University of South-Eastern Norway

www.usn.no

FM4017 Project 2025

SecurEL Sub-Pilot Module: Automated Protection Relay Settings for Dynamic Grid Operation

Faculty of Technology, Natural sciences and Maritime Sciences

Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FM4017 Project, 2025

Title: SecurEL Sub-Pilot Module: Automated Protection Relay Settings for Dynamic Grid Operation

This report forms part of the basis for assessing the students' performance in the course.

Project group:	MP-08-25
Group participants:	Knut Erland Strætkvern
Supervisor:	Hans-Petter Halvorsen, Saba Mylvaganam, USN.
	Marius Johansen, Espen Andresen, Elvia AS.
Project partner:	Elvia AS

University of South-Eastern Norway

www.usn.no

Summary:

The project aims to explore the concept of dynamic relay settings to contribute to dynamic power grid operation. Today relay parameters such as the UK (tripping) value can be considered as static values. This approach limits the ability to utilize the grid capacity in different operational scenarios. The main objective of the project is to demonstrate how the UK values from the digital protection relays can be retrieved into SCADA, simulated and adjusted remotely from SCADA back to the relays by the operators. The project involved developing three different prototypes and a storage concept. The prototypes are; data-flow simulator, SCADA and web-based simulation dashboard, all interconnected through a shared data storage concept.

- **Data-flow simulator prototype:** Visualizes the data-flow between the main components and systems in the project.
- **SCADA prototype:** Demonstrates retrieval and adjustment of the UK parameter and basic SCADA functionality.
- **Web-based simulation dashboard prototype**: Demonstrates simulation principles for relay parameters by using operational data and formulas.
- **Shared data storage concept:** Demonstrate storage principles that enable interaction and data exchange across multiple prototypes.

The prototypes together demonstrate the feasibility of dynamic relay settings. The developed applications can be regarded as lightweight prototypes but illustrate the concept of how OT and IT environments can connect, and how operational data can be used for simulation purposes. The project has some limitations that are described in the report, these are due to lack of integration with real OT equipment and GIS/NIS model, uses generated measurement values from SCADA and simplified simulations. The work provided valuable insight for Elvia and can be used as inspiration for future implementation and development to achieve dynamic grid operation.

Preface

This project and report mark the completion of my final course in the part-time master's program in IT and Automation at the University of South-Eastern Norway (USN) before starting my master's thesis. The project has been carried out by me as a student, in collaboration with USN supervisors and Elvia AS. The project is inspired by Elvia's sub-pilot in the national research project SecurEL.

The work has been challenging, especially due to limited time available for both report writing and the development of the different prototypes, as it has been carried out as a part-time study in combination with full-time employment. However, the project has been inspiring and provided many valuable lessons that I will take with me into future work. These are lessons I also hope will be beneficial for Elvia in their future work with their sub-pilot in SecurEL.

I would like to express my gratitude to Elvia for facilitating the opportunity to carry out this project in collaboration with them. I would also like to thank my supervisors Hans-Petter Halvorsen and Saba Mylvaganam at USN for their support, discussions and constrictive feedback throughout the process.

Hamar, 16.11.2025

Knut Erland Strætkvern

Contents

Nomenclature

Abbreviation	Explanation	Unit	Meaning
ADMS	Advanced Distribution Management System		System for managing electric distribution networks with functions like customer notifying and workorder handling
CT	Current Transformer		Transformers that measure current
GOOSE	Generic Object-Oriented Substation Event		IEC 61850 protocol for event-based communication
GUI	Graphical User Interface		Visual interface for user interaction with system software
MMS	Manufacturing Message Specification		Protocol in IEC 61850 for real-time communication
MU	Merging Unit		Device used in IEC 61850 that collects and transforms analog to digital signals
POC	Proof-of-concept		Experiment, prototype, pilot project to demonstrate feasibility of idea or concept
RTU	Remote Terminal Unit		Device that connects OT devices, signals, measurements with SCADA and IT solutions
SCADA	Supervisory Control and Data Acquisition		System for monitoring, control and alarm handling of industrial processes
SMV	Sampled Measured Values		Protocol in IEC 61850 for transmitting digitized current/voltage measurements
TPV	Tilknytning på vilkår / Connection subject to conditions		Grid connection granted on specific terms
UK	Utkoblingsverdi / Tripping value	[A]	Current threshold setting that defines when circuit breaker trips
VT	Voltage Transformer		Transformers that measure voltage
XAML	Extensible Application Markup Language		.NET language for defining user interfaces

1 Introduction

This chapter presents the overall context and premise of the project. It outlines Elvia's strategic objectives and how it is related to the project. The chapter then presents the objectives and the scope of the project. Finally, it provides an overview of the structure of the report by describing the main content of the chapters to the reader.

1.1 Elvia's strategic objectives

Elvia has a strategic goal of increasing their existing grid capacity by 20% in the coming years. In 2024 Elvia was able to increase the capacity by 7% corresponding to 480MW of extra capacity. Approximately 50% of the 480MW was achieved through conditional connection agreements (TPV) with customers. These agreements allow Elvia to disconnect a customer from the grid if necessary for operational reasons. Although under normal circumstances the customer remains connected and can consume or produce energy as usual. The remaining 50% of the capacity was achieved through updated risk assessments and alternative grid analyze methods [1].

Another possible contribution to the 20% target is to implement a solution to dynamically change tripping values (UK), for digital protection relays in the grid. Typically, in conventional practice, the UK values are set for the worst-case scenario at a reference temperature (typically 25 degrees Celsius) and are treated as static values that are rarely modified. By implementing a dynamic solution based on simulations, it becomes possible to increase the UK thresholds in scenarios where this is operationally acceptable. A hypothetical case where dynamic UK values are relevant is during the winter, when the ambient temperature around the power lines is significantly lower than the reference temperature. This would enable higher current flows through the power lines than originally intended and thereby contribute to higher utilization of the existing grid infrastructure. Other weather factors that can influence the current carrying capacity of power lines are wind speed and direction, solar radiation, humidity etc.

1.2 Project objectives

The goal of the project is to develop a data-flow simulator prototype that visualizes the data-flow for the dynamic UK settings solution for Elvia. The visualization is intended to illustrate how the data flows between different main components in the system. The goal of the visualization is not to go into detail in all the affected components, but to show the bigger picture. The simulator can be seen as a digitalization of the system sketch shown in figure 1.

In addition, the project aims to develop a simplified SCADA prototype that shows how the collected UK values can be presented to the operators in the SCADA GUI, as well as how the operators may write updated UK values back to the digital protection relays.

In addition to the SCADA prototype the project will aim to develop a web-based simulation dashboard prototype where the operator can perform grid simulations using the collected UK values as well as other necessary data from other systems. The simulation dashboard can be seen as a working tool for the operators to find the recommended UK values, before writing the updated UK values back to the digital protection relays using the SCADA system.

To ensure data consistency and interaction between the different prototypes, a shared data storage concept will be implemented. The concept can be considered as a simplified solution, but with the goal of demonstrating a storage concept between multiple prototypes. The storage concept ensures that all the prototypes can access and update the same underlying data source.

Figure 1 shows a proof-of-concept system sketch created together with Elvia of the dynamic UK settings concept. The different colored frames show different levels in the system. The orange frame represents the digital protection relays in the substation at bay level. The yellow frame represents the RTU in the substation at the station level. The green frame represents the control center level where the IT environment and operators are located. The control center is normally not located in the substation, but in a centralized location, for example at the company's headquarters. The arrows in between the different boxes/components indicate the data-flow in the system. The figure shows two different substations, substation A and B. And these substations have different RTU solutions, physical and virtual, and different data-flow paths for sending and retrieving UK values. The path with the physical RTU located at the station level is the standard path for data communication between the substation and the

SCADA system today, but it is desired in future work to test the implementation of a virtual RTU solution in a real substation as well. This project can be seen as feasibility study ahead of the implementation of this solution.

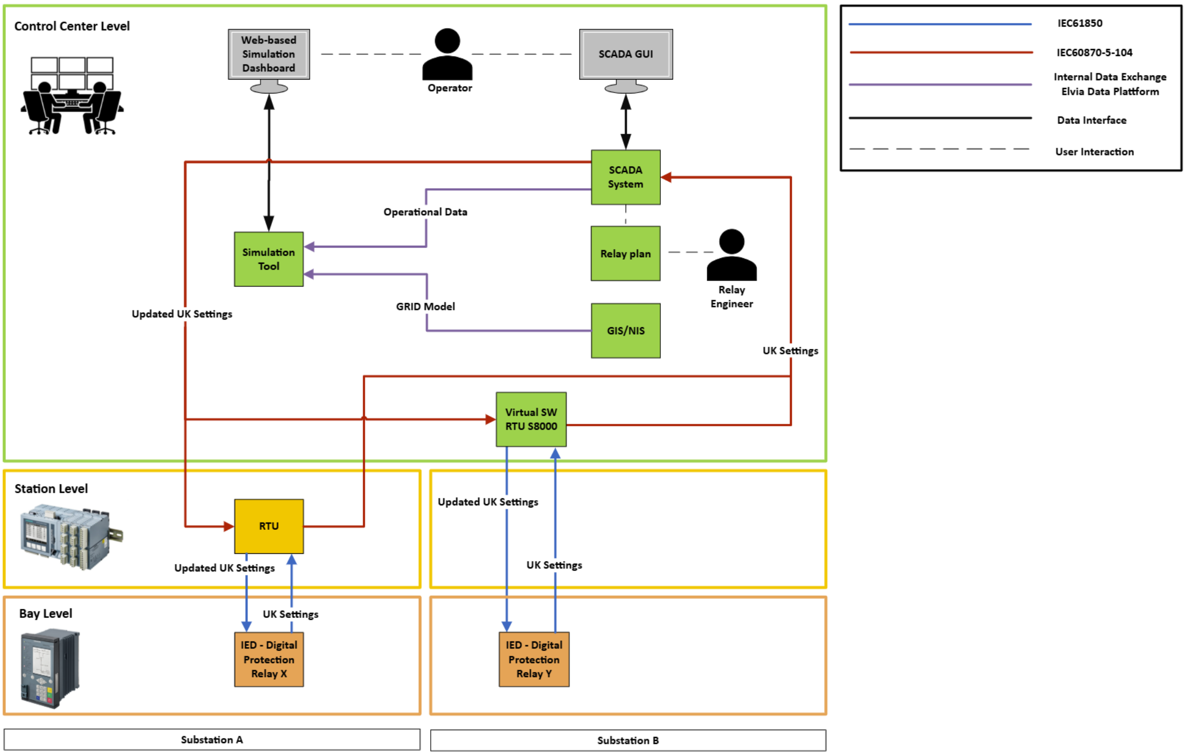


Figure 1 – Prototype architecture diagram that shows the overall concept of dynamic relay settings for Elvia. The diagram shows how the main components in the scope communicate with each other across different levels using different RTU solutions for different substations.

1.3 Project scope

The scope of this project is limited to demonstrating the main principles of the SCADA prototype and the web-based simulation dashboard prototype for the operators, as well as illustrating the overall data-flow in the system in the data-flow simulator prototype, all connected through a shared data storage concept ensuring common access to the same data. Developing a fully functional system would require significantly more detailed engineering of all elements. The aim of this project is therefore not to provide a complete technical solution ready for implementation but rather present a conceptual overview of how the solution could work in principle and be an inspiration for further development for Elvia.

1.4 Structure of the report

The technical report follows the IMRAD structure and is organized into these main chapters.

- **Chapter 1 – Introduction:** Introduces Elvia's strategic objectives, the overall objective of the project, the scope and the structure of the report.
- **Chapter 2 – Background:** Describes relevant theory and technical background, including power grid history, basic power grid theory, and an overview of protection and control systems with their components, standards and protocols.
- **Chapter 3 – Specification:** States the system requirements of the project, including functional and non-functional requirements, system boundaries, assumptions and data requirements.
- **Chapter 4 – Methods, tools and materials:** Describes the research approach, tools and implementation methods applied through the project.
- **Chapter 5 – Data-flow simulator prototype**: Presents the design, implementation and the results of the data-flow simulator prototype.
- **Chapter 6 – SCADA prototype**: Presents the design, implementation and the results of the SCADA prototype.
- **Chapter 7 – Web-based simulation dashboard prototype**: Presents the design, implementation and the results of the web-based simulation dashboard prototype.
- **Chapter 8 – Shared data storage concept:** Presents the design, implementation and the results of the shared data storage concept.
- **Chapter 9 – Discussion:** Discusses the main findings of the project, strengths and weaknesses of the prototypes, lessons learned and implications for Elvia.
- **Chapter 10 – Conclusion:** Summarizes the key findings, objectives and overall conclusions in the project.

2 Background

This chapter presents relevant technical and historical context for the project. It starts with some relevant power grid history followed by some fundamental power grid principles. The chapter then introduces key components, standards and protocols of modern protection and control systems. Together these sections establish the background necessary to understand the concept of the project.

2.1 Power grid history

In recent decades, there has been a significant increase in the number of production facilities connected to the power grid. For more than a hundred years, hydropower plants have constituted the dominant energy source in Norway. Hydropower plants are characterized by a high degree of predictability and quite straightforward controllability. Electricity production from hydropower facilities can therefore be adjusted to meet the need of the grid without major complications.

Since the early 2000s, wind power has played an increasingly important role in the power system, with wind farm being connected to the power grid. And over the past decade and especially during the last two to three years the expansion of solar power has accelerated with large-scale solar plants and smaller household solar installations being connected to the grid [2].

This development has introduced new challenges related to the operation and monitoring of the power system and especially related to protection relay settings. The power grid was originally designed for a centralized production and distribution model where the electricity was normally transported in one way and with few production units that were predictable and controllable.

The rapid increase in decentralized and converter-based production in the medium voltage grid and in the households has introduced bidirectional power flows. Frequent and rapid changes in production have become very common due to the high number of production units that are highly dependent on the wind and sunny weather conditions. These changes in the power grid complicate the protection relay settings individually, but also the overall system. The power grid has become very complex, and it can be difficult to see the big picture in different operating conditions. This leads to more situations with large-scale power outages, which are difficult to predict, manage and analyses in terms of their root causes. An example of this was the large power outage in Spain and Portugal on April 28, 2025, that affected around 50 million people. The blackout caused disruptions to the transport, infrastructure, hospitals a daily life in general. The conclusions drawn from the analysis on the incident remain debated and the exact root cause is unclear to many experts.

Another challenge is the grid capacity. Not only due to production facilities that are being connected, which require solid grid capacity to be able to deliver the production capacity. But there is also a high interest for customers and especially high-power industries to connect to the grid. Statnett has stated that the transmission grid capacity is full in the regions around Oslo and that there will be no available capacity for expansions or new large customers (megawatt) before 2030-2035 [3].

2.2 Basic power grid theory

To understand the objective of the project it is necessary to have some basic knowledge of how the power grid operates, and especially how the protection and control system in substations functions and interacts with the operational control center.

The power grid in Norway consists of three different grid levels: the transmission network (420kV - 300kV AC, some areas also 132kV), the regional network (132kV - 33kV AC) and the distribution network (22kV - 0.230kV AC) [4], where the voltage level between the different networks is transformed using transformers located in substations. The different grid levels can be seen in figure 2, and together the various networks operate as an integrated system that deliver electricity from producers to consumers. Statnett operates the transmission network and the power companies, like Elvia, operate the regional and distributional networks. It is crucial to keep a balance of production and consumption to maintain a stable grid frequency of 50Hz. If there is an imbalance between production and consumption the frequency will drop or increase from 50Hz. This may lead to power outages in the power grid due to automatic disconnection from protection relays, but it can also result in failure in electrical components, so maintaining the frequency at 50Hz is important.

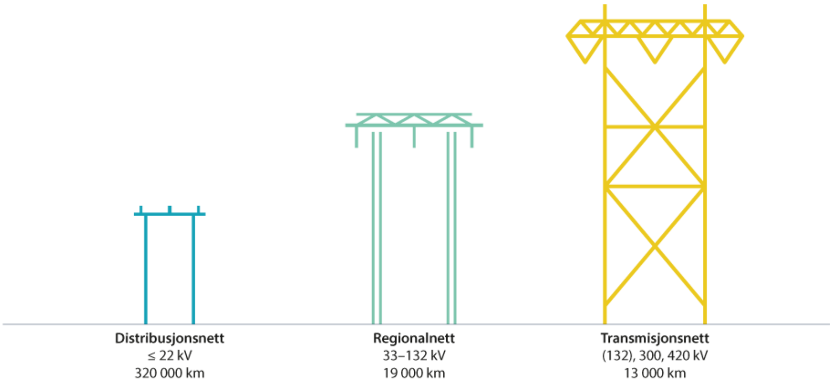


Figure 2 – Overview of the voltage level structure in the Norwegian power grid [5].

Figure 3 shows an example of a substation located above ground level. The substation is called Tonsen substation and is located in Oslo and is built in 1920. The substation is today listed as a cultural heritage site by NVE due to its historical role in the electrification of Norway. Tonsen consists of three transformers transforming voltages from 47kV to 11kV AC [6]. The transformers are located behind each separate grey chamber outside the substation, which can be seen in the picture. Inside the substation one finds high-voltage switchgear and busbars, protection and control systems among other equipment.

Figure 3 – Substation example of Tonsen substation located in Oslo [6].

2.3 Protection and control system

The protection and control system are crucial for operating a modern power substation. Their main function is to contribute to a safe, stable and reliable operation of the power grid using monitoring and protection equipment. The control system operates as a central hub of the substation, connecting the protection, monitoring, controlling and operational functions into a complete solution. The control system provides the operators with necessary tools to monitor and control the substation. The operators can operate the substation locally or remotely using the control system. This includes controlling of high-voltage breakers, real-time monitoring of measurements and more. The control system is kept separate from the high-voltage equipment in the substation. The control system voltages are typically 110-230V AC for input supply, while the protection and control equipment typically operates on 24-110V DC. The conversion from AC to DC supply is performed by using rectifiers in the substation.

Figure 4 shows a picture of a control room in a substation. The relay panels in the picture typically contain digital protection relays, IEDs, control devices, measuring devices, alarm and indication systems, network equipment, communication systems and more. Together all these functions ensure reliable protection, monitoring and control of the substation.

It is normal that each bay in the substation has their own separate relay panel with dedicated IEDs and other relevant equipment. This is why it is normal that relays panels are often very similar in a substation control system, because a lot of the relay panels contain equipment with same functionality. But if there is limited space in the control room it is possible to have multiple bays in the same relay panel. Relay panels for busbars, shared infrastructure, transformers are typically somewhat different than for bays, as they contain some other equipment that is tailored to their specific functions and requirements. Figure 4 shows that the two first relays panels from the left are similar, these panels are most likely for bays, while the next relay panels are different for other functionality in the control system possibly for transformers, busbars etc.

Figure 4 - Relay panels in a substation control rom. The panels contain different components related to the control system in a substation. The panels typically contain IEDs, measurement indicators, buttons, switches, RTU etc. [7].

2.3.1 Digital protection relay

The protection systems are designed to detect unnatural conditions in the power grid. These conditions can occur due to short circuits, overloads, ground faults, current deviations, frequency deviations and more. When an error occurs the digital protection relay will send signals to the circuit breakers to isolate the faulty circuit from the grid. This also triggers alarms or warnings to the RTU that will further notify the message locally through an alarm list in an HMI and remotely to the SCADA system [8].

An important task when configuring the protection relays is to configure them in way that when the error occurs the smallest affected area is isolated. This is called selectivity, meaning that the circuit breaker closest to the fault should trip, so that larger areas are not unnecessarily affected. If the protection relays do not isolate the fault this can potentially lead to equipment damage, fires and large power outages. The configuration of the digital protection relays is typically performed by a relay engineer based on a relay plan. The relay plan is a scheme/document that defines how the protection relays should be configured and coordinated in the power grid. The relay plan is set up by an engineer based on grid analysis. The replay plan includes the UK value that is the main parameter in this project.

Figure 5 shows a picture of a SIPROTEC 5 digital protection relay from Siemens. A digital protection relay is often referred to as an Intelligent Electronic Device (IED). The term IED is a bit more abstract, as it covers a wider range of devices with different functions and opportunities beyond just ensuring grid protection. Therefor the term digital protection relay will primarily be used in the report. But IED can occur in some sketches and figures.

Figure 5 – Example of an IED showing the Siemens SIPROTEC 5 digital protection relay. The device is used for monitoring, protecting and controlling electrical power systems by detecting faults and operating circuit breakers [9].

The SIPROTEC 5 IED front includes light indicators, buttons, key twisters and a screen. The indicators on the left side primary shows internal protection functions of the relay. This means that the indicators will indicate if the relay detects pickup conditions, trip commands and activation of different protection elements in the relay, for example if the relay has detected a ground fault. The indicators will normally light until they are acknowledged by an operator locally.

The right-side indicators are normally dedicated to the switchgear and the overall operational environment. Examples of these are interlocking, remote/local control, communication errors, system related alarms and circuit breaker status. All the indicators are logically programmable and the text next to the indicators lamps is simply printed on paper and can be updated and changed.

The front also has arrow buttons to navigate and select switchgear and input/output buttons to control the switchgear position. The display will update the switchgear position after completed command as confirmation of successful operation.

The rear side of an IED contains different electrical interfaces that connect the unit to the substation equipment. The interfaces are analog and digital input/outputs for current and voltage transformers, status signals, trip commands. The interface also has fiberoptic and RJ45 inputs for network communication. Figure 6 shows the rear side of a SIPROTEC 5 IED.

Figure 6 - Exampled of rear side of IED, showing a SIPROTEC 5 digital protection relay with multiple electronic interfaces [10].

Other manufacturers also offer IEDs, with similar functionality and they often look similar with similar content on the front panel and rear interfaces. Examples of other manufacturers are ABB/Hitachi/Schneider/VAMP.

2.3.2 RTU

One of the main components in the control system is the RTU. It is a physical field device that serves as an interface between local equipment and the SCADA system in the control center. This component is typically located in the control room in the substation together with the relay panels shown in figure 4. The RTU collects measurements, enables remote control, detects faults, transmits alarm and status messages and more. In addition, it works as a protocol converter between the substation and the control center for data communication. Communication between the RTU and the control center commonly uses the IEC 60870-5-104 protocol which is based on TCP/IP. RTUs are designed as highly reliable devices, built to operate in harsh environments with varying temperatures, vibrations and more [11].

Figure 7 shows a SICAM A8000 RTU from Siemens. RTUs come in different shapes with different solutions, but typically they can be expanded with various snap-on or slot modules, such as digital or analog I/O, communication ports or other interfaces. The RTUs can therefore be engineered differently in different facilities to meet the requirements.

Figure 7 - Siemens SICAM A8000 RTU. The unit works as a Remote Terminal Unit that is used for data acquisition and control in substations and distribution networks [12].

2.3.3 Virtual RTU

RTUs have been present on the market for several decades as reliable hardware components. But as hardware devices the RTUs require regular maintenance such as firmware updates, performance monitoring and hardware replacement due to faults. Therefor in recent years vendors have explored the possibility of implementing RTU functionality in software solutions instead. This has led to the development of the SICAM S8000 software product from Siemens. A software RTU essentially provides the same functions as a physical hardware RTU but can run on standard servers or virtual machines instead of dedicated equipment. The software RTU therefore benefits in terms of flexibility, scalability and easier maintenance and monitoring compared to traditional RTUs [13]. But there are security aspects to consider since one moves the unit outside the substation into the IT environment, which is unconventional in today's industry and is therefore still little tested in practice. The system sketch in figure 1 shows how the virtual RTU is located combined with the physical station RTU.

2.3.4 IEC 61850

IEC 61850 is an international standard for communication and automation in substations. It describes and defines how digital protection relays, RTUs, network devices and other automation components exchange information in substations. The IEC 61850 standard is designed to provide interoperability, meaning that the solution will function across different vendors and equipment.

A key feature of the IEC 61850 standard is its object-oriented data model, that standardizes how devices and signals are represented. This feature makes it easier to integrate devices from different vendors, simpler engineering and easier maintenance. The standard is divided into three methods for communication: MMS, GOOSE and SMV [14]. These protocols will not be described in detail in this report but figure 8 shows how they are integrated into the IEC 61850 architecture and how they are used to communicate across different levels and equipment in the architecture.

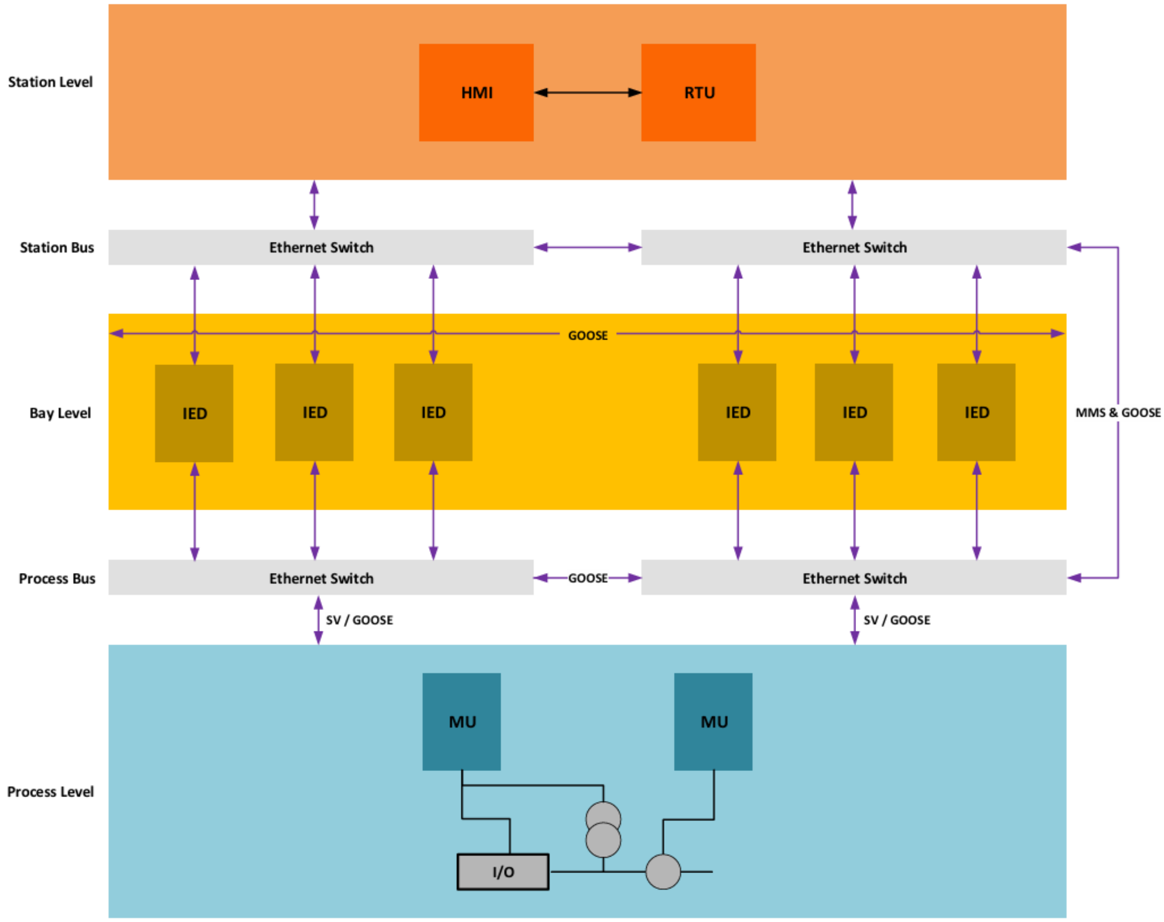


Figure 8 - The figure shows in detail the architecture of station- and process buss network in the IEC 61850 network. The architecture is separated into different levels using different protocols to interact between the different levels and components [14].

In the IEC 61850 architecture the RTU, IEDs, HMI and other station level devices are interconnected via the station bus, which uses fiber-optic or ethernet cables together with network switches. The standard also describes the process bus, which enables digitalization of measurements between the switchyard and the IEDs at process level. While the station bus handles communication at the station level, the process bus operates on the process side of the protection relays. It uses merging units (MU) to collect analog signals from CTs and VTs and convert them into digital signals transmitted to the IEDs using fiber-optic or ethernet cables. Process bus technology is typically applied in digital substations in combination with station bus. Compared to conventional copper-based wiring from CTs and VTs to the IEDs, process bus introduces more complexity and requires higher competence to set up, but certainly offers benefits of flexibility, reduced cabling and advanced functionality [15]. Summarized IEC 61850 is an international standard that enables interoperable communication in substations.

2.3.5 IEC 60870-5-104

IEC 60870-5-104 is a widely used communication protocol in the electrical power industry. It is an extension of the IEC 60870-5-101 protocol, designed to enable full network-based access for monitoring, control and automation of electrical power systems. While IEC 60870-5-101 uses serial data communication. IEC 60870-5-104 operates over TCP/IP, which makes it easier to integrate with modern IT/OT infrastructure. Both protocols are used to connect substations to control centers and allow for data communication [16]. Although IEC 60870-5-101 is an older protocol, which is still used in the industry today, IEC 60870-5-104 is generally preferred due to its compatibility with TCP/IP, higher data throughput and improved scalability. Figure 1 shows where the IEC 60870-5-104 protocol is used in the system architecture of a modern control system.

3 System specification

This chapter presents the specifications of the system. The goal is to describe the requirements, assumptions and constraints for the system. The requirements were collected and categorized using the FURPS+ model, where the model distinguishes between functional and non-functional requirements. To provide a clearer understanding of the system behavior, use-case diagrams and domain models were used to illustrate interactions between actors and different system elements.

3.1 System functional requirements

Functional requirements describe what the system should do from its perspective and its users. The functional requirements were identified through analysis of the technical specification and categorized into use cases that illustrate the interaction between actors and the use cases.

Table 1 shows the functional requirements that have been identified:

Table 1 - Functional requirements collected from the technical specification.

F – Functionality	1. Manage configuration – The user can access the system configuration to read and make changes in the data-flow simulator, SCADA and simulation dashboard prototypes.
	2. The data-flow simulator must graphically visualize the data-flow in the system between the main components.
	3. The UK values in the digital protection relays must be read upstream and updated downstream from SCADA.
	4. The SCADA prototype should demonstrate how the solution will appear for the operator with some basic functions like measurement values, status indications and switches and the UK values from the data-flow simulator.
	5. The web-based dashboard prototype should demonstrate how the solution is for the operator with some basic simulation functions and recommended relay settings as outcome with high focus on usability.
	6. Shared data storage should be accessible to all prototypes.

3.2 System non-functional requirements

Non-functional requirements describe the constraints, performance, usability and quality of the system. The non-functional requirements describe how the system should perform. The non-functional requirements were also collected using the FURPS+ model, where the model covers categories like usability, reliability, performance, supportability and other constraints.

Table 2 shows the non-functional requirements that have been identified:

Table 2 - Non-functional requirements collected from the technical specification.

U – Usability	1. The SCADA and simulation dashboard GUI should be intuitive and user friendly.
	2. The SCADA prototype should be similar to Elvia's existing SCADA GUI for easier implementation in existing systems.
	3. The simulation dashboard should be accessible via standard web-browser without extra software or addons.
	4. The data-flow simulator must be easy to understand with an intuitive GUI that can be used for presentations to stakeholders.
R – Reliability	1. Data must in general be updated in real time. But delays can be added to make it easier to see the data-flow in the data-flow simulator. Update buttons can be implemented if this avoids noise in the usability in the prototypes.
	2. The three prototypes must handle error scenarios without crashing when missing data, decimal or character input where integers are only allowed etc.
	3. The system should notify the operator if it is missing or corrupted data.
P – Performance	1. Simulations should be performed within reasonable time < 1 min.
	2. GUIs must run smoothly without significant lags.
S – Supportability	1. The prototypes must be easy to expand since the project focuses on a simplified solution.
	2. Documentation should follow the IMRAD structure and config file should make it easy to add more functions or scenarios.
+ Design challenges/ Limitations	1. The solution is meant to be a simplified prototype for demonstration and is not a fully functioning solution.
	2. The data used in the solution will be simulated and not actual measurement data.

	3. The GUIs will primarily demonstrate principles and not a fully technical solution.
	4. Data security and network solutions will not be studied in the project.
	5. GIS/NIS data will not be included in the scope.

3.3 System boundaries and assumptions

This chapter describes the scope of the system by describing its boundaries and assumptions. The boundaries describe what is included in the system solution and what lies outside. While assumptions tell what conditions that are relevant for design and development. These aspects are used to ensure that the requirements are realistic and that the expectations are met in the project.

3.3.1 System boundaries

The following topics are inside the system boundaries:

- **A data-flow simulator**, visualizing the flow of the UK values between the digital protection relays, RTUs, SCADA, GIS/NIS and simulation tool and dashboard.
- **A SCADA prototype,** that demonstrates how operators can view and adjust UK values in the GUI based on Elvia's SCADA GUI style.
- **A web-based simulation dashboard,** that enables operators to perform simplified simulations based on different scenarios and retrieved data.
- **Data storage concept,** that ensures that all prototypes can access and exchange data.

Figure 9 shows an overview of the system architecture at a high level. It shows how the prototypes are connected through the data storage concept (CSV), and the arrows in the figure indicate the data-flow. One can see that the SCADA prototype and the data-flow simulator prototype both have read/write access to update the storage system, while the web-based simulation dashboard prototype have read access only. One can also see that only the data-flow simulator prototype and the SCADA prototype retrieve data from the configuration data file. The web-based simulation dashboard prototype does not use the configuration file.

The text inside the monitors in figure 9 indicates the framework used to develop the different elements. The elements these are described later in the report.

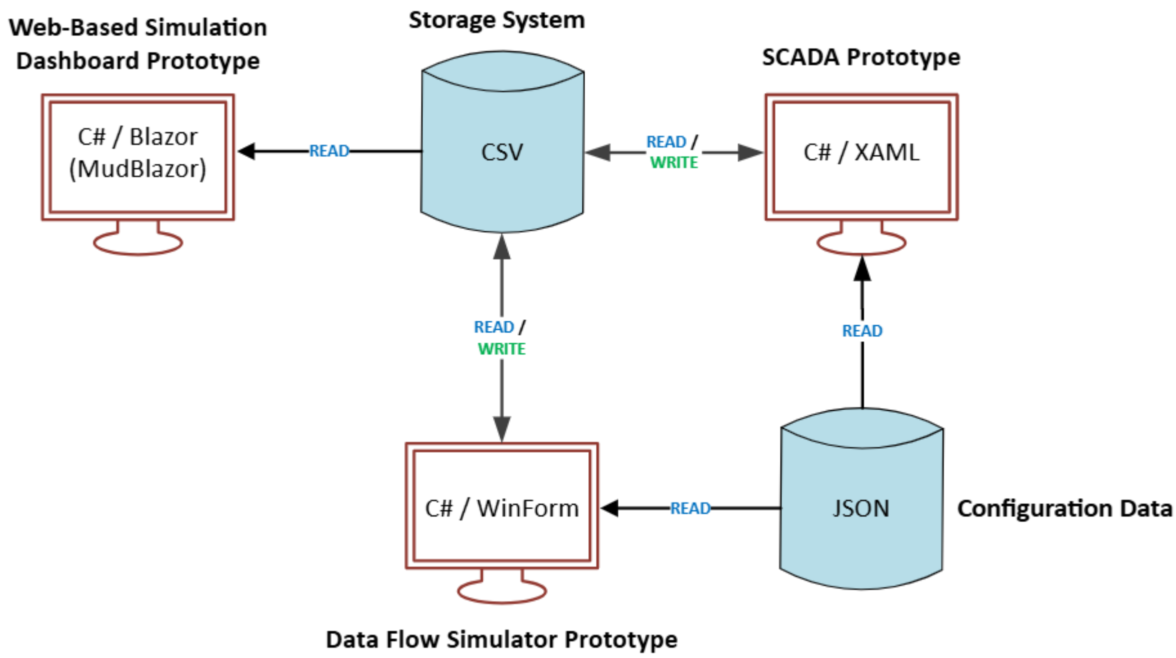


Figure 9 - High level system architecture for automated relay settings system. The diagram illustrates how the prototypes are connected through the storage system, and how some of the prototypes interact with the configuration data.

The following topics are <u>outside</u> the system boundaries:

- Connection to real physical hardware.
- Integration with Elvia's production systems.
- The grid-model from the GIS/NIS system.
- A full-scale simulation tool that delivers realistic simulation outcomes.
- Delivery of a fully functioning system ready for operational implementation.

3.3.2 Assumptions

The design and development of the prototypes are based on the following assumptions:

- All data in the prototypes are simulated data and not based on real measurements.
- The development of prototypes assumes that the operators have basic knowledge of IT and electrical power engineering.

- The prototypes are created as proof-of-concept and intend to serve as inspiration for further development by Elvia.
- Due to time and resource constraints, the system is delivered with simplified functionality where the goal is to demonstrate ideas and concepts.

3.4 Data requirements

The data requirements describe the data information that the system must handle. This includes the types of data, the structure of the data, storage of data and preprocessing.

3.4.1 Types of data

- **UK values:** User defined integer numbers defined in the config file, data-flow simulator or in the SCADA prototype.
- **Measurement values:** Simulated sensor measurements like voltage, current and power.
- **Indications**: Switch positions and digital signals simulated from the substation in SCADA GUI.
- **GIS/NIS data:** The power-grid model from the actual GIS/NIS Elvia system will not be used in the project.
- **Simulation result:** Recommended UK value outcome from the simulation dashboard.
- **Configuration data**: User inputs for config and fallback values for the different prototypes.

3.4.2 Data structure

- All exchanged data will be structured in a CSV format that enables exchange between prototypes.
- Configuration data will be stored in a JSON file that can be easily opened and edited using standard text editor.

3.4.3 Data storage

- A CSV file will be used for data storage sharing to ensure that alle of the prototypes can access the same updated data.
- The data-flow simulator and SCADA prototype shall have read/write access to the file, while the simulation dashboard shall only have read access.
- The content of the CSV file will be stored and used the next time the prototypes run.
- If the values do not exist in the CSV file, the prototypes will use fallback values in the configuration file.
- The CSV file will be open without authentication since the data is not sensitive or of importance in this project.

3.4.4 Preprocessing requirements

- The simulated data will be generated in real-time based on the actual parameters in the system.
- Outcomes in calculations in the prototypes shall be rounded and displayed with a maximum of one decimal or none.
- Input values in the configuration file or in the prototypes should only consist of a maximum of one decimal or as an integer. The prototypes will have functionality to avoid bad user inputs, while the configuration file does not, so it requires correct user input at this stage.

4 Methods, tools and materials

This chapter describes the methods, tools and materials used in the project. The work in the project was performed as a proof-of-concept study where the goal was to demonstrate the principles of dynamic protection relay settings, rather than developing a fully developed system. The work resulted in development of three different prototypes, that was applied with simulated input data and simplified calculations and simulation approaches. This approach was applied to reduce the complexity of the project and focus on the concept instead.

It is important to note that the study of this project does not include cyber security or network infrastructure solutions. A complete solution would require dedicated resources to help with these topics.

The following subchapters present the research approach, tools and techniques applied, data sources, testing and limitations of the chosen methods.

4.1 Research approach

The research approach was performed as proof-of-concept study to demonstrate the feasibility of dynamic protection relay settings. The goal in the project was to test and validate key principles through prototyping rather than a fully functioning system.

The development process was performed using an iterative approach for implementation and testing of functionality. A system sketch was developed, and requirements were identified to avoid misunderstandings and development of wrong solutions.

The approach excluded implementation of field testing, cyber security and network solutions. These aspects would be required in a complete solution but were considered outside the scope for this proof-of-concept project.

4.2 Tools and technologies

The research approach was supported by specific tools and technologies to enable prototyping and demonstration of the proof-of-concept. The main language used for development was C#, as it works well with the .NET framework, has good support for object-oriented programming and suitable for making prototype GUIs using the windows presentation foundation [17]. Different frameworks were used for frontend development of the different prototypes in the project. WinForms was used to develop the data-flow simulator prototype, as it offers simple and rapid interface development. XAML was used to design the GUI for the SCADA prototype, as it provides a declarative and structured approach to interface design [18].

The backend development was carried out in Visual Studio. It is a widely used integrated development environment that supports both C# and XAML, and it was chosen because it offers good debugging tools good integration with the .NET framework and is an efficient work platform for building and testing prototypes.

The development of the web-based simulation dashboard was carried out using Blazer WebAssembly in combination with MudBlazor. This allowed for development of a modern web-based interface. The combination allowed for the creation of an interactive and responsive dashboard using C# and the .NET framework running in the browser.

In addition to the programming tools, the project used a simple file-based storage format to manage configuration and shared data. The configuration data was stored in a JSON file. JSON files can be easily opened and edited using standard text editors. It was chosen because it is widely supported in multiple programming environments, which makes it suitable for prototyping and configuration management.

Furthermore, a shared data storage concept based on common CSV file was implemented. The CSV files ensure that all the prototypes can access and update operational data. CSV was chosen because it is a simple way to structure data and allow integration across multiple applications. It can be considered as a lightweight storage solution that is efficient for testing and works well in a proof-of-concept phase. In a fully developed solution other storage concepts such as SQL-based databases or cloud-based concepts in Azure should be considered instead of CSV.

To test the prototypes, simulated input data was used to demonstrate the functionality. This approach was used to replace the need for real hardware during the proof-of-concept phase. The approach also led to controlled experiments without any risks.

4.2.1 Version control with GitKraken

GitKraken was used during the development process of prototypes to handle version control. GitKraken offers a visual interface that makes it easy to commit and track changes throughout the development process. By using GitKraken it was easy to rollback to previous versions due to code bugs or if implemented changes did not work as planned. Overall GitKraken makes it easy to maintain control and stability throughout the development process [19].

4.3 Use of artificial intelligence

This chapter describes how artificial intelligence (AI) was used in the project. The chapter is divided into two sections, where the first one describes how it was used when writing the report and the second one describes how it was used during code development.

4.3.1 Use of artificial intelligence in writing the report

Chat GPT was used as a sparring partner during the writing process of the report [20]. The purpose of using AI when writing the report was to:

- Improve the structure and organization of the report
- Assist with translations from Norwegian to English
- Improve the overall flow and clarity of the text

No parts of the report were directly copied from AI generated text. However, some AI suggestions have been used as inspiration and partially merged into the text where it helped to improve the report. All the written content in the report has been reviewed and approved by the editor, so it reflects the authors own understanding of the topics.

4.3.2 Use of artificial intelligence in code development

Chat GPT was also used as a technical sparring partner during the development of the prototypes. The purpose of using AI during the development process was to:

- Improve the code structure and readability
- Identify weaknesses and suggest improvements
- Suggest alternative solutions to solve programming errors
- Offer development tips based on technical specifications
- Increase development efficiency

Some AI generated code snippets were partially integrated into the prototypes to solve problems and bugs, particularly where the code complexity was high. But all contributions were reviewed, tested and approved by the developer.

The use of AI during the development of the prototypes made it possible to achieve better and more complete prototype solutions than would have been achieved without such support. Without the use of AI during the development of the project the prototypes would have been less complete due to available resources in the project. It was considered beneficial to use AI to achieve more within the project timeframe and to better demonstrate the overall concept to the customer, Elvia. The use made it possible to illustrate a completer and more realistic version of the system.

4.4 Data sources

Since there was no direct link to physical equipment, the prototypes were developed and tested using simulated data. The configuration file can be used to set some user-defined parameters for some of the prototypes. The use of simulated data made it possible to test and demonstrate the prototypes under controlled conditions, but still reflecting realistic values.

4.5 Validation and testing methods

The prototypes were tested with simulated data to verify that the system could process and visualize the values correctly. The validation of the prototypes focused on functionality, usability and responsiveness of the GUIs, and not data quality and actual collected measurements.

4.6 Limitations of methods

The prototypes were tested using simulated data which is a limitation of the project. While it allowed controlled and flexible testing it certainly did not introduce the same complexity, noise or processing issues that is typically when collecting real measurements form industrial environments. In further work this approach will be necessary to acquire and test to ensure that a fully functioning system works in practice with real equipment.

Another limitation is that network and cyber security aspects were not included in this project and in the prototype solutions. These are key aspects in IT/OT systems, so this must be introduced in further work with the project. The solution focuses on proof-of-concept of functionality and to visualize rather than network and cyber security challenges.

Grid simulations performed in the simulation dashboard were carried out with simplifications. Performing grid simulations introduces high complexity and would require access to detailed and large volumes of grid data. This is beyond the scope of this project.

The prototypes can be regarded as lightweight solutions rather than fully functioning solutions. They do not serve advanced functions with multiple integrations, security mechanisms, redundancy, backup solutions and more, as the focus in the project was to demonstrate a proof-of-concept for dynamic relays settings.

Another limitation is related to the use of JSON and CSV files. These were selected because they are lightweight, easy to modify and suitable for rapid prototyping. But they have weaknesses regarding multi-user access, concurrency control and transaction handling. In a fully developed industrial solution a database or cloud-based storage system would be preferred.

In summary, the prototypes, configuration file and storage concept cannot be directly implemented in industrial facilities without significant further development, but they demonstrate the feasibility of the overall concept.

5 Data-flow simulator prototype

This chapter presents the results from the development of the data-flow simulator prototype. The simulator aims to visualize the data-flow between the digital protection relays, RTUs, SCADA, GIS/NIS and simulation tool and dashboard with some basic user functionality.

The chapter is divided into three subchapters. Subchapter 5.1 describes the design phase of the development. Subchapter 5.2 describes the implementation phase and subchapter 5.3 describes the results.

The development process using WinForms was greatly inspired and supported by various YouTube channels providing tutorials and practical demonstrations such as *Industrial IT and Automation* [21] and *Programentor* [22] and web-pages like *Microsoft documentation* [23], in combination with Chat GPT as sparring partner [20].

5.1 Design of data-flow simulator prototype

This subchapter presents the design of the data-flow simulator prototype, which was developed to demonstrate the data-flow in the system. The purpose of the simulator is to provide a clear and simplified representation of the overall data flows between different main components in the system. The simulator can be regarded as a digitalization of the system sketch shown in figure 1. The simulator is not intended as an operator tool, but rather as a utility for demonstration and learning.

The simulator GUI consists of:

- Boxes that represent the main components
- Pushbuttons for triggering data transfer
- Text boxes for user input data
- Indicator arrows showing the direction of data-flow

The direction indicators either blink green or red; green indicates upstream and red indicates downstream communication. When a component receives data, it blinks in a bright color to visualize that the component receives data. The process order is that the arrow next to the sending component blinks first, followed by the receiving component. The arrow indicators were developed with suggestions and help from Chat GPT [20].

The prototype first checks the CSV file and retrieves data if available. If the data is not present in the CSV file, or if the CSV file does not exist, the prototype retrieves data from the configuration file instead.

The application consists of two different paths for the digital protection relays.

- **Traditional path**: Data flows from the relay to a physical station located RTU using the IEC 61850 communication standard, and the IEC 60870-104 protocol for communication between the RTU at the station level and SCADA at the control center level. This path can be seen on the bottom left side of figure 10 for Substation A.
- **Digitized path**: Data flows from the relay to a virtual RTU running as software in the IT-environment of the control operational center. This route also uses IEC 61850 but in a different way, since the RTU is software-based and located at the control center level. Communication here between the station and the control center level is handled by IEC 61850, while IEC 60870-5-104 is used between the virtual RTU and SCADA inside the control center level. This path can be seen on the bottom right side in figure 10 for Substation B.

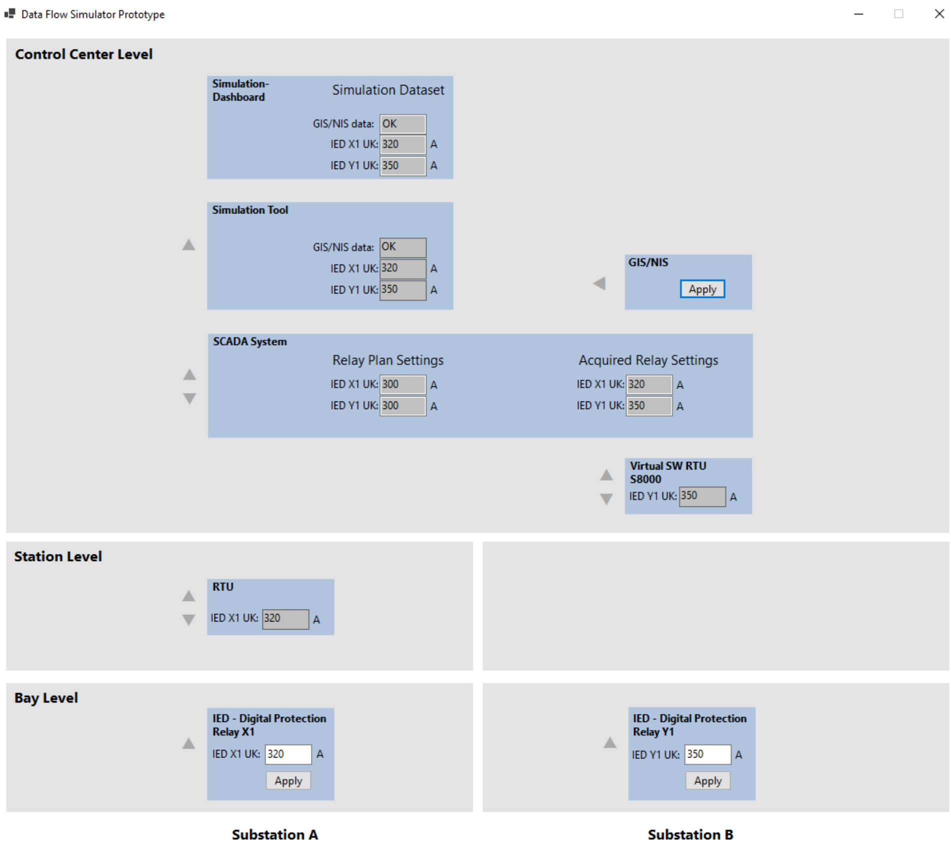


Figure 10 - Data-flow simulator prototype GUI showing the main elements and data-flow in the dynamic relay settings scope. The GUI is inspired by the architecture shown in figure 1.

The GUI illustrates how the UK value is updated from the digital protection relays, through the different RTUs into SCADA and the simulation tool and finally displayed in the simulation dashboard. The simulation dashboard also shows that the grid model from the GIS/NIS system is available, but in practice this is neglected in this project. When the operator modifies the UK values in the SCADA prototype GUI to the CSV file, the simulator will retrieve the UK values from the CSV file and flow downstream from the SCADA box through the RTUs and to the digital protection relays, and upstream to the simulation dashboard.

The content inside the SCADA and simulation dashboard boxes within the data-flow simulator prototype is highly simplified and only serves to demonstrate the principles of the system solution. Grey textboxes in the simulator indicate read only values, while white textboxes are editable for input values.

Figure 11 shows some snapshots of the different color indications in the data-flow simulator prototype.

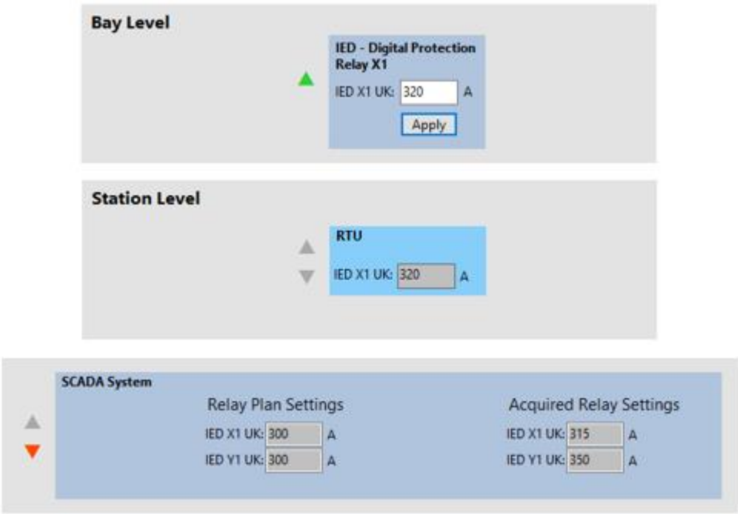


Figure 11 - Color indications in the data-flow simulator prototype GUI. The colors indicate direction of flow or receiving component of data.

The color indications in figure 11 mean the following:

- **Green arrow** = upstream data-flow
- **Bright blue highlight** = receiving component
- **Red arrow** = downstream data-flow

5.2 Implementation of data-flow simulator prototype

While the previous subchapter focuses on the design of the simulator, this subchapter focuses on the steps to implement the simulator. It describes the techniques, algorithms and other logic to achieve the functionality described in subchapter 5.1.

The data-flow simulator prototype was implemented in C# using Windows Forms within the .NET framework, and the GUI was developed in Visual Studio. Windows Forms was selected because it provides a simple and very flexible way to design GUIs for prototype applications [24].

The different components in the prototype GUI like buttons and text boxes were implemented as standard form elements with event-driven functionality. When the operator triggers data-transfer the prototype executes event handlers that update the graphical elements to reflect the data-flow. This includes color changes in arrows and components to visualize sending and receiving operations.

The simulator retrieves data from the CSV file if available. If the CSV file does not exit or lacks relevant data, the simulator loads default parameters from the JSON configuration file instead. This logic ensures that the simulator always can be initialized with valid data and uses the last registered data from the CSV file if they exist.

The prototype demonstrates two logical paths for simulating traditional and digitized data communication routes for physical and virtual RTUs. Each path represents a simplified version of the corresponding IEC communication standards, IEC 61850 and IEC 60870-5-104, to demonstrate how UK values are transferred through different system levels.

The implementation focuses on demonstrating a proof-of-concept solution for data-flow and synchronization instead of replicating the exact technical behavior of the communication protocols, as this would require a lot more details and complexity. A PSEUDO code description of the simulator logic is described in Appendix B. It contains a high-level description of key functionality for the data-flow simulator prototype.

5.3 Results of data-flow simulator prototype

This subchapter presents the results obtained from running the simulator. The subchapter focuses on evaluating the simulator's performance, by examining its behavior under different conditions and assessing whether the defined requirements are met.

The data-flow simulator prototype successfully demonstrated the functionality intended and provided a clear visualization of how the data-flow between the main system components. The simulator displayed the interaction between the digital protection relay, RTUs, SCADA, GIS/NIS and the simulation tool and dashboard. It allows users to observe both upstream and downstream communication flows.

The color indications implemented give a solid overview of how the data moves in the system and makes it easy to follow the data transfer. The green arrows indicate upstream data transfer, and red arrows indicate downstream data transfer and receiving components are highlighted before they receive data.

The simulator correctly retrieved and updated data from the CSV file and used the JSON configuration file if necessary. This demonstrated that the data storage concept worked as intended.

The testing of the prototype showed that the prototype is stable and responsive and all the elements in the GUI updated correctly without errors.

The results confirmed that the simulator effectively visualizes the data-flow and helps to understand the system concept. The solution demonstrates the feasibility of using a lightweight Windows Forms prototype to represent and test the dynamic UK settings data-flow.

6 SCADA prototype

This chapter presents the results from the development of the SCADA prototype created in the project. The prototype aims to demonstrate how the retrieved UK values from the digital protection relays and other operational data in a substation can be visualized in a graphical user interface in SCADA. The goal is to demonstrate how the operators can manually adjust the UK values in the SCADA prototype back to the digital protection relays.

The prototype represents a simplified version of a real-time SCADA system showing a single line diagram for a substation. It is designed with similar principles used by Elvia when designing a substation in their SCADA.

The chapter is divided into three subchapters. Subchapter 6.1 describes the design phase of development. Subchapter 6.2 describes the implementation phase and subchapter 6.3 describes the results.

The development process using XAML was greatly inspired and supported by various YouTube channels providing tutorials and practical demonstrations such as *Kampa Plays* [25] and web pages like *Tutorialspoint* [26] and *Microsoft documentation* [27], in combination with Chat GPT as sparring partner [20].

6.1 Design of SCADA prototype

This subchapter presents the design of the SCADA prototype developed in the project. The goal of the prototype is to illustrate how the retrieved UK values and operational data can be visualized for the operators. The prototype also demonstrates how the operators can adjust the UK values and how to update it back to the digital protection relay. The design is based on demonstrating basic monitoring and control principles that are relevant to UK values, whereas other SCADA functionality and solutions are not implemented or highly simplified.

The SCADA substation GUI consists of:

- Grid components: bays, switches, transformers, busbars
- Generators: windmill and hydropower plant

- Sliders to adjust production from generators
- Loads: represented by house and factory icons
- Measurements: active and reactive power, voltage, current and UK value
- Information boxes: to show substation details
- Pushbutton and slider for applying adjusted UK value

The measurements are based on parameters in the configuration file. The prototype checks the CSV file for measurements first, and if the exits they are retrieved and used in the prototype. If the values are not present in the CSV file, the prototype will use the configuration file instead.

The GUI allows the user to adjust the generated production from a windmill and a hydropower plant; the production icons were created by using ChatGPT for help [20]. The combined produced power is distributed on four different loads divided on two different substations; the loads icons were created using ChatGPT for help [20]. The load ratio can be defined by the user in the configuration file, and measurements in the GUI will be recalculated whenever the production power changes using the production sliders.

The GUI has two bays (X1 and Y1) in separate substations (A and B) where the operator can manually adjust the UK value in the measurement box. When the operator changes the UK values in the measurement box and apply the value, the value will be written to digital protection relay X1 for Substation A or Y1 for Substation B in the data-flow simulator prototype.

Figure 12 and 13 shows a snapshot of the SCADA prototype, showing a GUI of Substation A and B. Both GUIs consist of bays, switches, transformers, busbars and measurement boxes. Appendix A shows a table that explains the symbols in the SCADA GUI. The GUIs also consist of buttons to switch between the Substation GUIs. This is because the substations are logically connected in this system. Substation A supplies substation B with power through bay B1 in substation A and received through bay A1 in substation B. The SCADA GUI is inspired by Siemens Spectrum Power solutions used by Elvia [28].

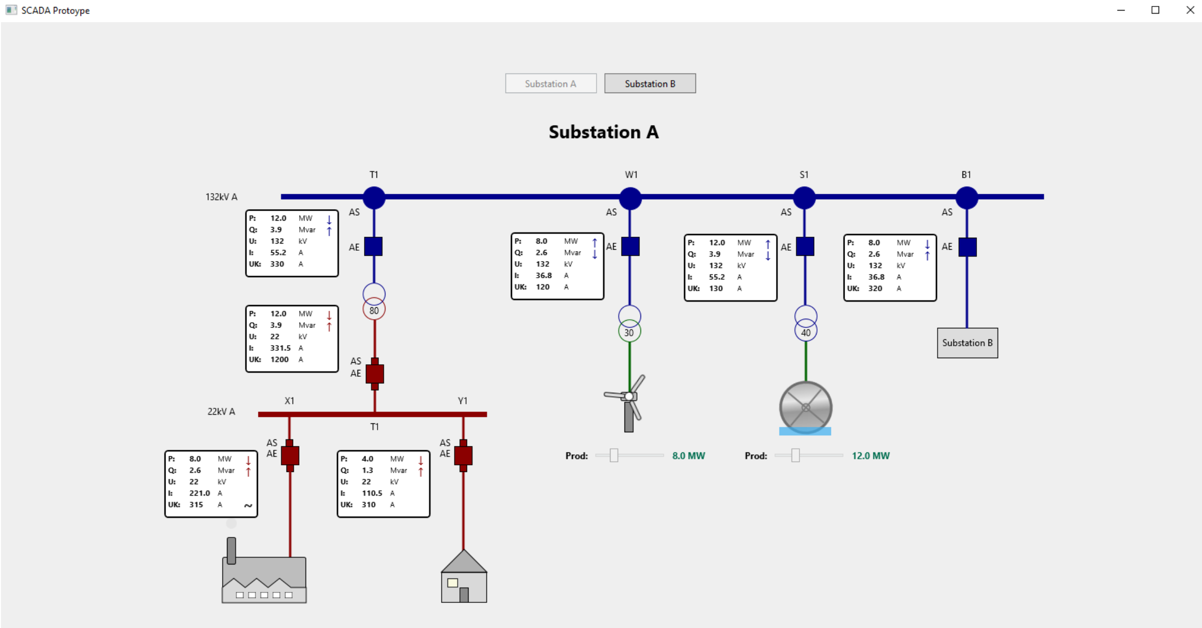


Figure 12 - SCADA Prototype GUI of substation A. Showing an overview of a single line diagram of a substation with measurements, switches, busbars, transformers, production and loads. The GUI is used by the operators for monitoring and control in the control center [28].

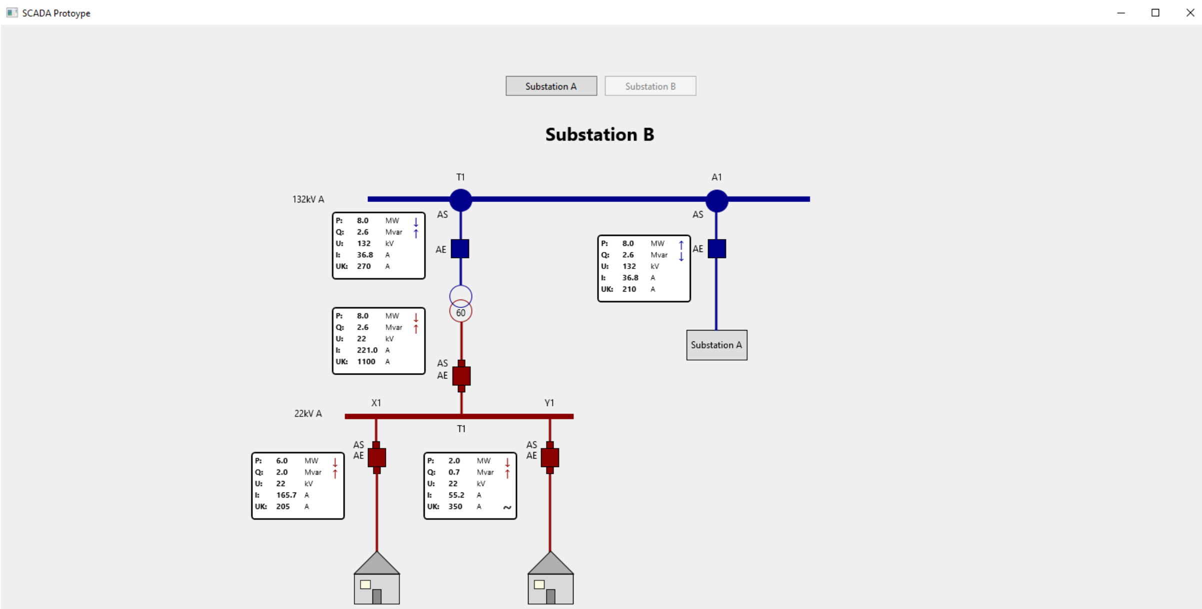


Figure 13 - SCADA Prototype GUI of substation B. Showing an overview of a single line diagram of a substation with measurements, switches, busbars, transformers and loads. The GUI is used by the operators for monitoring and control in the control center [28].

Figure 14 shows a snapshot of two popup windows that appear after the operator right clicks the adjustable UK values in the GUI. Adjustable UK values in SCADA are shown by the "~" symbol in the measurement boxes. The operator can then adjust the UK value using a slider and the "+/- "buttons to fine tune the value. The combination of the slider and "+/- "buttons is used instead of textbox to avoid invalid inputs from the operator. The outer limits of the UK values are defined in the configuration file for separate bays.

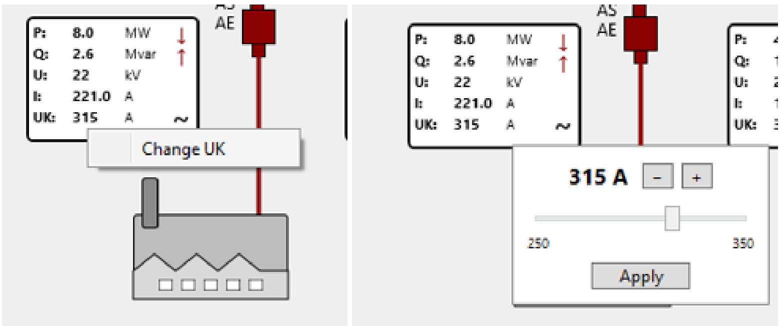


Figure 14 - Change UK menu showing the popup menu when the operator right clicks the UK value in the SCADA GUI. The UK value is adjusted by the operator and the applied to the digital protection relays in the substation.

Figure 15 shows a snapshot of a substation detail popup box that opens when the operator clicks the substation name. The details are read-only values in the SCADA GUI but can be adjusted in the configuration file. The details are exported to CSV and used in the web-based simulation dashboard. The idea is to supply the operator with main details of the substation so the operator quickly can get an overview of the substation details.

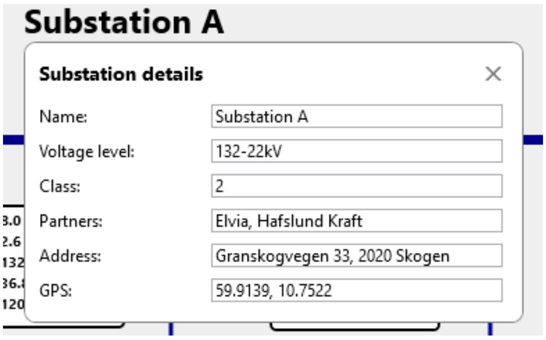


Figure 15 - Substation details showing the popup menu when the operator clicks the substation name in the SCADA GUI.

6.2 Implementation of SCADA prototype

The implementation was performed using XAML and C# within the .NET framework. The prototype logic and data handling were implemented in C#, while the GUI was implemented in XAML. This solution resulted in a combination of both front-end and back-end functionality.

The SCADA prototype uses the same storage concepts as the other prototypes, CSV for operational data and JSON for configuration data. The prototype uses parameters from the CSV file if they are available or data from the configuration file as fallback parameters. The SCADA prototype has read/write access to the CSV file and updates the CSV file with measurements and updated UK values that are written to the digital protection relays. While the prototype has read access only to the configuration file.

The communication and updating process is both events driven, and timer based. When the operator manually updates the UK value in the SCADA prototype, the new value is written to the CSV file, the data-flow simulator also uses a timer and reads the updated UK value from the CSV file and updates the GUI in the simulator. The measurements are written to the CSV file whenever there is a change in power production. The prototype uses a timer and checks the CSV for updated UK values updated from the digital protection relays in the data-flow simulator and updates the SCADA prototype GUI if applicable.

The prototype was developed and tested using Visual Studio. The program provided necessary tools for designing the GUIs, debugging and visualization.

6.3 Results of SCADA prototype

The SCADA prototype successfully demonstrated the main principles of monitoring and control in the dynamic relays settings solution. The prototype demonstrates how the UK values are retrieved from the digital protection relays, and how the operator can adjust the parameters in the GUI.

The updated UK values are written to the CSV file from the SCADA prototype and retrieved by the data-flow simulator prototype and written downstream through the RTU and to the digital protection relays. This verifies that the shared data concept storage functioned as intended.

The prototype retrieved and displayed the operational data from the CSV file and updated it if changes in measurements occurred in the SCADA prototype. If the data was not present in the CSV file, the fallback parameters in the configuration data were used.

The developed SCADA prototype was created as a simplified proof-of-concept solution. The prototype does not communicate with any real industrial OT equipment, but uses simulations, however the results demonstrated that the fundamentals logic for handling the data worked as intended. The prototype therefore demonstrates the feasibility of extending the concept to a more complete SCADA system in the future for Elvia.

A PSEUDO code description of the SCADA logic is described in Appendix C. It contains a high-level description of key functionality for the SCADA prototype.

7 Web-based simulation dashboard prototype

This chapter describes the development of a web-based simulation dashboard prototype. The dashboard was created to serve operators the opportunity to perform grid simulation using operational data. The development was carried out using C# and the .NET framework with the combination of Blazor WebAssembly and MudBlazor. All the development was carried out in Visual Studio.

The chapter is divided into three subchapters. Subchapter 7.1 describes the design phase of development. Subchapter 7.2 describes the implementation phase and subchapter 7.3 describes the results.

The development process using MudBlazor was greatly inspired and supported by various YouTube channels providing tutorials and practical demonstrations such as *Just Blazor Programming* [29], *dotnet* [30] and *ZeBit* [31] and web-pages like *Microsoft documentation* [32], in combination with Chat GPT as sparring partner [20].

7.1 Design of web-based simulation dashboard prototype

This subchapter presents the design of the web-based simulation dashboard prototype in the project. The dashboard was designed to provide a highly user-friendly interface for operators working with operation of the power grid. The dashboard was designed to provide simulation opportunities using operational data from the SCADA prototype. The goal of the dashboard is that the operators can perform grid simulations resulting in recommended UK settings, that the operators can update in the SCADA prototype. For simplicity the dashboard does not include the GIS/NIS model which would be required in a fully developed system. A fully developed system would also require a simulation tool running back-end serving relevant formulas and models for performing high complexity simulations. The dashboard prototype used simplified formulas instead for demonstration purposes only.

The dashboard web page consists of the following pages:

- Home
- Substation
- Measurements
- Simulation

These pages are described in the next chapters in the report.

7.1.1 Home page

The "Home" page represents the front page of the dashboard. It is the first page that appears when the user opens the web interface. It contains some green shortcut buttons to the other pages and some quick stats that demonstrate how many substations, bays and measurements that are monitored and retrieved in the dashboard. The menu in the left border can be hidden if the user presses the hamburger icon next to the header, but it is default open when the user opens the web page

Figure 16 shows a snapshot of the "Home" page, showing the different pages and calculated statistics based on retrieved data.

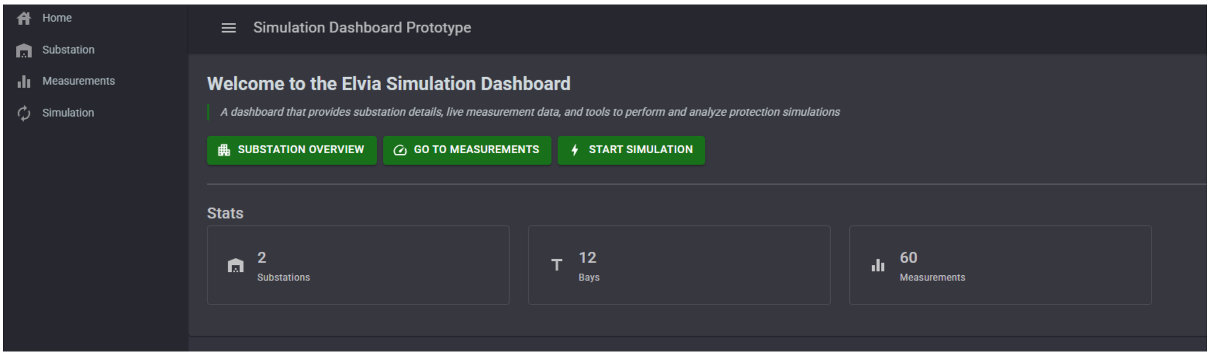


Figure 16 – "Home" page in the web-based simulation dashboard prototype showing some quick stats of the received operational data from SCADA prototype.

7.1.2 Substation page

The “Substation” page contains main information about each substation. The goal is to quickly provide the operator with some main details about each substation. The details are retrieved from SCADA, and the picture retrieved is stored with the CSV file. The page has a filter menu where the operator can choose which substation to view.

Figure 17 shows a snapshot of the “Substation” page, showing the details retrieved from the CSV file and SCADA and a picture of the substation.

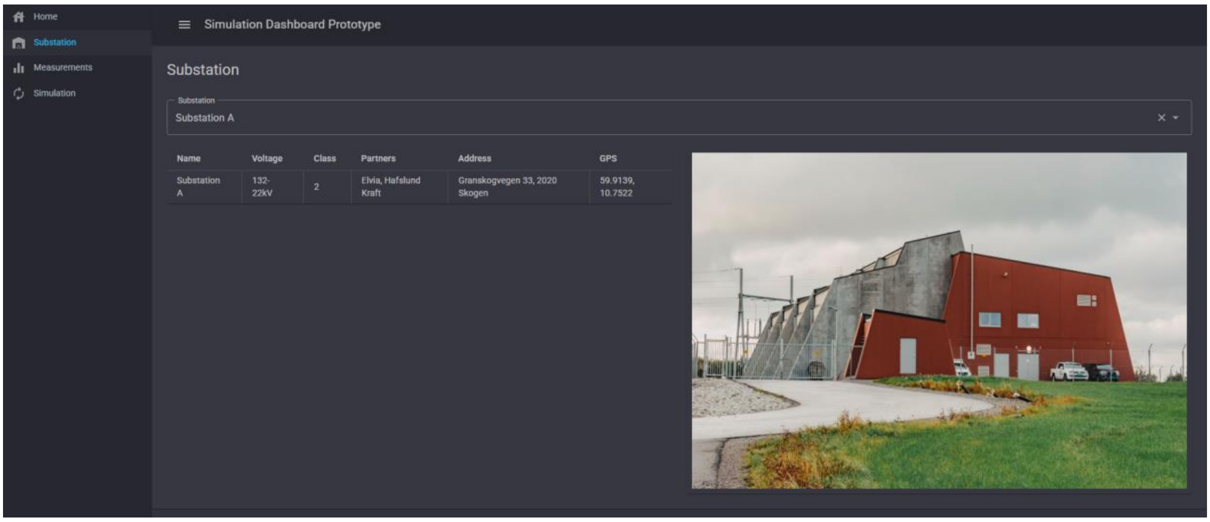


Figure 17 - "Substation” page in the web-based simulation dashboard prototype showing an example of a substation with substation details received from SCADA. The substation is from Opstad in Hå [33].

7.1.3 Measurements page

The “Measurement” page contains all the measurements retrieved from SCADA. The page has a filter where the operator can choose which measurements to look at. The operator must choose a substation and then choose to view all the measurements from the substation or apply more filters to narrow down the results. The page has a “refresh” button that the user can use to retrieve the latest measurements from the CSV file. This is to avoid continuous changes in the measurements if they are retrieved continuously. Because this can lead to noise for the operator analyzing the data.

Figure 18 shows a snapshot of the “Measurement” page, showing the measurements retrieved from the CSV file and SCADA for the selected substation and bay.

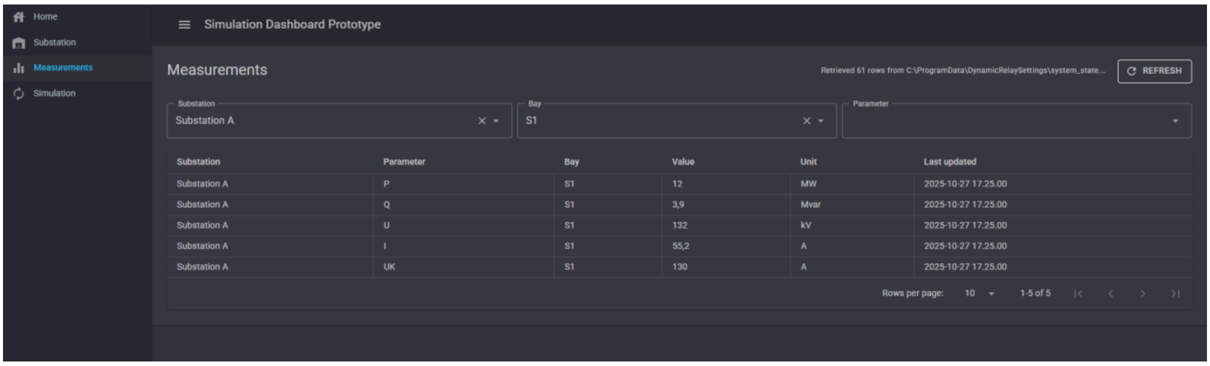


Figure 18 - "Measurement" page in the Web-Based Simulation Dashboard Prototype showing how the measurement data revied from SCADA appear after applying a filter.

7.1.4 Simulation page

The “Simulation” page is divided into several sections. The operator must first choose a substation and a bay to analyze at the top of the page in the filter menu.

After the operator chooses a substation and bay the “Information Section” and “Simulation Section” appears, this can be shown in figure 19. The “Information Section” contains retrieved details of the current measurement (I_measured), limiting current of the bay (I_max) and UK value (UK) from CSV. It also contains results when comparing the values up against each other. The table under “Measured Data Results” shows which values that are compared and the resulting margin and status. The status column shows if the margin is within the limits or not. These calculations are based on measurement data, and not simulations. They therefore demonstrate a real time picture of the status in the system.

The status coloring uses the following logic:

- Margin >= 10 % = Green - OK
- Margin < 10 % = Orange - Warning
- Margin =< 0 % = Red – Alarm

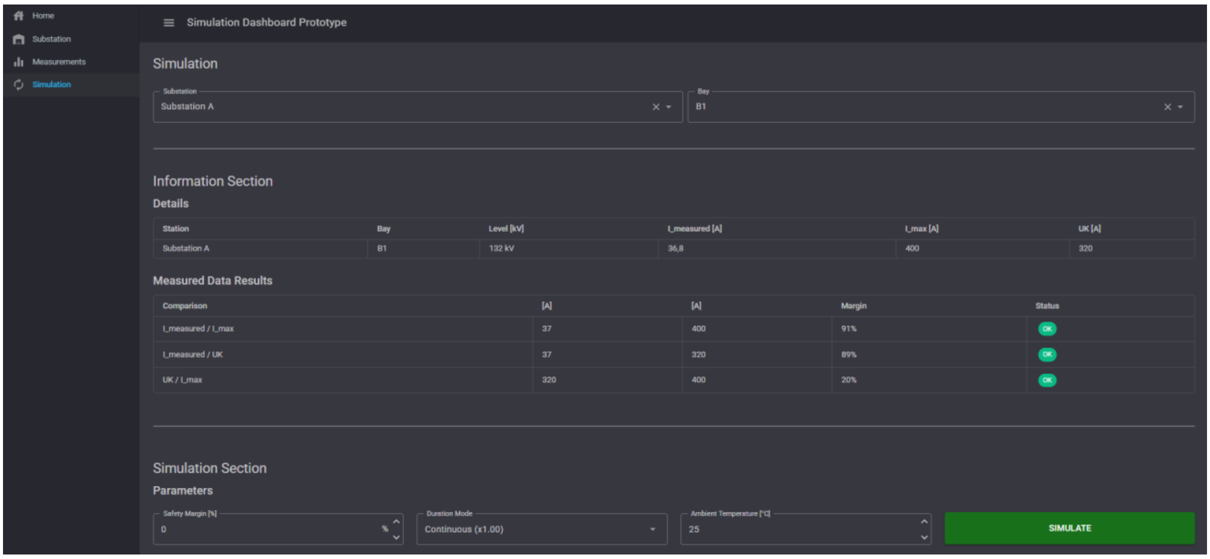


Figure 19 - "Simulation" page in the web-based simulation dashboard prototype showing the information section after applying a filter.

If the operator wants to perform simulations using the retrieved data from CSV he can use the “Simulation section”. The “Simulation section" contains some parameters that can be adjusted by the user to simulate different outcomes. The outcomes are in the same structure as in the “Measured data results” section, but with different results. The results also show a recommended UK setting value that is calculated based on the measurement data and the chosen parameters. The operator can therefore simulate different use cases using different parameters. The parameters and formulas in the simulation section are simplified and just to demonstrate a proof-of-concept on how the simulation section can look like.

The three parameters for simulation are: Safety Margin, Duration Mode and Ambient Temperature and they are all parameters that will affect the calculation of the I_max value of the limiting component in the simulations of the selected bay, this can for example be a cable. Increasement of safety margin will lower the I_max value, more aggressive duration mode will increase the value and increasement of ambient temperature will lower the I_max value. All the parameters will affect the calculated recommended UK value. The formulas in the simulation section were created by inspiration of formulas suggested by Chat GPT [20]. Appendix E shows the formulas used in the simulation with examples of calculation.

In a fully developed system this section will require a lot more work but is almost neglected in this project due to its high complexity, lack of GIS/NIS data and lack of back-end simulation tool. Figure 20 shows a picture of three adjustable parameters, a “Simulate” button and the recommended UK settings value. The results appear after pressing the “Simulate” button.

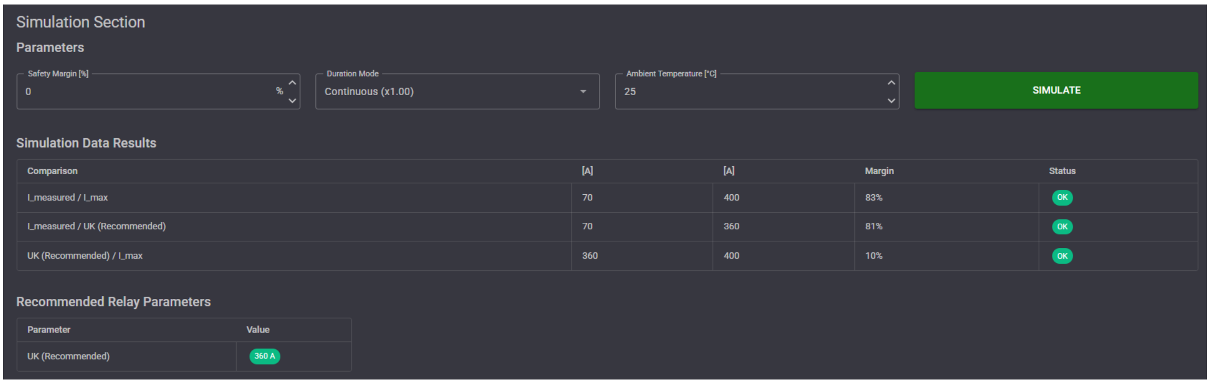


Figure 20 - Simulation Section in the "Simulation" page showing the simulation section with the recommended UK settings after adjusting the simulation parameters and pressing the simulation button.

7.2 Implementation of web-based simulation dashboard prototype

The implementation was carried out using C# and the .NET framework in Visual Studio. The dashboard was implemented as a Blazor WebAssembly prototype. This means that all the logic is executed in the web browser without the requirement of a backend server.

The GUI was built using MudBlazor. It provides a library of predeveloped and modern components ready for implementation in the C# code. Examples of predeveloped components are charts, tables, buttons, sliders etc. The components are customizable, easy to implement and work well with other components [34].

The operational data, substation details and limitation details are all retrieved from CSV files in the storage concept.

A PSEUDO code description of the web-based simulation dashboard logic is described in Appendix D. It contains a high-level description of key functionality for the web-based simulation dashboard prototype.

7.3 Results of web-based simulation dashboard prototype

The web-based simulation dashboard prototype demonstrates the principles that was requested in the specification. The simulation dashboard gives the operators the opportunity to look at substation details, measurements or perform simple simulations and receive a recommended UK setting by using the different pages in the dashboard.

The goal of the prototype was not to perform accurate simulations with accurate results, but to serve as a proof-of-concept solution that can be used for inspiration when developing a fully developed system.

In a fully developed system it may be wise to use simulation software as a back-end solution and connect it to the web-based simulation dashboard using an API if the simulation software offers this. This is because the simulations will most likely be very computationally demanding, which is better handled by dedicated software rather than running directly on the web page.

8 Shared data storage concept

This chapter will present shared data storage concept created in the project. The purpose of the storage concept is to ensure that all the prototypes can access and update the same operational data. The shared data storage concept serves as a simple, but effective solution to store operational data from different prototypes.

Subchapter 8.1 describes the design phase of the development. Subchapter 8.2 describes the implementation phase and subchapter 8.3 describes the results.

The development process of the shared data storage concept was greatly inspired and supported by various YouTube channels providing tutorials and practical demonstrations such as *Ravindra Devrani* [35] and *The Amazing Codeverse* [36], in combination with Chat GPT as sparring partner [20].

8.1 Design of shared data storage concept

The design of the shared data storage concept is based on the principle that all the prototypes can access it, but the read/write access is defined in the different prototypes respectively. Figure 9 shows the architecture of the implementation of the storage concepts for the CSV file and JSON file for the configuration and operational data.

The operational data are stored in a CSV file, which provides a simple structure for exchanging data between different independent prototypes. The configuration data such as parameter settings, default values etc. are stored in a JSON file.

When data is available in the CSV file the prototypes are designed to use these data but will use fallback values in the JSON file if the values are not present in the CSV file.

8.2 Implementation of shared data storage concept

The implementation was performed in C# using the .NET framework. It uses standard libraries for reading and writing CSV and JSON formats.

When a prototype starts it first checks if the CSV file exists and contains valid data. If the data is found, it is loaded and visualized. If the file is missing or incomplete, the prototype automatically falls back to reading from the JSON configuration file.

The implementation ensures that the prototypes can access common configuration data and operational data. The implementation does not require internet connection, and the file formats are compatible with multiple programming environments.

8.3 Results of shared data storage concept

The shared data concepts worked as intended during the testing of the prototypes. All the prototypes were able to access and update the same data source without conflicts. The solution ensures synchronization across the data-flow simulator prototype, SCADA prototype and the web-based simulation dashboard.

Figure 21 shows a snapshot of the CSV file for operational data from the SCADA prototype.

	A	B	C	D	E
1	substation,parameter,value,unit,last_updated				
2	Substation A,P_T1_132,12.0,MW,2025-10-21 17.14.03				
3	Substation A,Q_T1_132,3.9,Mvar,2025-10-21 17.14.03				
4	Substation A,U_T1_132,132,kV,2025-10-21 17.14.03				
5	Substation A,I_T1_132,55.2,A,2025-10-21 17.14.03				
6	Substation A,UK_T1_132,330,A,2025-10-21 17.14.03				

Figure 21 - CSV file of operation data showing a snapshot of some of the operational data from SCADA.

Figure 22 shows a snapshot of the CSV file for substation details from the SCADA prototype.

	A	B	C	D	E	F	G
1	Name	Voltage	Class	Partners	Address	GPS	
2	Substation A	132-22kV	2	Elvia, Hafslu	Granskogveg	59.9139, 10.7522	
3	Substation B	132-22kV	1	Elvia	Furuskogveg	60.4720, 7.5200	
4							
5							

Figure 22 - CSV file of substation details showing a snapshot of some of the substation details from SCADA.

Figure 23 shows a snapshot of the CSV file for limitation data for the limiting component for the different bays. These values are just generated to be able to demonstrate calculations and simulations in the web-based simulation dashboard.

	A	B	C
1	substation,parameter,I_max		
2	Substation A,T1_132,400		
3	Substation A,T1_22,1200		
4	Substation A,X1,300		
5	Substation A,Y1,315		
6	Substation A,W1,125		
7	Substation A,S1,150		
8	Substation A,B1,400		
9	Substation B,X1,250		
10	Substation B,Y1,200		
11	Substation B,A1,250		
12	Substation B,T1_132,300		
13	Substation B,T1_22,1200		

Figure 23 - CSV file of limitation data showing some generated limitation data that is used for simulations in the Web-Based Simulation Dashboard.

Figure 24 shows a snapshot of the JSON file for configuration data. The configuration file contains data that is read by the different prototypes.

```
"Simulation": {
    ////Config Parameters

    ///Relay Plan UK [A]
    //Substation A
    "DefaultRelayPlanX1A_UK": "300",
    //Substation B
    "DefaultRelayPlanY1B_UK": "300",

    ////SCADA Prototype Config Parameters
    ///Production [MW]
    //Substation A
    "DefaultWaterProductionA": "12",
    "DefaultWindProductionA": "8",
    //Substation B

    ///UK [A]
    //Substation A
    //Parameters
    "DefaultWindProductionA_UK": "120",
    "DefaultWaterProductionA_UK": "130",
    "DefaultX1A_UK": "302",
    "LowerX1A_UKLimit": "250",
    "UpperX1A_UKLimit": "350",
    "DefaultY1A_UK": "310",
    "DefaultB1A_UK": "320",
    "DefaultT1A_22UK": "1200",
    "DefaultT1A_132UK": "330",
    //Substation B
    //Parameters
    "DefaultX1B_UK": "205",
    "LowerY1B_UKLimit": "150",
    "UpperY1B_UKLimit": "250",
    "DefaultY1B_UK": "206",
    "DefaultA1B_UK": "210",
    "DefaultT1B_22UK": "1100",
    "DefaultT1B_132UK": "270",
```

Figure 24 - JSON configuration data file showing a snapshot of some of the configuration data used by some of the prototypes.

The shared data solution worked well as part of a proof-of-concept solution for demonstrating principles. They allowed for quick testing, and easy implementation with the prototypes. The concepts do have limitations which are described earlier in the report. For a fully developed system a more robust database or cloud-based storage solution would be required.

9 Discussion

This chapter discusses the main findings in the results chapter. The discussion chapter focuses on how the prototypes performed; their limitations and the lessons learned throughout the development process. The discussion chapter will also describe the implications this type of solution will have for Elvia and which considerations that should be made for future implementation and development of a fully working system. The components discussed in the chapter is – the data-flow simulator prototype, SCADA prototype, web-based simulation dashboard prototype and the shared data storage concept.

9.1 Strengths of the prototypes

The goal of the project was to demonstrate a proof-of concept of dynamic relay settings to enable dynamic grid operation. The strengths of the prototypes lie in their ability to give an overview and demonstrate how the concept works across multiple levels and in different systems in the project. The prototypes combined successfully managed to graphically visualize the data-flow from OT components in a substation into the IT environment where data was processed, simulated and updated back to the OT components.

The prototypes have different main functionalities to demonstrate separate concepts of the system. The data-flow simulator prototype provides a clear and simple visualization of data transfer between the main components in the system across all the prototypes, that can be used for demonstration and training purposes. The SCADA prototype gives a simple overview of single line diagrams of multiple substations and how they are connected. The SCADA prototype has limited functionality but shows the main concept on how the operators can monitor different measurements, including the UK value. The prototype also illustrates well how a solution can appear to manually update the UK value from SCADA. This solution does not exist in Elvia's SCADA system today, but hopefully this can be used as inspiration for implementation. The web-based simulation dashboard prototype demonstrates how operational data from SCADA can be retrieved and used for grid simulation purposes. The simulations are very simplified and do not represent a realistic image of how a fully developed system will appear due to high complexity, but it serves as proof-of-concept for how the operators could evaluate and test different scenarios with different parameters when performing simulations.

The prototypes were all linked together using a shared data storage concept. This solution successfully synchronized data between the prototypes using CSV and JSON files. The solution can be considered as a lightweight solution, with limitations, but it demonstrates the concept of shared data storage without using complex back-end solutions or network connection.

The combined system including all the components can therefore be considered as a lightweight functional proof-of-concept for dynamic relay settings. It demonstrates the feasibility of modular and scalable architecture.

9.2 Limitations of the prototypes

The prototypes do have their limitations regarding modern expectations in power-sensitive environments with high demands for performance and uptime. The prototypes are not connected to any OT infrastructure. This means that real components like digital protection relays, RTUs or SCADA systems are not involved. This also means that the communication protocols or standards that are used in today's industry are not included in the project. This limits the ability to evaluate the solution achievability, performance, latency and security aspects.

Another important major limitation is the lack of GIS/NIS data and full protection model integration from the digital protection relays, since this report only focuses on the UK value. These elements are crucial to include to perform realistic grid simulations. However, these elements introduce high complexity and demand a high level of competence on the subject to include them in a fully developed solution. The simulation dashboard also has limitations in the simulation functionality. The functionality is just based on some simplified formulas rather than real power system models. This reduces realism and precision but demonstrates the principles of performing simulations.

The storage concept worked well for testing but is not scalable for production use. Especially CSV files, since this is an old and limited file format that does not work well for frequent real-time data exchange, updates and multi-user environments. Both CSV and JSON can be subjects to version conflict, lacks access control and cyber security mechanisms required in a modern shared data storage concept.

9.3 Lessons learned in the project

The development process of prototypes and shared data storage concept resulted in several valuable lessons for further development and work.

The importance of conducting solid and thorough research work on data formats and integration methods is evident. Due to time restraint, decisions on data formats for storage solutions were probably made too early. The data-flow simulator prototype was developed first using WinForms but should ideally be developed using XAML like the SCADA prototype to achieve a more consistent design language, improved maintainability and easier integration. XAML also resulted in a prototype that looks more modern, easier to adjust and flexible.

The project also highlighted the need for better code structure and the opportunity for reusability. The combination of time pressure and lack of programming experience and knowledge lead to some solutions that could have been solved in more efficient ways. But this would also have required significantly more research and development time, which would likely have slowed down progress and resulted in less to demonstrate as proof-of-concept, but it would have been easier to continue and probably offer more flexibility and scalability.

A key learning point is that the development of each prototype conducted in valuable insights and experiences that influenced the development of the next prototype. This shows how early testing and visualization can improve further development, understanding and introduction of new ideas and solutions. The idea through the hole project was to develop a solution that can be used to explain the solution to stakeholders that have limited knowledge on the project with graphical visualization instead of static figures and texts.

The project working as a proof-of-concept gives valuable lessons regarding development, but more importantly it serves as a starting point for the development of fully developed system for Elvia. The prototypes can be shown to relevant stakeholders, and it will be easy for them to give input on improvements, changes or ideas they like about the prototypes.

9.4 Implications and further work

For Elvia the developed prototypes and concept have several implications. The project demonstrates that achieving digitalization of protection relay settings and data-flow regarding this, is feasible and can be implemented gradually through step-by-step integration. All the work in the project can be used by Elvia to demonstrate the concept or to expand the prototypes to introduce more complexity.

Before starting the development of a fully developed solution there are several technical and organizational considerations to evaluate and resolve. Here is a list of the main considerations to look in to:

Technical requirements

- Find out if the relay settings can be retrieved from the digital protection relays. It is recommended to start with one parameter like the UK value to SCADA.
- Find out if the UK settings can be updated from SCADA to the digital protection relay from SCADA.
- Find out the requirement to achieve the two-way communication from the digital protection relays to SCADA. Is IEC 61850 a requirement? Are there limitations regarding the control system suppliers? How to track the updated settings regarding changes, errors and confirmation that modifications have been performed?
- Evaluate where all the protection relay parameters should be stored. It is likely that today's SCADA system is not the ideal location for all of these. External storage database should be considered. Maybe by using the virtual S8000 RTU in combination with a database.
- Find a suitable simulation tool that can run as back-end to the simulation dashboard. PSS Sincal or PowerFactory are examples used in the power industry today.
- Identify how the grid model can be retrieved from GIS/NIS, and which data that are necessary to perform accurate simulations.

Integration and data exchange

- Identify which data that should be retrieved from SCADA to the simulation dashboard, and how to retrieve them. Protocols like OPC UA or MQTT can be considered.
- Define infrastructure requirements. Identify hardware, network and software infrastructure that is necessary to support a fully functioning system.
- Clarify time synchronization requirements. It is important that all systems use synchronized timestamps to ensure accurate data-flow across SCADA, relays and simulations.

Cybersecurity and data management

- Define cyber security requirements. Any type of data communication is subject to cyber security threats, and since Elvia is working with high sensitivity data, this is a very important topic. The use of standards and protocols for the power industry should be preferred instead of other solutions.
- Establish data ownership and responsibility. Identify which department is responsible for each system, maintenance and data management.
- Define backup and recovery requirements. Specify how the data can be restored, how often backup is performed etc.
- Perform risk and vulnerability analyses. It is important to perform risk analyses to identify potential weaknesses and threats to the solution. This includes throughout the entire solution. How does the solution cope with fail scenarios, incorrect data, missing data, hacking, and infrastructure failure. The outcome of this analysis should serve as input to system design and a cyber security strategy.

Project organization and development process

- Investigate existing market solutions and consult with relevant suppliers. Before starting any type of development, it is recommended to review commercial and research-based solutions in the market and discuss them with potential partners or vendors.

- Establish a project organization with dedicated key resources with suitable knowledge on the separate fields in the project, and how to make decisions in the project. Remember to include the operators that will use the finished product in the development process to ensure usability at early stage.
- Create test cases and validate the results. The consequences of bad simulation results and decisions can be massive, so accurate validation is crucial. It can be wise to start with laboratory equipment first and then choose a substation pilot that has little consequence if the testing goes bad.
- Create a roadmap with milestones where the complexity is gradually increased through iterative development.
- Create a plan for training and competence. It is important that the operators, engineers and developers receive necessary documentation and training to be able to use, fix and maintain the solution.

10 Conclusion

This chapter concludes the work by summarizing the project objectives, findings and achievements.

10.1 Summary of objectives and results

In this project the main objective was to develop and demonstrate a proof-of-concept of dynamic relay settings to enable more flexible and dynamic operation of the power grid. The concept was demonstrated by developing three prototype applications – the data-flow simulator prototype illustrating data-flow between components, SCADA prototype demonstrating monitoring and parameter updates and the web-based simulation dashboard prototype to demonstrate simulation principles. The shared data storage concept successfully enabled synchronization of data between all the prototypes.

In summary, the project successfully achieved its objective, by demonstrating the feasibility of dynamic relay settings using simulations and SCADA functionality.

10.2 Key findings

Several key findings were identified throughout the project.

- **Graphical visualization is powerful –** By building prototypes one can at an early stage easier understand how systems will interact with each other, identify challenges and weaknesses and communicate complex concepts to relevant stakeholders.
- **Iterative development adds value –** Each of the prototypes were mostly developed separately. But the development of a prototype influenced the development of the next one.
- **Lightweight concepts can demonstrate complex ideas -** The project showed that the use of simple concepts like CSV and JSON can effectively demonstrate the idea of shared data and synchronization between different systems.

- **Research and planning are crucial at early stage –** Early decisions can typically influence the path of direction in the project. Therefor it is crucial to spend enough time at early stage in the project. Using more time on research and architecture design could have increased reusability, scalability and integration potential in the project.
- **Proof-of-concept helps to reduce project risks –** The proof-of-concept of the dynamic relay settings project demonstrates how helpful lightweight prototypes can be to visualize ideas, concepts and solutions before starting a full-scale development process. The results of the project can be used to update and find better and more accurate requirements for the fully developed system.

10.3 Final remarks

The project has shown that dynamic relays settings are feasible and can potentially be realized through gradual and iterative development where complexity is gradually increased. For Elvia the project provides valuable insight by demonstrating the concept and identifying technical and organizational challenges that should be addressed before development and large-scale implementation.

The project can be seen as a first step to achieving dynamic relay settings in dynamic grid operation for Elvia, and the lessons learned throughout the project forms a solid starting point to modernize their protection systems and enable more flexible and intelligent power grid operation.

References

[1] B. Davidsen, "Slik økte Norges største nettselskap kapasiteten i nettet," 03 03 2025. [Online]. Available: https://www.europower.no/nett/slik-okte-norges-storste-nettselskap-kapasiteten-i-nettet/2-1-1782616. [Accessed 03 10 2025].

[2] A. K. Rosvold, "Store Norske Leksikon," 02 06 2025. [Online]. Available: https://snl.no/vindkraftverk. [Accessed 15 11 2025].

[3] NTB Kommunikasjon, "NTB," 06 10 2023. [Online]. Available: https://kommunikasjon.ntb.no/pressemelding/18007258/fullt-i-stromnettet-pa-ostlandet-frem-til-2035?publisherId=17847555&lang=no. [Accessed 03 10 2025].

[4] Energidepartementet, "STRØMNETTET ER VIKTIG INFRASTRUKTUR," 18 09 2025. [Online]. Available: https://energifaktanorge.no/norsk-energiforsyning/kraftnett/. [Accessed 13 11 2025].

[5] Regjeringen, "Strømnettet i Norge," [Online]. Available: https://www.regjeringen.no/no/dokumenter/nou-2022-6/id2918464/?ch=4. [Accessed 22 09 2025].

[6] NVE, "Tonsen," 16 08 2021. [Online]. Available: https://www.nve.no/om-nve/nves-listefoerte-kulturminner/transformatorstasjoner/tonsen/. [Accessed 03 10 2025].

[7] Tekna, "Brukermøte kontrollanlegg 2024," [Online]. Available: https://www.tekna.no/kontrollanlegg/. [Accessed 22 09 2025].

[8] Delgado Relay Protection Referance, "Introduction to Digital Relays," 31 05 2024. [Online]. Available: https://delgadorelayprotectionreference.com/public/digital-numerical-relays/digital-numerical-relays-Introduction_to_Digital_Relays/. [Accessed 03 10 2025].

[9] Siemens, "Feeder and Overcurrent Protection SIPROTEC 7SJ85," 13 11 2025. [Online]. Available: https://assets.new.siemens.com/siemens/assets/api/uuid:6ef1c43e-cfe6-4f28-88a2-63e18e469d2b/siprotec-7sj85-profile.pdf. [Accessed 13 11 2025].

[10] EMAC, "7SJ85 SIEMENS," EMAC, 13 11 2025. [Online]. Available: https://emacstores.com/tienda/sistemas-de-alimentacion/protecciones-electricas/7sj85-siemens/. [Accessed 13 11 2025].

[11] Wikipedia, "Remote terminal unit," 14 07 2025. [Online]. Available: https://en.wikipedia.org/wiki/Remote_terminal_unit. [Accessed 03 10 2025].

[12] Siemens, "Automation and remote terminal units - SICAM A8000," [Online]. Available: https://www.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/substation-automation/automation-and-remote-terminal-units-sicam-a8000-series.html. [Accessed 26 09 2025].

[13] Siemens, "SICAM S8000," [Online]. Available: https://www.siemens.com/us/en/products/energy/energy-automation-and-smart-grid/power-automation-platform-sicam8/sicam-s8000.html. [Accessed 28 09 2025].

[14] EMQX, "IEC 61850 Protocol: Features, Information Model, and Combination with MQTT," 30 11 2023. [Online]. Available: https://www.emqx.com/en/blog/iec-61850-protocol. [Accessed 03 10 2025].

[15] T. Charton, D. Binon and R. Troost, "Where has the Process Bus got us so far?," 01 12 2021. [Online]. Available: https://www.pacw.org/versions-of-the-standard-what-you-need-to-know-2. [Accessed 03 10 2025].

[16] iGrid T&D, "igrid-td.com," [Online]. Available: https://www.igrid-td.com/smartguide/communicationprotocols/iec-60870-5-104/. [Accessed 03 10 2025].

[17] Wikipedia, "Windows Presentation Foundation," 29 10 2025. [Online]. Available: https://en.wikipedia.org/wiki/Windows_Presentation_Foundation. [Accessed 15 10 2025].

[18] Wikipedia, "Extensible Application Markup Language," 15 06 2025. [Online]. Available: https://en.wikipedia.org/wiki/Extensible_Application_Markup_Language. [Accessed 12 10 2025].

[19] Axosoft, "Gitkraken," Axosoft, 13 11 2025. [Online]. Available: https://www.gitkraken.com/. [Accessed 13 11 2025].

[20] OpenAI, "ChatGPT," OpenAI, 12 11 2025. [Online]. Available: https://chatgpt.com/. [Accessed 05 11 2025].

[21] H.-P. Halvorsen, "Youtube," 4 05 2021. [Online]. Available: https://www.youtube.com/watch?v=I6Mq79Dai7M. [Accessed 14 10 2025].

[22] Programentor, "Youtube," 17 11 2023. [Online]. Available: https://www.youtube.com/watch?v=Tix0H-YJ9Fk. [Accessed 22 10 2025].

[23] Microsoft, "Create a Windows Forms app in Visual Studio with C#," 17 10 2025. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=visualstudio. [Accessed 17 10 2025].

[24] Wikipedia, "Windows Forms," 5 06 2025. [Online]. Available: https://en.wikipedia.org/wiki/Windows_Forms. [Accessed 16 11 2025].

[25] Kampa, "Youtube," 03 11 2022. [Online]. Available: https://www.youtube.com/watch?v=t9ivUosw_iI&list=PLih2KERbY1HHOOJ2C6FOrVXIwg4AZ-hk1&index=1. [Accessed 01 11 2025].

[26] Tutorialspoint, "Tutorialspoint," 10 11 2025. [Online]. Available: https://www.tutorialspoint.com/xaml/xaml_quick_guide.htm. [Accessed 10 11 2025].

[27] Microsoft, "XAML tools documentation," 5 11 2025. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/xaml-tools/?view=vs-2022. [Accessed 5 11 2025].

[28] Siemens, "Making power grid complexity manageable," 11 11 2025. [Online]. Available: https://www.siemens.com/global/en/products/energy/grid-software/operation/grid-control.html?acz=1&gad_source=1&gad_campaignid=11428336540&gclid=CjwKCAiA2svIBhB-EiwARWDPjrx6GxjBArP2MsLo_ZHHZ05V5xoTcUMxvmqYsnmkSeCCqoBWGzlT7xoCv44QAvD_BwE. [Accessed 11 11 2025].

[29] J. B. Programming, "YouTube," 23 03 2022. [Online]. Available: https://www.youtube.com/@justblazorprogramming. [Accessed 29 10 2025].

[30] D. Roth, "Youtube," 25 01 2024. [Online]. Available: https://www.youtube.com/watch?v=llDc88XE--Q. [Accessed 2 11 2025].

[31] ZetBit, "YouTube," 24 06 2023. [Online]. Available: https://www.youtube.com/@zetbit. [Accessed 27 10 2025].

[32] Microsoft, "ASP.NET Core Blazor," 06 11 2025. [Online]. Available: https://learn.microsoft.com/nb-no/aspnet/core/blazor/?view=aspnetcore-10.0&WT.mc_id=dotnet-35129-website. [Accessed 06 11 2025].

[33] Hå Element AS, "Trafostasjon på Opstad," 11 11 2025. [Online]. Available: https://haelement.no/referanser/trafostasjon. [Accessed 11 11 2025].

[34] MudBlazor, "Explore MudBlazor," Gardnet AB, 12 11 2025. [Online]. Available: https://mudblazor.com/docs/overview. [Accessed 12 11 2025].

[35] R. Devrani, "Youtube," 19 05 2023. [Online]. Available: The development process using MudBlazor was greatly inspired and supported by various YouTube channels providing tutorials and practical demonstrations such as Just Blazor Programming [29], dotnet [30] and ZeBit [31] and web-pages like Microsoft documenta. [Accessed 06 11 2025].

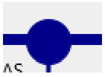
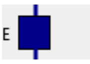
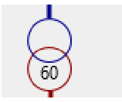
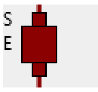
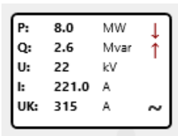
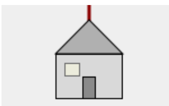
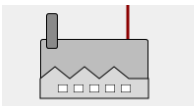
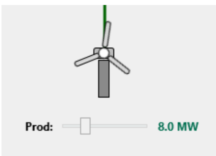
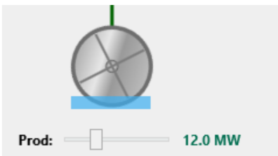
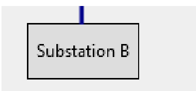
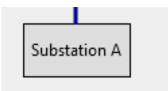
[36] H. Raza, "Youtube," 25 01 2025. [Online]. Available: https://www.youtube.com/watch?v=QkmBmek2l7g. [Accessed 04 11 2025].

[37] V. Willumsen and O. R. Valmot, "Snart 150 år med strøm i Norge: Her er elektrisitetens historie," 11 02 2018. [Online]. Available: https://www.tu.no/artikler/snart-150-ar-med-strom-i-norge-her-er-elektrisitetens-historie/429910. [Accessed 03 10 2025].

[38] Siemens, "Digital protection relays and control - SIPROTEC 5," [Online]. Available: https://www.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/protection-relays-and-control/siprotec-5.html. [Accessed 25 09 2025].

Appendices

Appendix A – SCADA prototype icon explanation table

Icon	Explanation
	Disconnector
	Circuit breaker
	Transformer
	Withdrawable circuit breaker with disconnector
	Busbar
	Measurement box - Arrow symbol indicates direction of measurement flow - Wave symbol indicates adjustable UK value
 	Load
 	Production
 	Shortcut buttons between views in SCADA

Appendix B – Data-flow simulator prototype PSEUDO code

```
DATA-FLOW SIMULATOR PROTOTYPE PSUEDO CODE
//Knut Erland Strætkvern

PROGRAM START

1. INITIALIZATION

    - Load config.json
    - Check if system_state.csv exits
        * if missing
            * create system_state.csv
            * write default values to system_state.csv
    - Read UK_X1 and UK_Y1 from system_state.csv
        * if values missing
            * use default values from config
            * write them to system_state.csv

    - Insert initial values into:
        * IED, RTU, SWRTU, Relay plan, SCADA, Simulation tool, Simulation dashboard
    - Configure timers
        * two timers for polling CSV for UK_X1 and UK_Y1
        * animation timers for blinking and arrow indicators
    - Create arrow indicators
        * IED, RTU, SWRTU, SCADA, Simulation tool, Simulation dashboard

2. MAIN FLOW: IED -> RTU/SW RTU -> SCADA -> CSV -> SIMULATION TOOL -> SIMULATION DASHBOARD
when "Apply" button is pressed on IED

    - Validate valid integer input
        * if invalid
            * prompt NAN! and abort flow
    - Trigger arrow and blinking animation IED -> RTU/SW RTU -> SCADA
    - Update UK value in RTU/SW RTU and SCADA textboxes
    - Write UK values to system_state.csv
    - Read UK value from system_state.csv to Simulation tool
    - Trigger arrow and blinking animation SCADA -> Simulation tool -> Simulation dashboard
    - Update UK value in Simulation tool and Simulation dashboardand textboxes

3. MAIN FLOW: SCADA PROTOTYPE -> CSV -> SCADA (SIMULATOR) -> (RTU/SW RTU -> IED and SIMULATION TOOL -> SIMULATION DASHBOARD)
continous evry 1s for UK_X1 and UK_Y1

    - Read UK_X1 and UK_Y1 from system_state.csv
        * if value changed
            * blink SCADA box in simulator
            * update SCADA textboxes with new retrieved UK values from system_state.csv
            * trigger parallell arrow and blinking animation downstream to IED and upstream to Simulation dashboard
            * update UK value in corresponding textboxes

4. MAIN FLOW: GIS/NIS -> CSV -> SIMULATION TOOL -> SIMULATION DASHBOARD
when "Apply" button is pressed on GIS/NIS

    - Write "OK" status to CSV with "GIS/NIS" as element text
    - Trigger arrow and blinking animation GIS/NIS -> Simulation tool -> Simulation dashboard
    - Read status from system_state.csv
    - Update UK value in Simulation tool and Simulation dashboard

PROGRAM END
```

Appendix C – SCADA prototype PSEUDO code

```
SCADA PROTOTYPE PSEUDO CODE
//Knut Erland Strætkvern

PROGRAM START

1. INITIALIZATION

    - Load config.json
    - Check if system_state.csv exits
        * if missing
            * create system_state.csv

    - Read UK_X1 and UK_Y1 from system_state.csv
        * if value missing
            * use default value from config
            * write the default UK value to system_state.csv

    - Load substation data from config
        * name, voltage level, class, partners, address, GPS
        * insert into read-only text fields for Substation A and B
        * write all metadata to substation_details.csv

    - Initialize production in Substation A
        * adjust wind and hydro sliders from config
        * calculate initial P/Q/I for W1 and S1 from config
        * insert initial values into measurement boxes for W1 and S1

    - Calculate initial power flow
        * use total generation sum in Substation A from W1 and S1
        * distribute P/Q to bays in Substation A and Substation B
        * update P/Q/U/I for:
            * T1 (132 kV and 22 kV) in A and B
            * X1A, Y1A, X1B, Y1B
            * B1 (export from Substation A to B) and A1 (import from substation A to B)
        * apply I/UK colour coding in all measurement boxes

    - Write initial P/Q/U/I/UK values for Substation A and B to system_state.csv

    - Configure timer
        * one timer for reading system_state.csv every 1 s for UK_X1 and UK_Y1

    - Connect production slider events
        * when wind or hydro slider drag is finished
            * write updated measurements to system_state.csv

2. MAIN FLOW: PRODUCTION CHANGE, MEASUREMENT UPDATE -> CSV

    - When wind or hydro production is changed
        * Recalculate P/Q/I for S1 and W1
        * Update measurement boxes
        * Recalculate overall power flow
        *update P/Q/U/I for:
            * T1 in Substation A and B
            * X1A, Y1A, X1B, Y1B
            * B1 (export) and A1 (import)
        * update I/UK colour coding

    - When wind or hydro slider drag is finished
        * write all updated P/Q/U/I/UK values for Substation A and B to system_state.csv
```

```
3. MAIN FLOW: UK ADJUSTMENT IN SCADA PROTOTYPE -> CSV

    - When operator opens UK popup for X1 in substation A
        * load min/maks UK limits for X1 from config.json
        * set popup slider to current UK value for X1

    - When UK is adjusted and "Apply" is pressed for X1 in substation A
        * set UK_X1 from slider value
        * write UK_X1 to system_state.csv
        * update I/UK colour for X1

    - When operator opens UK popup for Y1 in substation B
        * load min/maks UK limits for X1 from config.json
        * set popup slider to current UK value for Y1

    - When UK is adjusted and "Apply" is pressed for Y1 in substation B
        * set UK_Y1 from slider value
        * write UK_Y1 to system_state.csv
        * update I/UK colour for Y1

4. MAIN FLOW: UK ADJUSTMENT IN DATA-FLOW SIMULATOR PROTOTYPE -> CSV -> SCADA PROTOTYPE

    - Read UK_X1 for substation A from system_state.csv evry 1s
        * if uk value changed
            * update UK_X1 text in SCADA prototype measurement box
            * update I/UK colour for X1

    - Read UK_Y1 for substation B from system_state.csv evry 1s
        * if uk value changed
            * update UK_Y1 text in SCADA prototype measurement box
            * update I/UK colour for Y1

PROGRAM END
```

Appendix D – Web-based simulation dashboard prototype PSEUDO code

```
WEB-BASED SIMULATION DASHBOARD PROTOTYPE PSUEDO CODE
//Knut Erland Strætkvern

PROGRAM START

1. INITIALIZATION

    - Define file locations for inputs
        * system_state.csv
        * substation_details.csv
        * limitation_details.csv

2. HOME PAGE

    - When page load:
        * read system_state.csv
        * parse each row into seperate categories
    - Count statestics
        * nr of substations
        * nr of bays
        * nr of measurements
    - Display interface
        * show introduction text
        * show page shortcuts
        * show counted statestics

3. STATION PAGE

    - When page load:
        * read substation_details.csv
        * parse each row into seperate categories
    - Build filter list
        * substation
    - When substation is chosen in filter
        * show retrieved metadata from substation_details.csv in table
        * show image of substation

4. MEASUREMENTS PAGE

    - When first load or on Refresh click
        * read system_state.csv
        * parse each row into seperate categories
        * update "last loaded" timestamp
    - Build filter list
        * substation
        * bay
        * parameter
    - Show table based on selected filter
```

```
5. SIMULATION PAGE

    - When first load
        * read system_state.csv
        * parse each row into seperate categories
        * load limitation_details.csv (I_max per bay)
    - Build filter list
        * substation
        * bay
    - Show "Information Section"
    - Show "Details"
    - Show details table base on selected filter
        * table with level (kV), I_measured, I_max, UK for filtered substation and bay
    - Show "Measured Data Results"
    - Show measured data results table base on selected filter
        * table with comparisons of level (kV), I_measured, I_max, UK
        * compute and show margin and status indikators = OK - green / Warning - orange / Alarm - red
    - Simulation parameters that user can edit
        * safety margin [%]
        * duration mode
        * ambient temperature [°C]
    - When SIMULATE button is pressed
        * adjust I_max using safety margin, duration mode factor and temperature derating
        * compute recommended UK
    - Show "Simulation Data Results"
        * table with comparisons of level (kV), I_measured, I_max, UK but with recommended UK
    - Show "Recommended Relay Parameters"
        * table with UK (Recommended)

PROGRAM END
```

Appendix E – Simulation formulas for web-based simulation dashboard prototype

Safety margin formula:

$$I_{max.adjusted} = I_{max.base} * (1 - \frac{safety\ margin[\%]}{100}) \quad (1)$$

Example:

$$I_{max.adjusted} = 400A * \left(1 - \frac{10}{100}\right) = 360A$$

Duration mode formula:

$$I_{max.adjusted} = I_{max.adjusted} * k \quad (2)$$

Where:

$continous \rightarrow k = 1.0$
$1\ hour \rightarrow k = 1.1$
$30\ min \rightarrow k = 1.25$

Example:

$$I_{max.adjusted} = 360A * 1.1 = 396A$$

Ambient temperature formula:

$$I_{max.adjusted} = I_{max.adjusted} * [1 - (T - 25) * 0.005], \quad for\ T > 25^{o}C \quad (3)$$

Example:

$$I_{max.adjusted} = 396A * [1 - (30 - 25) * 0.005] = 386.1A, \quad where\ T = 30^{o}C$$

Recommended UK value formulas:

$$UK_{I_{max.base}} = I_{max.adjusted} * (1 - 0.10) \quad (4)$$

$$UK_{min} = I_{measured} * (1 + 0.05) \quad (5)$$

$$UK = \max(UK_{I_{max.base}}, UK_{min}), \quad (6)$$

to ensure that recommended UK is not below $I_{measured}$

$$UK_{recommended} = round(UK) \quad (7)$$

Example:

$$UK_{I_{max.base}} = 396A * (1 - 0.10) = 356.4A$$

$$UK_{min} = 300A * (1 + 0.05) = 315A$$

$$UK = \max(UK_{I_{max.base}}, UK_{min}) = 356.4A$$

$$UK_{recommended} = round(UK) = 356A$$

Margin formula:

$$Margin = \frac{B - A}{A} * 100\% \quad (8)$$

<u>Example:</u>

$$Margin = \frac{I_{max.adjusted} - I_{measured}}{I_{max.adjusted}} * 100\%$$

$$Margin = \frac{386A - 70A}{386A} * 100\% = 82\%$$