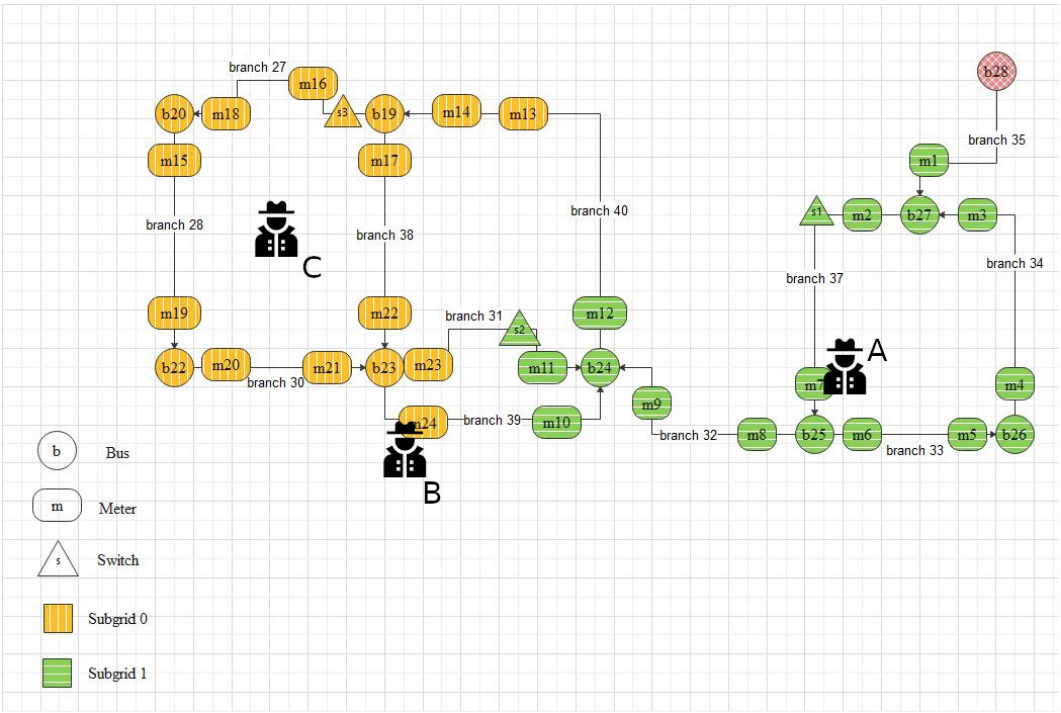

Figure 6.4: Attack points within the electrical grid which are detectable with the proposed approach.

nipulation within a border region, i.e. a manipulation at the components attached to the shared power line, are not detectable. Some of the manipulations can be revealed with a matching requirement that get evaluated even if not all needed data is present. But some requirements are simply not evaluable without the information from the neighbourhood substation.

Using the hierarchical approach, including an exchange mechanism for the needed information, these attacks can be detected. As an attack on the shared power line between two subgrids can, in a bad case, partially detach one of the subgrids from the rest of the main grid, it is utmost crucial to supervise the border regions.

- **C Manipulation which is only visible in the border region**
  As scenario type 4 successfully demonstrated, there are attacks on one subgrid that simply are not detectable by this subgrid alone. This includes e.g. a replay of historical data, either partially or for the complete subgrid, which does not lead to local alerts when evaluated.

  As the intruder model contains these possibility, it must be assumed that an intruder is be able to exchange the complete data traffic within that substation or the data traffic communicated from the subgrid to the outside (e.g. the network operator or the monitoring system). However,

if the intruder is only able to exchange parts of the communicated data traffic within the electrical grid and not of every substation, it is highly likely that the neighbourhood monitors are able to detect the exchange. In case of such an attack, the reported data from both sides of the border region simply do not match. Of course the monitoring system can not decide which side is manipulated, but the violation of requirements itself is detected quite fast at every border region which has a non-attacked field station.

The proposed monitoring system can not detect further attacks which relay on other tactics, like e.g. DoS attacks. Therefore, it can only serve as an addition to IPDS.

6.3 Review of the model

This section reviews the underlying theoretical model which is introduced in Chapter 3 and the hierarchical approach behind it. As demonstrated in the previous section, the proposed model and its implementation are capable to detect manipulations within the simulated electrical grid both in the local areas as well as in border regions connecting two local field stations. Even though the evaluation of the formulated requirements is able to detect all the manipulations that were presented within the attack scenarios, it is reasonable to assume that further requirements could cover manipulations which currently are not detected. Such further requirements include other physical laws, safety regulations regarding existing or new components.

The proposed model supports a high flexibility and expandability with respect to the available components, realisable grid topologies and used security requirements. Therefore, it is suited to be used and adapted in future research to encourage further developments regarding (smart) grid organisation. Especially the focus on evaluating requirements on a fragmented grid, mostly at a local, field station level, but also on neighbourhood level, matches the current development of DEM. The variable subgrids sizes, that do not necessarily need to consists one complete field station, as used in the scope of this thesis, can be dynamically used to suit the needs of a small LV micro grid. Depending on current developments within the micro grid organization it might be reasonable to later define special types of subgrids within the model, to reflect these developments.

Furthermore the three scopes the model includes (*local*, *neighbourhood* and *global*) could be further adjusted to meet the current trends. It is reasonable to add another level between the current neighbourhood and the global level

which represents an amalgamation between a few close subgrids to a distinct community. This way, the hierarchical evaluation approach is be strengthened and the fragmentation of the gird supported to aid the decentralized management. These community regions could consists of a distinct subgrid topology and aid both, the security within the community as well as the fast exchange with other similar community regions.

6.4 Review of the testbed

In this section the chosen testbed approach is reviewed. Additionally limitations and resulting restrictions of the testbed are discussed.

First of all the proposed testing approach is only a first step to evaluate the feasibility of the proposed model and implementation. To really evaluate its feasibility for a real world application, testing on real hardware is out of question. However, the proposed testbed including the Mosaik Co-Simulation framework can help to gain first insights. Of course, the number of tested data sets plays a critical role to evaluate the feasibility of the testing approach. In the proposed evaluation the number of used data sets was chosen, with both a proof of concept and easy understanding for a reader in mind. Further evaluations, especially those targeting the scalability and the performance, need to take larger data sets into account.

Currently the simulation implemented with Mosaik only covers some of the components that are available in the theoretical model. Therefore, the testing of the missing components is not possible within the implementation yet. To achieve a better evaluation, additional simulators for the Mosaik Co-Simulation framework need to be implemented, to simulate interlocks, fuses and protective relays and transformers. Furthermore, there are currently no grid specific configurations available to evaluate e.g. REQ 11a and REQ 11b in a reasonable way. Such expansions of the simulation can help to build more diverse electrical grids and to check against more complicated attacks and requirements. In addition, the measurement within the simulation itself, or if applied to real hardware, are currently not directly evaluated. In case of a larger simulation (with Mosaik or another co-simulation framework) delays in the computation might play a critical role while evaluating for the monitoring system. As discussed in Section 6.1 the inaccuracy the computation already affects the proposed implementation and leads to the need of rounding or temporary deactivation of requirements.

Table 6.1 gives an overview which of the in Section 3.2.1 defined requirements currently can be tested with the proposed simulation configuration and the

Requirement	Evaluation scope	Testable with the testbed?
REQ 1 L	local	yes
REQ 2 L	local	yes
REQ 3 L	local	yes
REQ 4 L	local	yes
REQ 3 N	neighbourhood	yes
REQ 4 N	neighbourhood	yes
REQ 5a L	local	yes
REQ 5b L	local	yes
REQ 6a L	local	no, no transformers directly available in the proposed testbed
REQ 6b L	local	no, no transformers directly available in the proposed testbed
REQ 7 L	local	yes
REQ 8 L	local	yes
REQ 9 L	local	no, no fuses and protective relays directly available in the proposed testbed
REQ 10 L	local	no, no fuses and protective relays directly available in the proposed testbed
REQ 11a L	local	no, no transformers directly available in the proposed testbed
REQ 11b L	local	no, no transformers directly available in the proposed testbed
REQ 12 L	local	yes
REQ 13 G	global	no, because out of scope - but generally, yes
REQ 14a L	local	no, no RTU configuration available, but generally, yes
REQ 14b L	local	no, no RTU configuration available, but generally, yes
REQ 15a L	local	no, no interlocks directly available in the proposed testbed
REQ 15b L	local	no, no interlocks directly available in the proposed testbed

Table 6.1: Physical and safety requirements, their matching scope and if they are testable with the testbed.

available simulators. For the current development stage of the tested approach, it is sufficient to evaluate the possible requirements, however, future work should aim to expand the number of testable requirements.

An additional limitation concerning the complexity of the simulation is the limited Modbus TCP coils and registers of the Modbus/TCP Server representing the RTU per subgrid. As the Modbus/TCP protocol only offers a given number of 16 bit holding registers, which in case of the Mosaik RTU simulator store either the measured current or the measured voltage of a meter, the number of meters a subgrid can contain is limited. An alternative solution is to allow more than one RTU simulator for one subgrid in this case.

Another big limitation to the current testbed is the restricted monitoring of network traffic. In the proposed proof-of-concept work, the input for the monitoring system or rather the local monitors as the entry point of the monitoring system, consists of either reading the coils and registers at the local Modbus/TCP Server directly or re-reading the information from saved data files. An adaption of the testbed for example by connecting it to a larger-scale IDS like Zeek (formerly known as Bro) as proposed by Chromik ([3]) allows further evaluation form a better network traffic perspective.

Furthermore the current possibilities to attack the simulation by either overwrite Modbus/TCP coils and registers directly or exchange small snippets that simulate a PLC with malfunctioning version are quite inconvenient and very detailed. More accessible approaches aid further testing of the implemented monitoring system.

6.5 Review of the implementation

This section reviews the proposed implementation. At first, the solution requirements stated in Section 5.1 are reviewed. Following, technical limitations to the implementation are discussed and additional improvements to these limitations are proposed. Finally, in Subsection 6.5.3, two design decisions that were made during the implementation and their alternatives are explored in more detail.

6.5.1 Solution requirement fulfillment

Section 5.1 defined nine functional and non-functional requirements for the proposed implementation. In the following, these nine requirement and their degree of fulfillment are shortly reviewed.

Model coverage

The proposed implementation covers the necessary components to evaluate the requirements re-visited in Section 6.4, which are testable by the current state of the testbed. Furthermore, additional components can be easily integrated into the virtual grid to expand the evaluation.

Input data

The proposed implementation is able to accept both, a live connection to the testbed as well as CSV files containing appropriate simulation data as input. The test data matching the described scenario type evaluation from Section 4.1.3 is provided and further explanations according to the conducted manipulations are provided in the corresponding README file.

Fault tolerance

The implementation catches all manipulations which are present within the test data. However, the accuracy of live data may lead to false-positives, as mentioned in 6.1.

Usability

The proposed implementation uses multiple outputs to the command line to inform the user about the current progress of an evaluation. The information always includes which part of the monitoring system issued the output. Further information on how to start and operate the implementation are given in the corresponding README files.

Documentation

Each directory contains a matching README file which explains the directory and its content. Corresponding top-level README files contain further

information about how to manipulate the implementation and concrete command line commands on how to start it. The provided source code is well documented.

System requirements

The implementation runs on a normal office computer without any large delays. Additionally, the testbed contains a configuration entry to slow down the simulation for a better possibility to observe the values.

Code language and external dependencies

The proposed implementation is done with Python 3.7. Each version number of needed external dependencies is documented.

License

No commercial nor proprietary software was used while implementing the monitoring system or the testbed. All used Mosaik components fall under the GNU LGPL v2.1 license (see Table 4.1). However, a publication of the source code is currently not planned.

Operational safety, installation, availability, GDPR

The needed information to install and use the implementation are given in the matching README file. No further discussion of this topic is needed.

6.5.2 Limitations

The first, and probably most prominent, limitation of the proposed implementation is the restriction of the monitoring to two subgrids and one shared border region between these subgrids, resulting in two local and two neighbourhood monitors. However, as explained in Section 5.2.1, the general monitor management cam be expanded to monitor additional subgrids and matching border regions. In accordance with that, a reasonable evaluation regarding the scalability and the performance for more subgrids or in general a complete electrical grid is not yet possible. In this context, a big question is be how many neighbours each subgrid has, i.e. if it is either a loosely mashed subgrid where each subgrid only shares border regions with a few other subgrids or if each subgrids is heavily connected to many other subgrids. New findings in this direction still have to be developed, mainly following the new trends arising from DEM and grid organization in general.

Another limitation is that the proposed implementation focuses only on the evaluation of data sets. A pre-evaluation of commands issued to the RTU within a subgrid is currently not done. However, this approach already has been explored by Chromik [3] for local evaluation and can be added to the implementation in future work. In this case the intercepting of the issued

commands needs further access to the network traffic e.g with Zeek (formerly known as BRO) as already mentioned in Section 6.4.

Lastly, it should be stressed that the proposed implementation is built upon the assumption that a secure communication between each part of the monitoring system is possible. For example, this includes a secure communication between a local monitor of subgrid Ω_i and the neighbourhood monitor of subgrid Ω_j which may be far away, geographically. In future work, an appropriate communication mechanism for this communication has to be be found. It should be noted that also this communication channels, too, might be a possible entry point for an attack against the monitoring system itself. Additional security mechanisms in this regard can include e.g. a comparison between the evaluation results of each pair of neighbourhood monitors after each evaluation to minimize the risk that one was corrupted.

6.5.3 Design Decisions

The proposed implementation of the monitoring system focused mostly on simplicity and readability, without hard requirements regarding time or space complexity. This focus led to a few design decisions to support building a good proof of concept work to evaluate the feasibility and limitations of the proposed model. In the following two of these design decisions and their alternatives are discussed. This discussion gives insight on further possible improvements or additional research.

Pre-evaluation of requirements

The first design decision is to evaluate each requirement in its matching scope. In the proposed monitoring system each type of monitor evaluate the corresponding scope: the local monitor evaluates the requirements from the local scope, the neighbourhood monitor evaluates the requirements from the neighbourhood scope and, if it would have been implemented, the global monitor implements the requirements from the global scope.

Against the odds, for some requirements an evaluation at a point earlier than their indicated position is partially possible. This effects mostly target requirements with conjunctions. A good example for this phenomenon is REQ 3N:

$$\textbf{REQ 3 N: } \forall\ L_i \in \mathcal{L}_a \wedge L_i.\text{or} = \text{(a,b)}\ \exists\ S_j \in L_i.\text{S:}$$
$$S_j = 0 \rightarrow \forall\ M_k \in L_i.\text{M} : M_k.\text{I} = 0$$

An alert should be triggered if a power line connecting two subgrids has an open switch and one of the meters attached to the power line is receiving current. Currently, this requirement is evaluated within the *neighbourhood* scope. In this evaluation phase the neighbourhood monitor has all the information about the state of the switch and which current is measured at each meter. Therefore

the neighbourhood monitors can easily evaluate the complete requirement. However, each of the involved local monitor could have evaluated the complete requirement on beforehand, because the information about the meters on the power line which belongs to the other subgrid, are missing. If the switch is present at the side of subgrid Ω_i and its state is currently the *open*, the local monitor could trigger already an alert, if at least one of the meters at its side, receives current. Of course, the neighbourhood monitor still would need to evaluate the requirement with the additional information from subgrid Ω_j.

As a consequence, it is possible to reduce the calculation load on the neighbourhood monitor and the global monitor by shifting the evaluation partly to the local monitor. To do this, a corresponding formulation for the partly evaluation of neighbourhood and global requirements is needed. In case of a real-world application or a much larger simulation grid this might be useful to accelerate the evaluation and, especially, receive faster notifications of possible alerts.

Parallelization of scope evaluation

The second design decision is to parallelize the different scopes of the evaluation. In the proposed implementation, a sequential approach was chosen: a new set of input data is first evaluated within the local scope, then sent to the neighbourhood monitor and evaluated there and afterwards is be sent to the global monitor to evaluate requirements within the global scope.

It is possible for the local monitor to directly sent its newly received input data to its neighboured monitors and the global monitor, so the three monitor types could evaluate the new data concurrently. In case of a truly distributed monitoring system, this accelerates the evaluation process by far. Again, in the scope of this thesis the sequential approach is chosen for the sake of simplicity. Additionally, a violation of a requirement, which may be caused by a malfunctioning device or an intruder, is detected as local and as early as possible. The tracing, which component(s) caused the violation is easier to understand as if it is detected e.g. in the global scope first because of different evaluation order and speed. Concerning content, the sequential approach prevents faulty reading to reach more global parts of the monitoring system and remains local. Depending on the available resources it might be reasonable to not evaluate sensory readings further which already contain a known violation of a requirement.

7 Conclusion

In this concluding chapter of the thesis the outcome are reviewed. In a first section the results of each chapter are summarized. Following the summary, Section 7.2 revisits the research questions from the introduction. Afterwards Section 7.3 points out limitations to aspects of this thesis and possible improvements. Additionally, thoughts and ideas on possible future work are presented in Section 7.4.

7.1 Summary

Electricity plays a critical role in the modern society as we rely on both the provided quality (e.g. a steady frequency) as well as the nearly permanent availability in many different aspects of our lives. Therefore an appropriate degree of security to protect both the quality and the availability is indisputable. With this motivation in mind, Chapter 1 proposes a hierarchical and process-aware approach to monitor SCADA networks operating electrical grids.

Chapter 2 highlights the two underlying fields that are from a necessary background: the more physical-oriented world of electricity and energy management on the one hand, and SCADA networks as the ICS controlling electrical grids on the other. This includes deeper insights into the currently emerging shift from a centralized to a decentralized energy management approach as well as the challenges of monitoring SCADA networks in contrast to standard IT software. An overview of known historical incidents, their reasons and current strategies and research to prevent such incidents complemented the chapter. Overall, this background stresses how important it is to include both aspects, the physical process itself and the network operating the electrical grid, to create a process-aware monitoring approach. Consequently, the proposed context-aware approach focuses on attacks against an electrical grid, which seem unobjectionable from a syntactical point of view, but indicate a harming of the physical components from a contextual perspective nevertheless.

Upon this foundation, Chapter 3 defines an theoretical model of an electrical grid which emphasized a fragmentation into subgrids. Additionally, physical- and safety-related requirements are defined which can serve as an indication whether the current state of the electrical grid seems to be contextual plausible. The defined requirements can be sorted according to their needed scope of

knowledge and this way be evaluated as local and as directly as possible. It is defined which information need to be communicated to evaluate the further scopes on a neighbourhood with neighboured subgrids and on global level. Together with the fragmentation into subgrids, the requirements and their corresponding evaluation scope build the basis for a hierarchical, distributed monitoring of an electrical grid.

Since testing the proposed monitoring approach and its implementation on real hardware is not practical, Chapter 4 describes an appropriate testbed and testing approach. This includes the definition of a test objective and four scenario types to test the feasibility of the proposed monitoring system. This assists the evaluation of which types of attacks and discrepancies can be detected compared to previously proposed local process-aware monitoring approaches. Furthermore, an electrical grid simulation based on the co-simulation framework Mosaik is presented, which supports the fragmentation into multiple subgrids corresponding the proposed theoretical model. Following the description of the testbed, Chapter 5 explains the proposed implementation of the IDS. The chapter explains with which technical requirements in mind the implementation has been developed and how the different parts of the distributed monitoring system communicate. Additionally, the direct evaluation of the defined physical- and safety-related requirements which were defined in Chapter 3 is described in more detail. Note, that the proposed implementation is as proof of concept work focused on simplicity to evaluate the underlying model and monitoring approach. In addition, a live-connection to the testbed as well as the possibility to use static input files for regression testing and retraceability of the evaluation is presented.

Afterwards, Chapter 6 revisited the defined test scenario types and the proposed theoretical model, the testbed and the implementation. Furthermore, the feasibility of the proposed monitoring approach is shown in this chapter. It is evaluated which types of attacks can be detected with respect to both, the hierarchically-distributed and the process-aware elements of the monitoring system.

7.2 Review of Research Questions

In this section detailed conclusions to the research questions posed in the introduction are drawn. Further, below each research question is visited again and shortly answered.

Research Question 1: How can the (local) sensory readings from the underlying physical processes benefit the overall security of the SCADA network operating the electrical grid?

It is shown that previous researchers in the field of SCADA networks operating electrical grids mostly focused on syntactically-wrong and therefore detectable attacks. However, it has to be assumed that an intruder is able to infiltrate the system and exchanging syntactically correct information e.g. sensory data. This thesis followed the process-aware approach and issued an exemplary list of physical- and safety related commands that are used to evaluate if the state of the electrical grid can be considered as safe and stable. In case of a implausible exchanged sensor value the proposed requirements will most likely get violated and the system alerts that either a material failure or an intrusion has happened.

Note that this approach of monitoring SCADA networks, which operate electrical grids, can only serve as an addition to other IPDS. Further security mechanisms are always needed to protect against e.g. DoS attacks.

Research Question 2: How can a hierarchical, distributed evaluation of safety requirements within an electrical grid be organized?

To achieve a hierarchical, distributed evaluation, firstly a fragmentation of the electrical grid needs to be created. The proposed abstract model of an electrical grid focuses on a fragmentation into different subgrids which are connected with power lines and share a so-called border region. Each component attached to a power line within a border region belongs to either of the neighbouring subgrids. Note, that the size and structure of the subgrids is highly flexible. In the scope of this thesis they are chosen to include roughly one field station and its surrounding components. However, note that subgrids in general can be of a completely different shape.

In accordance with the fragmentation, the defined physical- and safety-related requirements are split with respect to the knowledge scope that is needed for their evaluation. Consequently, there is one scope of requirements that can be evaluated within a subgrid (the *local* scope), one for the requirements which focus on shared border regions between neighbouring substation (the *neighbourhood* scope) and a *global* scope. Each subgrid can evaluate the requirements from the *local* scope individually and in a distributed way. After

an exchange of information, the *neighbourhood* scope can be evaluated in a distributed way at each substation, too. This thesis proposed one neighbourhood monitor per shared border region which is locally available at each subgrid. The evaluation of global commands was not evaluated during this thesis, but can be added following the same approach or at a central position.

Research Question 3: Which additional insights can be gained from using information from neighbouring field stations?

The data exchanged with neighbouring field stations (i.e. with at least one shared power line between them) supports with two aspects: first of all the shared border region, which might contain power lines, switches, meters etc. can be supervised by the monitoring system by evaluating the neighbourhood requirements. Within a truly local approach the evaluation of requirements in the border region is not possible. As these regions connect the different parts of the electrical grid, it is crucial to supervise them to prevent larger damage spreading in case of any malfunctioning.

Secondly, the evaluation of a shared border region can help to detect whether one of the participating subgrids is manipulated. For example, if the complete local traffic of a substation is exchanged with historical traffic which shows no faults, the local evaluation mostly likely does not detect this, since the local requirements are not violated. However, the manipulated data exchanged about the border region would not match the data measured on the other subgrid and an alert can be triggered.

7.3 Limitations

Even tough a few technical limitations to the testbed already have been discussed in Sections 6.4 and 6.5.2, this section reviews them from a broader perspective.

Currently the proposed testbed and the proposed implementation both are quite limited regarding to the implemented component types and regarding to the number of implemented components, too. To achieve a small prototype which can serve as a proof of concept for the proposed theoretical model, the testbed and the corresponding implementation are sufficient. However, it must be noted that the simulated electrical grid with two monitored subgrids and not every component type present which is defined in the model, only can give a first insight into the feasibility of the proposed monitoring approach. Further improvements and expansions, both with the simulated and monitored components as well with the number of subgrids and their different topologies, can

give even more detailed insights on how to develop a process-aware monitoring system for large real world applications.

Another limitation of the proposed approach is that currently commands to the subgrids can not be intercepted and evaluated prior to their execution. Therefore, the current monitoring system can only estimates after a possible manipulated command has been issued, if the execution led to an unsafe state and an alert should be triggered. Such an addition, which e.g. has been developed similarly for the local scope by Chromik in [3], would be an important aspect in future work. Such an improvement can be included by a combination with a large-scale IDS like Zeek (formerly known as Bro) to aid further analysis of network traffic.

7.4 Future Work

Apart from the improvements already proposed to the known limitations in Section 6.4, 6.5.2 and 7.3, there are three aspects which seem to contain promising ideas for possible future work:

A) The development of different subgrid types with special properties **B)** The investigation of different grid topologies and their impact on the kind and number of shared border regions, **C)** The expansion of the hierarchical approach to the global scope.

A) The idea to develop different subgrid types follows the current trends emerging from the energy transition and DEM. Additionally, this approach exploits the flexibility of the defined subgrid model further. It could be useful to formulate different sub-types of the proposed generic subgrid, which could contain e.g. a small part of a LV smart grid which contains a neighbourhood with a lot of PV panels or for example a charging station for electric vehicles. For this sub-types matching physical and safety requirements could be defined which do not find application in a generic subgrid. This approach can aid to better secure critical regions of an electrical grid according to their special needs.

B) As already mentioned, the proposed implementation only features two subgrids and their border region. However, currently new approaches for grid topologies are explored to achieve a better energy management, mostly in distributed ways. Insights from this perspective could be taken more into consideration, like the number of subgrid neighbours and the arrangement with e.g. a ring. This could lead to a better evaluation of the relevant broader regions. Furthermore, especially, within ring structures it would be interesting to examine a monitor using information from more than two subgrids and possibly monitoring properties of the ring itself. By doing this further redun-

dancy could be implemented into the monitoring and mitigate single points of failures within the system.

C) The global evaluation scope is definitely a reasonable research opportunity. Both, from a contextual perspective to evaluate requirements that need knowledge from the complete grid as well as systematical to evaluate the performance and scalability of a central entity could provide more insights to the monitoring system. It might be reasonable, to create multiple global monitors spread over the complete electrical grid to achieve a good monitoring system. Regarding the evaluation of requirements from the global scope, the preliminary requirement evaluation mentioned in Subsection 6.5.3 also might be interesting.

Bibliography

[1] Assante, M.: *Confirmation of a Coordinated Attack on the Ukrainian Power Grid.* https://www.sans.org/blog/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid/. Version: 2016 (Last accessed on March 16, 2021)

[2] ICS-CERT: *Alert (IR-ALERT-H-16-056-01) Cyber-Attack Against Ukrainian Critical Infrastructure.* https://us-cert.cisa.gov/ics/alerts/IR-ALERT-H-16-056-01. Version: 2016 (Last accessed on March 16, 2021)

[3] Chromik, Justyna Joanna: *Process-aware SCADA traffic monitoring: A local approach.* Netherlands, University of Twente, Diss., Juli 2019. http://dx.doi.org/10.3990/1.9789036548014. – DOI 10.3990/1.9789036548014

[4] Chromik, Justyna Joanna ; Remke, Anne Katharina Ingrid ; Haverkort, Boudewijn R.H.M.: Improving SCADA security of a local process with a power grid model. In: *Proceedings of the 4th International Symposium for ICS SCADA Cyber Security Research, ICS-CSR 2016*, BCS Learning Development Ltd., 8 2016 (Electronic Workshops in Computing). – ISBN 1477–9358, S. 114–123. – eemcs-eprint-27159

[5] Chromik, J. J. ; Pilch, C. ; Brackmann, P. ; Duhme, C. ; Everinghoff, F. ; Giberlein, A. ; Teodorowicz, T. ; Wieland, J. ; Haverkort, B. R. ; Remke, A.: Context-aware local Intrusion Detection in SCADA systems: A testbed and two showcases. In: *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2017, S. 467–472

[6] Chromik, Justyna Joanna ; Remke, Anne Katharina Ingrid ; Haverkort, Boudewijn R.H.M.: An integrated testbed for locally monitoring SCADA systems in smart grids. In: *Energy Informatics* 1 (2018), 11, S. 1–29. http://dx.doi.org/10.1186/s42162-018-0058-7. – DOI 10.1186/s42162–018–0058–7. – ISSN 2520–8942

[7] Skrabec, Quentin R.: *The 100 Most Significant Events in American Business: An Encyclopedia.* Santa Barbara, California : ABC-CLIO, 2012

[8] Hoogsteen, Gerwin: *A Cyber-Physical Systems Perspective on Decentralized Energy Management*. Netherlands, University of Twente, Diss., 12 2017. http://dx.doi.org/10.3990/1.9789036544320. – DOI 10.3990/1.9789036544320

[9] van der Klauw, Thijs: *Decentralized Energy Management with Profile Steering: Resource Allocation Problems in Energy Management*. Netherlands, University of Twente, Diss., 5 2017. http://dx.doi.org/10.3990/1.9789036543019. – DOI 10.3990/1.9789036543019. – CTIT Ph.D. thesis series no. 17-424

[10] Cook, John ; Oreskes, Naomi ; Doran, Peter T. ; Anderegg, William R L. ; Verheggen, Bart ; Maibach, Ed W. ; Carlton, J S. ; Lewandowsky, Stephan ; Skuce, Andrew G. ; Green, Sarah A. ; Nuccitelli, Dana ; Jacobs, Peter ; Richardson, Mark ; Winkler, Bärbel ; Painting, Rob ; Rice, Ken: Consensus on consensus: a synthesis of consensus estimates on human-caused global warming. In: *Environmental Research Letters* 11 (2016), apr, Nr. 4, 048002. http://dx.doi.org/10.1088/1748-9326/11/4/048002. – DOI 10.1088/1748–9326/11/4/048002

[11] Climate Change, United Nations C.: *Kyoto protocol to the United Nations framework convention on climate change*. http://unfccc.int/resource/docs/convkp/kpeng.pdf. Version: 1998 (Last accessed on March 16, 2021)

[12] Climate Change, United Nations C.: *Adaption of the Paris agreement*. https://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf. Version: 2015 (Last accessed on March 16, 2021)

[13] Toersche, Hermen: *Effective and efficient coordination of flexibility in smart grids*. Netherlands, University of Twente, Diss., 10 2016. http://dx.doi.org/10.3990/1.9789036541978. – DOI 10.3990/1.9789036541978

[14] (IEA), International Energy A.: *Technology Roadmap - Smart Grids*. https://www.iea.org/reports/technology-roadmap-smart-grids. Version: 2011 (Last accessed on March 16, 2021)

[15] Radoglou-Grammatikis, P. I. ; Sarigiannidis, P. G.: Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems. In: *IEEE Access* 7 (2019), S. 46595–46620. http://dx.doi.org/10.1109/ACCESS.2019.2909807. – DOI 10.1109/ACCESS.2019.2909807

[16] Deka, D. ; Backhaus, S. ; Chertkov, M.: Learning topology of the power distribution grid with and without missing data. In: *2016 European Control Conference (ECC)*, 2016, S. 313–320

[17] Khavari, F. ; Badri, A. ; Zangeneh, A. ; Shafiekhani, M.: A comparison of centralized and decentralized energy-management models of multi-microgrid systems. In: *2017 Smart Grid Conference (SGC)*, 2017, S. 1–6

[18] Galloway, B. ; Hancke, G. P.: Introduction to Industrial Control Networks. In: *IEEE Communications Surveys Tutorials* 15 (2013), Nr. 2, S. 860–880. http://dx.doi.org/10.1109/SURV.2012.071812.00124. – DOI 10.1109/SURV.2012.071812.00124

[19] McLaughlin, S. ; Konstantinou, C. ; Wang, X. ; Davi, L. ; Sadeghi, A. ; Maniatakos, M. ; Karri, R.: The Cybersecurity Landscape in Industrial Control Systems. In: *Proceedings of the IEEE* 104 (2016), Nr. 5, S. 1039–1057. http://dx.doi.org/10.1109/JPROC.2015.2512235. – DOI 10.1109/JPROC.2015.2512235

[20] IEEE Standard for SCADA and Automation Systems. In: *IEEE Std C37.1-2007 (Revision of IEEE Std C37.1-1994)* (2008), S. 1–143. http://dx.doi.org/10.1109/IEEESTD.2008.4518930. – DOI 10.1109/IEEESTD.2008.4518930

[21] Antón, S. D. ; Fraunholz, D. ; Lipps, C. ; Pohl, F. ; Zimmermann, M. ; Schotten, H. D.: Two decades of SCADA exploitation: A brief history. In: *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 2017, S. 98–104

[22] Sommestad, T. ; Ericsson, G. N. ; Nordlander, J.: SCADA system cyber security — A comparison of standards. In: *IEEE PES General Meeting*, 2010, S. 1–8

[23] Wang, Wenye ; Lu, Zhuo: Cyber security in the Smart Grid: Survey and challenges. In: *Computer Networks* 57 (2013), Nr. 5, 1344 - 1371. http://dx.doi.org/https://doi.org/10.1016/j.comnet.2012.12.017. – DOI https://doi.org/10.1016/j.comnet.2012.12.017. – ISSN 1389–1286

[24] Zhu, B. ; Joseph, A. ; Sastry, S.: A Taxonomy of Cyber Attacks on SCADA Systems. In: *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 2011, S. 380–388

[25] Nicholson, A. ; Webber, S. ; Dyer, S. ; Patel, T. ; Janicke, H.: SCADA security in the light of Cyber-Warfare.

In: *Computers Security* 31 (2012), Nr. 4, 418-436. http://dx.doi.org/https://doi.org/10.1016/j.cose.2012.02.009. – DOI https://doi.org/10.1016/j.cose.2012.02.009. – ISSN 0167–4048

[26] Kenner, Susanne ; Thaler, Raphael ; Kucera, Markus ; Volbert, Klaus ; Waas, Thomas: Comparison of smart grid architectures for monitoring and analyzing power grid data via Modbus and REST. In: *EURASIP Journal on Embedded Systems* 2017 (2016), aug, Nr. 1. http://dx.doi.org/10.1186/s13639-016-0045-7. – DOI 10.1186/s13639–016–0045–7

[27] Irfan A. Siddavatam, Tanay S. Sachin Parekh P. Sachin Parekh ; Kazi, Frauk: Testing and Validation of Modbus/TCP Protocol for Secure SCADA Commnunication in CPS using Formal Methods. In: *Scalable Computing: Practice and Experience* 18 (2017), Nr. 4. http://dx.doi.org/10.12694/scpe.v18i4.1331. – DOI 10.12694/scpe.v18i4.1331

[28] Sanders, et a. Chris: *Applied Network Security Monitoring : Collection, Detection, and Analysis*. Rockland, MA : Elsevier Science Technology Books, 2014

[29] Ko, C. ; Ruschitzka, M. ; Levitt, K.: Execution monitoring of security-critical programs in distributed systems: a specification-based approach. In: *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097)*, 1997, S. 175–187

[30] Database, RISI Online I.: *Slammer Impact on Ohio Nuclear Plant*. https://www.risidata.com/Database/Detail/slammer-impact-on-ohio-nuclear-plant. Version: 2003 (Last accessed on March 16, 2021)

[31] Zeller, M.: Myth or reality — Does the Aurora vulnerability pose a risk to my generator? In: *2011 64th Annual Conference for Protective Relay Engineers*, 2011, S. 130–136

[32] Kushner, D.: *The Real Story of Stuxnet*. https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet. Version: 2013 (Last accessed on March 16, 2021)

[33] Database, RISI Online I.: *Malware Targets Uranium Enrichment Facility*. https://www.risidata.com/Database/Detail/malware_targets_uranium_enrichment_facility. Version: 2010 (Last accessed on March 16, 2021)

[34] Team, Symantec Security Response Attack I.: *Dragonfly: Western energy sector targeted by sophisticated attack group*. https://symantec-enterprise-blogs.security.com/blogs/

threat-intelligence/dragonfly-energy-sector-cyber-attacks. Version: 2017 (Last accessed on March 16, 2021)

[35] Database, RISI Online I.: *Russian-Based Dragonfly Group Attacks Energy Industry.* https://www.risidata.com/Database/Detail/russian-based-dragonfly-group-attacks-energy-industry. Version: 2014 (Last accessed on March 16, 2021)

[36] Hines, P. ; Blumsack, S. ; Sanchez, E. C. ; Barrows, C.: The Topological and Electrical Structure of Power Grids. In: *2010 43rd Hawaii International Conference on System Sciences*, 2010, S. 1–10

[37] Zhou, C. ; Huang, S. ; Xiong, N. ; Yang, S. ; Li, H. ; Qin, Y. ; Li, X.: Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45 (2015), Nr. 10, S. 1345–1360. http://dx.doi.org/10.1109/TSMC.2015.2415763. – DOI 10.1109/TSMC.2015.2415763

[38] Ullah, I. ; Mahmoud, Q. H.: A hybrid model for anomaly-based intrusion detection in SCADA networks. In: *2017 IEEE International Conference on Big Data (Big Data)*, 2017, S. 2160–2167

[39] Premaratne, U. K. ; Samarabandu, J. ; Sidhu, T. S. ; Beresh, R. ; Tan, J.: An Intrusion Detection System for IEC61850 Automated Substations. In: *IEEE Transactions on Power Delivery* 25 (2010), Nr. 4, S. 2376–2383. http://dx.doi.org/10.1109/TPWRD.2010.2050076. – DOI 10.1109/TPWRD.2010.2050076

[40] Lin, Hui ; Slagell, Adam ; Di Martino, Catello ; Kalbarczyk, Zbigniew ; Iyer, Ravishankar K.: Adapting Bro into SCADA: Building a Specification-Based Intrusion Detection System for the DNP3 Protocol. In: *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*. New York, NY, USA : Association for Computing Machinery, 2013 (CSIIRW '13). – ISBN 9781450316873

[41] Ndonda, Gorby ; Sadre, Ramin: A Two-level Intrusion Detection System for Industrial Control System Networks using P4, 2018, S. 31–40

[42] Koutsandria, G. ; Muthukumar, V. ; Parvania, M. ; Peisert, S. ; McParland, C. ; Scaglione, A.: A hybrid network IDS for protective digital relays in the power transmission grid. In: *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, S. 908–913

[43] Pan, Shengyi ; Morris, Thomas ; Adhikari, Uttam: A specification-based intrusion detection framework for cyber-physical environment in

electric power system. In: *International Journal of Network Security* 17 (2015), 01, S. 174–188

[44] Rrushi, Julian L. ; Campbell, Roy H.: *Intrusion Detection in Electrical Substations 1 Detecting Attacks in Power Plant Interfacing Substations through Probabilistic Validation of AttackEffect Bindings*

[45] Valdes, A. ; Cheung, S.: Communication pattern anomaly detection in process control systems. In: *2009 IEEE Conference on Technologies for Homeland Security*, 2009, S. 22–29

[46] Wetzel, L. Jonathan: *A SEQUENCE-AWARE INTRUSION DETECTION SYSTEM FOR ETHERNET/IP INDUSTRIAL CONTROL NETWORKS*. `https://calhoun.nps.edu/handle/10945/66048`. Version: 2020

[47] Caselli, Marco ; Zambon, Emmanuele ; Kargl, Frank: Sequence-Aware Intrusion Detection in Industrial Control Systems. In: *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*. New York, NY, USA : Association for Computing Machinery, 2015 (CPSS '15). – ISBN 9781450334488, 13–24

[48] Awad, Abdalkarim ; Bazan, Peter ; German, Reinhard: SGsim: Co-simulation Framework for ICT-Enabled Power Distribution Grids. In: Remke, Anne (Hrsg.) ; Haverkort, Boudewijn R. (Hrsg.): *Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems*. Cham : Springer International Publishing, 2016. – ISBN 978–3–319–31559–1, S. 5–8

[49] Davis, C. M. ; Tate, J. E. ; Okhravi, H. ; Grier, C. ; Overbye, T. J. ; Nicol, D.: SCADA Cyber Security Testbed Development. In: *2006 38th North American Power Symposium*, 2006, S. 483–488

[50] Sadi, M. A. H. ; Ali, M. H. ; Dasgupta, D. ; Abercrombie, R. K. ; Kher, S.: Co-Simulation Platform for Characterizing Cyber Attacks in Cyber Physical Systems. In: *2015 IEEE Symposium Series on Computational Intelligence*, 2015, S. 1244–1251

[51] Schloegl, F. ; Rohjans, S. ; Lehnhoff, S. ; Velasquez, J. ; Steinbrink, C. ; Palensky, P.: Towards a classification scheme for co-simulation approaches in energy systems. In: *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, 2015, S. 516–521

[52] OFFIS e.V.: *Mosaik: A flexible Smart Grid co-simulation framework.* `https://gitlab.com/mosaik`. Version: 2020 (Last accessed on March 16, 2021)

[53] Chromik, Justyna Joanna: *Framework for locally implemented model-based traffic monitoring implementing Mosaik co-simulation.* `https://github.com/jjchromik/mosaik-cosim`. Version: 2018 (Last accessed on March 16, 2021)

[54] OFFIS e.V.: *Mosaik: Read the Docs - Overview.* `https://mosaik.readthedocs.io/en/latest/overview.html`. Version: 2020 (Last accessed on March 16, 2021)

[55] Scherfke, Stefan: *Mosaik: A flexible Smart Grid co-simulation framework.* `https://gitlab.com/mosaik/mosaik`. Version: 2019 (Last accessed on March 16, 2021)

[56] Scherfke, Stefan: *Mosaik-CSV: Pseudo simulator presenting CSV data as simulation models for mosaik.* `https://gitlab.com/mosaik/mosaik-csv`. Version: 2020 (Last accessed on March 16, 2021)

[57] Scherfke, Stefan: *Mosaik-HDF5: Store mosaik simulation data in an HDF5 database.* `https://gitlab.com/mosaik/mosaik-hdf5`. Version: 2019 (Last accessed on March 16, 2021)

[58] Scherfke, Stefan ; Lünsdorf, Ontje: *Mosaik-HouseholdSim: Pseudo simulator for residual loads based on load profiles.* `https://gitlab.com/mosaik/mosaik-householdsim`. Version: 2019 (Last accessed on March 16, 2021)

[59] Scherfke, Stefan ; El-Ama, André: *Mosaik-PyPower: Mosaik API implementation for PYPOWER .* `https://gitlab.com/mosaik/mosaik-pypower`. Version: 2020 (Last accessed on March 16, 2021)

[60] Scherfke, Stefan ; Jeddeloh, Gunnar: *Mosaik-Web Visualization: Web visualization for mosaik simulations.* `https://gitlab.com/mosaik/mosaik-web`. Version: 2019 (Last accessed on March 16, 2021)

[61] Lincoln, Richard: *PyPower: A power flow and Optimal Power Flow solver.* `https://github.com/rwl/PYPOWER`. Version: 2020 (Last accessed on March 16, 2021)

[62] OFFIS e.V.: *mosaik: Installation and Downloads.* `http://mosaik.offis.de/install/`. Version: 2021 (Last accessed on March 16, 2021)

[63] Foundation., Python S.: *Python 3.7.10 documentation.* `https://docs.python.org/3.7/`. Version: 2021 (Last accessed on March 16, 2021)

Picture Credit

Figure 2.1 has been designed using icons by Freepik and Becris on `https://www.flaticon.com/`.
Figure 2.2 has been designed using icons by Freepik, prettycons and smashicons on `https://www.flaticon.com/`.
Figure 3.1 has been designed using icons by Freepik and smashicons on `https://www.flaticon.com/`.
Figure 6.4 has been designed using icons by Freepik on `https://www.flaticon.com/`.

Declaration of Academic Integrity

I hereby confirm that this thesis on *„A hieracichal approach to monitoring SCADA networks“* is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited.

Verena Menzel, Münster, March 16, 2021

I agree to have my thesis checked in order to rule out potential similarities with other works and to have my thesis stored in a database for this purpose.

Verena Menzel, Münster, March 16, 2021

Eidesstattliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit über *„Eine hierachische Herangehensweise zur Überwachung von SCADA Netzwerken“* selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

Verena Menzel, Münster, 16. März 2021

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

Verena Menzel, Münster, 16. März 2021