ENERGY DATA AND AUTOMATION ARCHITECTURE REPORT

Deliverable D3.4

Co-funded by the European Union

6GSNS

The TARGET-X project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No: 101096614

GRANT AGREEMENT	101096614
PROJECT TITLE	Trial Platform foR 5G EvoluTion – Cross-Industry On Large Scale
PROJECT ACRONYM	TARGET-X
PROJECT WEBSITE	www.target-x.eu
PROJECT IDENTIFIER	https://doi.org/10.3030/101096614
PROGRAMME	HORIZON-JU-SNS-2022-STREAM-D-01-01 — SNS Large Scale Trials and Pilots (LST&Ps) with Verticals
PROJECT START	01-01-2023
DURATION	30 Months
DELIVERABLE TYPE	Deliverable
CONTRIBUTING WORK PACKAGES	WP3
DISSEMINATION LEVEL	Public
DUE DATE	M24
ACTUAL SUBMISSION DATE	15.12.2024
RESPONSIBLE ORGANIZATION	RWTH-ACS
EDITOR(S)	Manuel Pitz
VERSION	1.0
STATUS:	Final
SHORT ABSTRACT	This deliverable provides an overview of the planned hardware and software architecture. Software wise storage, visualization and deployment is discussed. On the hardware side the development of synchronization, 5G module carrier and analog acquisition is given.
KEY WORDS	Energy, 5G, Digital Transformation, Real-Time
CONTRIBUTOR(S)	Manuel Pitz, Benish Khan, Vincent Bareiß, Lena Thiesbrummel, Matthias Nowak, Calvin Katt, Simon Prein

Disclaimer

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the other granting authorities. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

Within the TARGET-X project, the integration of 5G in the energy, construction, manufacturing, and automotive verticals is evaluated. The goal is to identify already available features and possible features for 6G that can benefit these verticals. The energy vertical targets mainly the topics of monitoring, energy awareness and consumption. The developed software and hardware are not only used for grid-specific use cases, but also in other verticals to evaluate the energy consumption of a specific process.

This deliverable provides an overview of the hardware and software architecture for the energy vertical. This includes the edge cloud as well as the field devices. On the edge cloud side the architecture includes services for provisioning, supervision and control of edge devices, as well as storage and visualization of collected data. Furthermore, supporting services like access control and a configuration database are needed. On the field side, two different types of hardware are described. For the monitoring of voltages and currents, the 5G edgePMU is presented, and for energy awareness, an outdoor capable measurement box, the Meter-X is proposed. This box includes a 5G edgePMU at its core, but also additional metering and safety hardware needed for grid compliance. A detailed description and evaluation of different hardware like the 5G edgePMU, Meter-X and its components, evaluation equipment, 5G carrier board, synchronization, and edge infrastructure is given. The deployment in the energy, construction and robotics testbed is showcased and the first exemplary results of the measurements done are provided.

For the evaluation of different commercial 5G modules, a carrier board was developed to allow high-speed connection to the communication hardware. Thorough latency tests of two 5G modules, one from SIMCom and another from Quectel, were evaluated using an Ericsson EP5G private 5G network. Results show that, for different package sizes, the latency for the Quectel Module can be as low as 13ms and the SIMCom module shows higher latencies. Meanwhile, pricing does not differ in a major way between the non RedCap modules. RedCap Modules from Quectel were also acquired, testing is planned for early 2025. The pricing is with 118 € about one third of the other modules.
For Validation of the developed hardware, a reference voltage amplifier is developed to safely simulate a sinusoidal signal of a power grid. It has shown an amplitude error of below 0.04 % with a very flat frequency response in the relevant frequencies, outperforming a commercially available reference device.
First prototypes for an input stage system were developed and tested to allow for more flexible use compared to commercial systems. It was tested for DC voltage measurements with an error of less than 200 µV within the full ±5 V output scale.
For proper synchronization of measurements to a global time reference, a PLL board was designed which can synchronize to a GPS time reference with a timing jitter between 15 ns and 100 ns, depending which oscillator option is selected. The resulting hardware products, Meter-X and 5G edgePMU, were successfully field tested with various loads ranging from small robots arms to large loads like tower cranes.

Table of Contents

List of Figures

List of Acronyms and Abbreviations

5G	5th Generation
5G SA	5G Standalone
5G NSA	5G Non-Standalone
AC	Alternating Current
ADC	Analog to Digital Converter
BOM	Bill Of Material
CSV	Comma Separated Values
DC	Direct Current
EP5G	Ericsson Private 5G
GOOSE	Generic Object Oriented Substation Event
HDF5	Hierarchical Data Format5
HTTPS	Hypertext Transfer Protocol Secure
LV grid	Low Voltage grid
MIB	Management Information Base
MQTT	Message Queuing Telemetry Transport
NAT	Network address translation
PCB	Printed Circuit Board
PDC	Professional Darts Corporation
PLL	Phase-locked Loop
PMU	Phasor Measurement Unit
PPS	Pulse Per Second
QoS	Quality of Service
RedCap	Reduced Capability
REST API	REpresentational State Transfer Application Programming Interface
RMS	Root Mean Square
SCADA	Supervisory Control and Data Acquisition
SD card	Secure Digital Card
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol

TVE	Total Vector Error
UI	User Interface
URLLC	Ultra-Reliable Low Latency Communication
USB	Universal Serial Bus
UUID	Universal Unique Identifier
VPN	Virtual Private Network
ZFS	Zettabyte File System

1 Introduction

Within the TARGET-X project, the 5G energy vertical is strongly focused on monitoring and energy awareness. These two use cases have a different set of requirements. For the energy awareness, an easy to use mobile unit with millisecond accurate time synchronization and single one measurement per second is needed (Meter-X). On the other hand, for the grid monitoring use case, high time accuracy in the microsecond range and tens of phasor estimations per second are needed (5G edgePMU). The results of the grid monitoring and energy awareness use case are especially important for the low voltage grid, where, until now, monitoring is not too common [2]. Especially with the increase in volatile energy sources which are often deployed in the low voltage grid, a better understanding is key for a stable operation in the future. To achieve this goal, a major component needed is a 5G-enabled software and hardware architecture that can utilize 5G edge cloud and low latency communication. This architecture has to be viewed holistically, including the field devices as well as the edge cloud infrastructure. Furthermore, it is important to include the actual communication components in this process. This will allow for an architecture that can be utilized across the different verticals and will increase energy awareness and grid monitoring capabilities in the fields of construction, manufacturing, and energy. The gathered data is planned to be stored beyond the project duration to allow for future use within an academic environment.

This deliverable provides an overview of the software architecture and 5G hardware development for the energy use case. This specific deliverable is the second revision of the architecture overview. It provides an extended overview of the developed software and hardware. Furthermore, it provides additional information on the 5G integration.

1.1 Document structure

This document is structured into five sections. Section 1 provides a brief introduction. Section 2 provides an overview of the planned architecture. The detailed description of the different components is split into two blocks. The first block (Section 3) focuses on the field device, in this case the Meter-X device for slow energy monitoring and the 5G edgePMU, which can measure voltage and current with high sampling rates and be synchronized to a GPS clock. Here, the hardware as well as the software side are discussed. The second block (Section 4) provides insights into the edge cloud software architecture. This includes communication, deployment, supervision, and storage. Finally, the document presents the conclusion in Section 5.

1.2 Relation to other activities

This deliverable is a final version of the energy data architecture that will be developed and deployed within the TARGET-X project. This architecture as well as the 5G-enabled energy measurement devices will be deployed within different verticals. Thus, there is a strong connection to WP5 (Construction) that is focused on construction and deconstruction processes and their energy footprint. Furthermore, the energy footprint of the robotics use case will be studied. This creates a strong connection to WP2 (Robotics) where the developed measurement device will also be deployed. For the definition of use cases and definition of requirements, collaboration with WP1

(Methodological assessment framework) is implemented. Finally, collaboration with WP6 (Technology evolution beyond 5G) is done for the evaluation of the 5G features used within the 5G-enabled measurement box.

2 Architecture overview

This section will provide an overview of the planned architecture. The main objective of this architecture is to streamline the deployment of and access to measurement data that is acquired in the field. A special focus is set on auto deployment to decrease the need for manual configuration and control actions, which is cumbersome when deploying a higher number of devices. Furthermore, the data storage and sharing question is tackled by splitting the storage into a long-term and a short-term storage, as well as always storing the metadata bundled with the actual measurements. This allows for reuse of the acquired data in follow-up projects and enables new data-driven use cases. It will provide insights into how the different components interact. The specific subcomponents will then be described in detail in the following subsections. As shown in Figure 1 the architecture is split into three major components. The top components are the services running within the edge cloud. Then the edge cloud is connected to the field devices via the 5G network through a VPN.

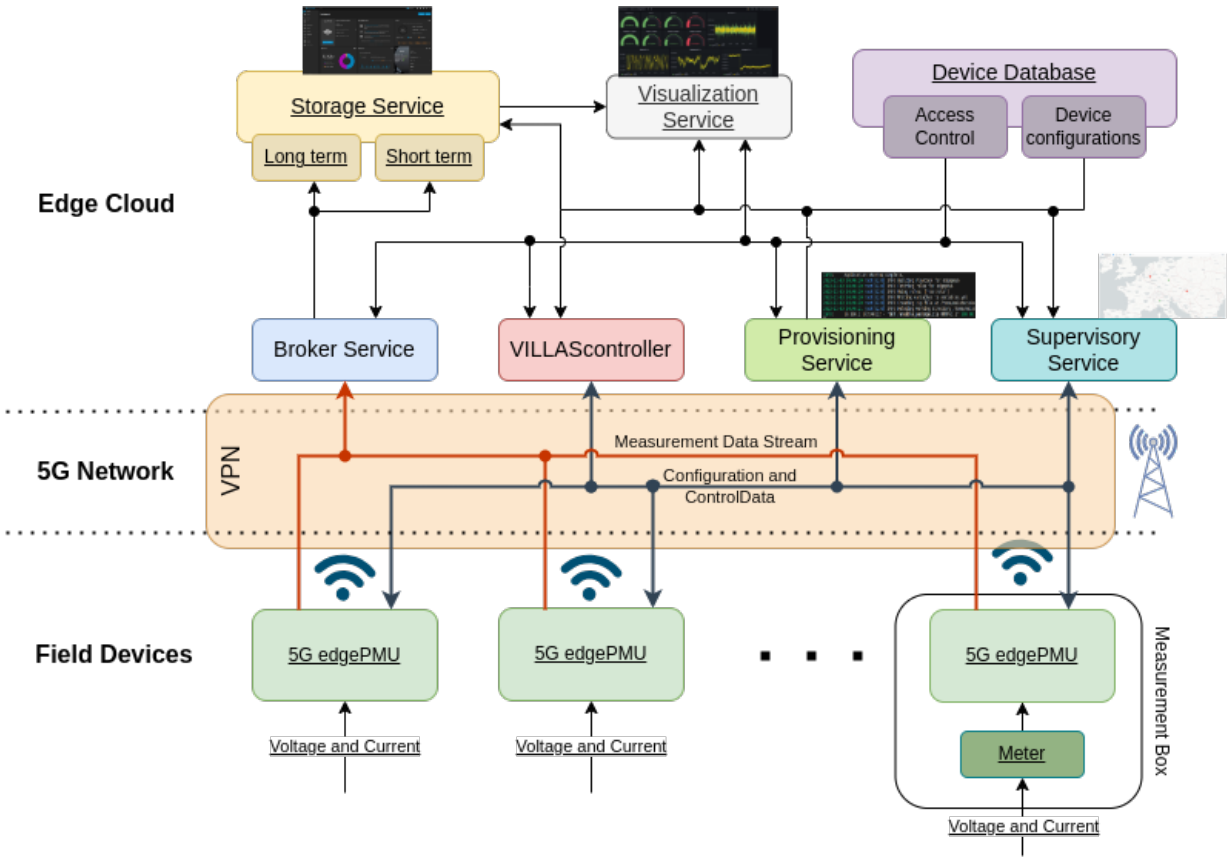

Figure 1: General architecture of the edge cloud and field side for the energy vertical.

The 5G edgePMU device is based on a 5G-enabled Raspberry Pi that is equipped with data acquisition hardware. The edgePMU is controlled (Section 3.4) and configured (Section 4.1) using the edge-cloud architecture. Furthermore, it streams the measurement results to the edge-cloud

infrastructure using a Broker Service such as Message Queuing Telemetry Transport (MQTT) (Section 4.3). To allow for a secure connection, the edgePMU is connected to the cloud infrastructure via a Virtual Private Network (VPN) (Section 4.2). This secures the basic communication and allows for reaching the device even behind firewalls or Network Address Translation (NAT) systems. Since multiple devices are connected to the same Broker Service, the different device data streams have to be isolated from each other. The access control is handled by the Access Control part of the Database Service.

A third major component is the Provisioning Service that is controlling the configuration of field devices (Section 4.1). To allow for a scalable infrastructure, the number of manual interventions for configuration and operation needs to be minimized. This is especially true for the configuration of the field devices. Apart from their device names and credentials, the devices could be installed with different hardware and software configurations that need device-specific configuration changes. Some examples are different 5G modules, GPS receivers or different data acquisition systems. To allow for simple and automated deployment, Ansible [1] is used. More details on the planned deployment stack are given in Section 4.1.

A fourth core component is the Storage Service. A major challenge is the long-term storage. The envisioned Storage Service can efficiently cope with long-term storage of either phasors, metering data or raw samples from multiple edgePMU devices (Section 4.5). To reduce the risk of major data corruption events, a file-based approach is under investigation. The files should contain not only the actual measurements but also metadata, e.g., deployed sensor types, grid information, position information and more. The storage should be split into two types. Long-term storage and short-term storage for fast visualization.

The last major component of the architecture is the Visualization Service (Section 4.6). This component should be accessible by the end users to allow access to short-term and long-term storage. It should also be possible to grant specific users access rights to specific datasets.

3 5G enabled Measurement Device

Within the project, it was decided to work on two different measurement devices, since the requirements of the energy awareness use case and the grid monitoring use case are partially contradicting each other. This applies in particular to long-term energy monitoring, where the use and extension of off-the-shelf metering devices is much more suitable than fast sampling of voltage and current. Furthermore, the deployment differs strongly. For energy awareness, a portable but rugged device that can be connected by non-electricians is beneficial, where in the case of energy monitoring the use case is usually stationary and external voltage and current sensors have to be adapted to the measurement equipment. For these reasons, the Meter-X device is developed for the application in the energy awareness trials, and the 5G edgePMU is further developed for the grid monitoring use case.

The Meter-X is a Raspberry Pi based hardware that utilizes open-source software to acquire measurements via Modbus RS485. The device is built as a rugged outdoor box and includes standard three-phase connectors for in-line measurements. The box is equipped with a 5G module and can automatically connect to the local edge cloud.

The edgePMU is also a Raspberry Pi-based hardware that utilizes open-source software to acquire measurements via the Universal Serial Bus (USB)-based Analog to Digital Converter (ADC) device. The software used to implement the interface to the ADC and for the handling of measurements is VILLASnode [8].

This is described in the following subsection. First, the hardware updates and auxiliary hardware will be discussed, and in a second subsection, the software stack and its extension are focused on. An introduction to the edgePMU is given in Deliverable 3.1 [3].

3.1 Hardware

This subsection provides information about the planned hardware developments. Here the major topics are the input stage that connects the edgePMU to the measured grid, needed hardware developments for the evaluation, the development of the 5G connectivity, and time synchronization.

3.1.1 Voltage Amplifier for Device Validation

For hardware validation, the testing should be as realistic as possible. This includes in particular the evaluation of the input stage, which includes dividers that might have a relevant frequency and amplitude dependency. To identify such behavior, a voltage amplifier was developed. The specifications required for device validation in this case are as follows: ground referenced outputs, ideally output voltages up to 253 V AC, as this is the typical upper limit of the European low voltage grid, output currents around 10 mA, low amplitude nonlinearity and phase error over the used output frequency and amplitude range. To evaluate the approach, a first revision was developed within a bachelor thesis. The hardware is depicted in Figure 2. The results of two tests are shown in Figure 3 and Figure 4. The first test is focusing on high-precision amplitude measurements based on bench multimeters (Figure 3) whereas the second test (Figure 4) utilizes a frequency response analyzer (Venable 350c) for the evaluation. It can be seen that the amplitude error stays below 0.04 % for the relevant frequencies. Furthermore, it can be seen that the overall frequency response is

flat and the phase is constant to around 10 kHz. In comparison with a commercially available device that does not satisfy the voltage requirements, it can be seen that with the in-house developed device, a better phase and amplitude performance can be achieved.

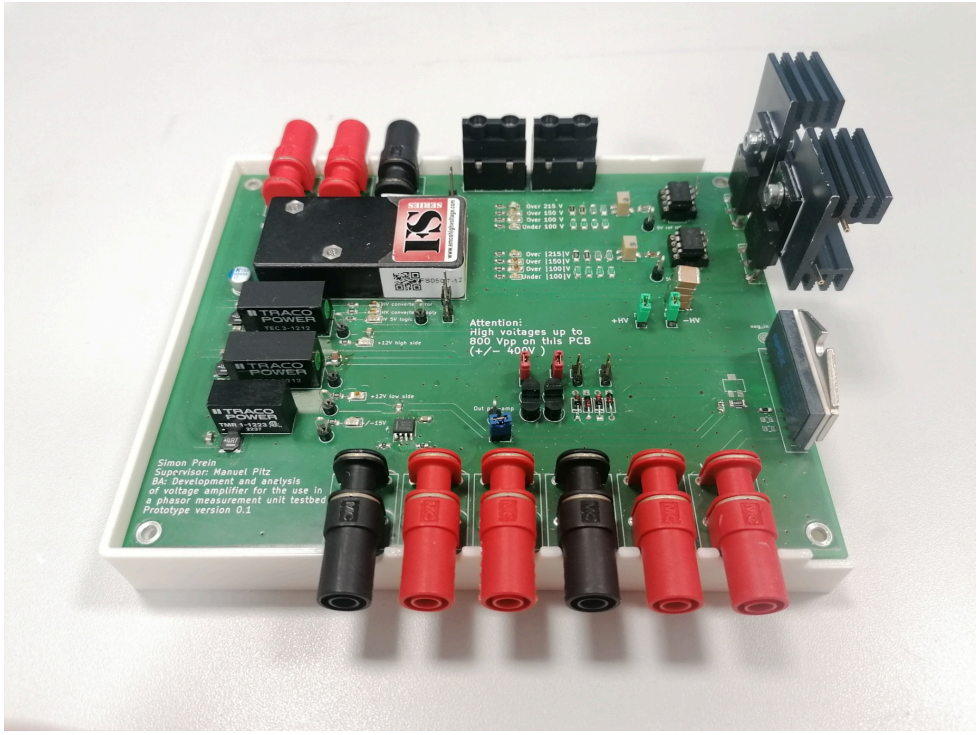

Figure 2: Picture of the developed prototype of the amplifier.

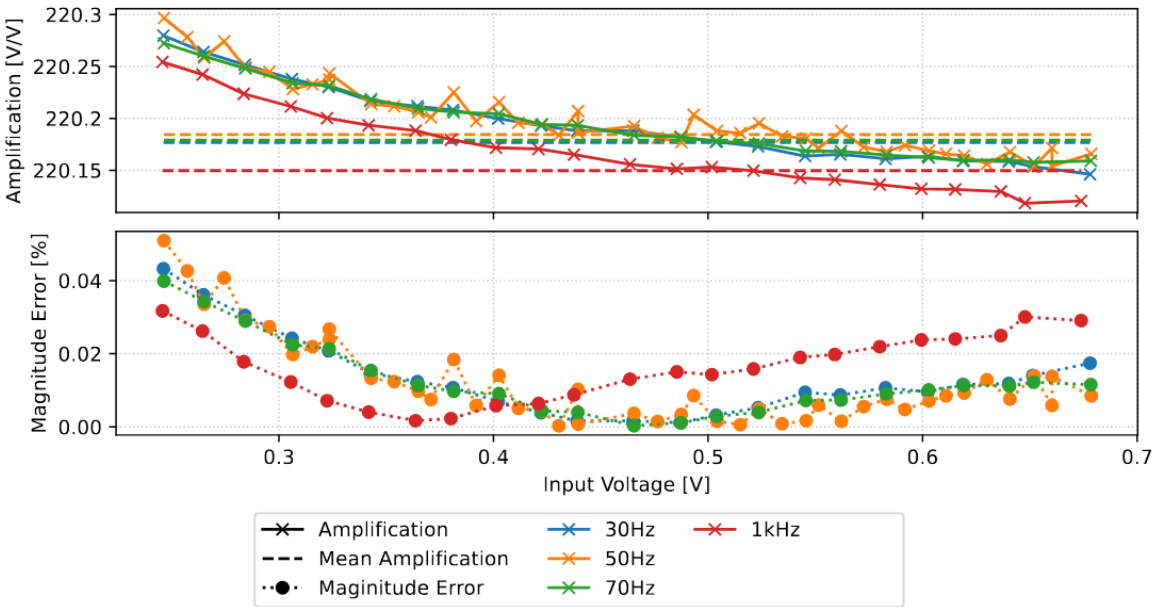

Figure 3: Test results for frequency dependency of amplification. Absolute amplification (top) and amplification error (bottom).

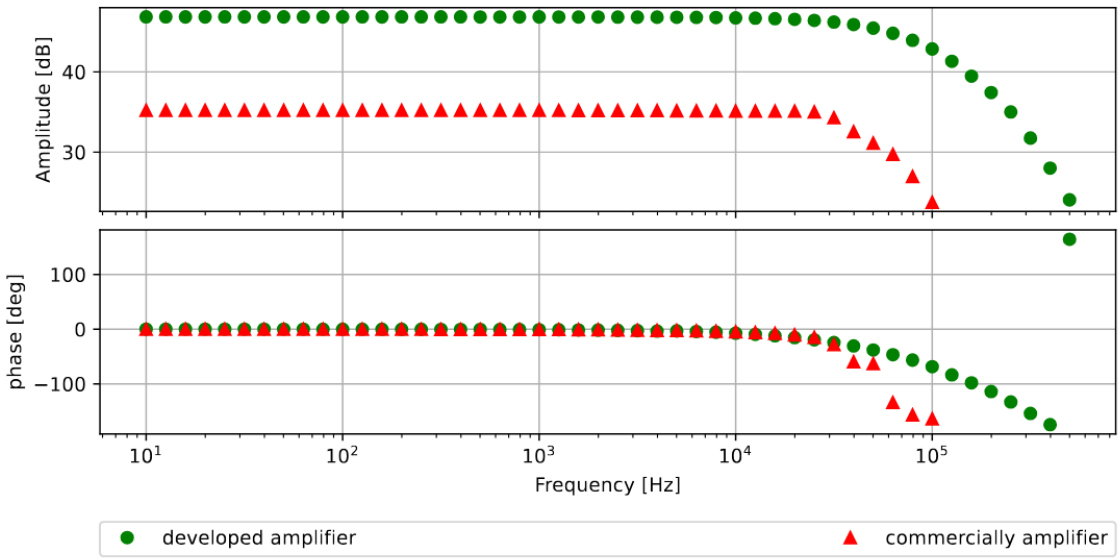

Figure 4: Comparison of amplitude and phase against a commercial amplifier.

3.1.2 Input Stage

The existing hardware only allows for the measurement of low voltage inputs, specifically for up to ±10 V on seven channels in parallel. These input levels are not directly compatible with most use cases, one of them is measuring the low voltage grid directly. Therefore, the edgePMU will be extended with a Low Voltage (LV) grid-compatible input stage that allows for direct measurement of 230 V Alternating Current (AC) grid. Furthermore, the input stage will provide inputs for current transducers so that the device can measure both voltage and current, not needing third-party signal

conditioning hardware. A first mockup of the planned hardware is shown in Figure 5. The shown version was already built as a prototype, and the first tests showed promising results. For the Direct Current (DC) case, the error for a ±10 V input configuration lies within 200 µV as shown in Figure 6. The second revision of the input protection is still under development. Part of the work done is the design and verification of adequate input protection in case of overvoltage. The designed input protection circuit has no significant effect on the accuracy of the measurement device. For the time being a less flexible and rather expensive voltage divider is used. The main difference of the commercial version and the in-house development is the easy adoption of the in-house version to the different use cases, where the commercial device can only be ordered with a limited number of configurations.

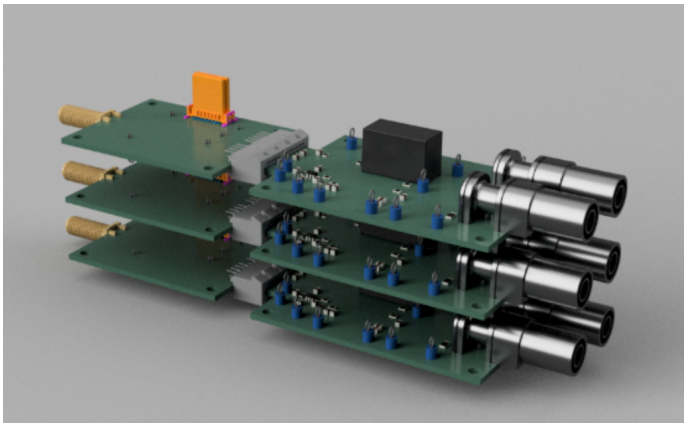

Figure 5: Three phase modular and isolated signal adaptation with inputs on the right side and analog output on the left side.

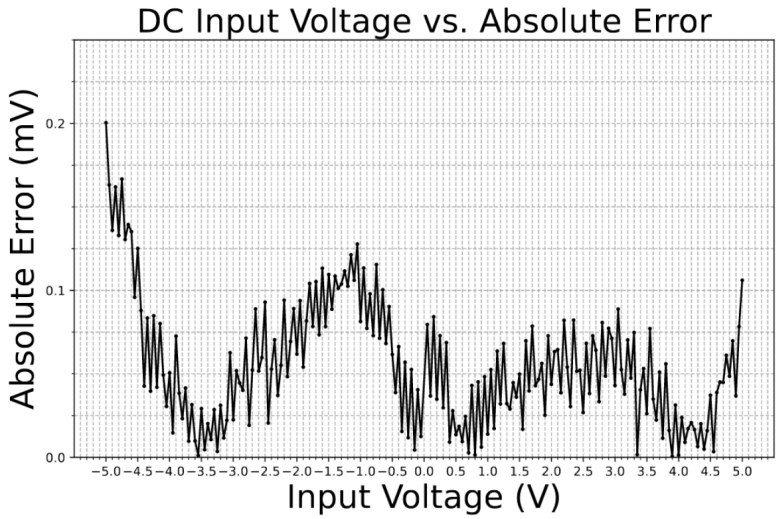

Figure 6: Absolute error for DC voltage measurement with an error of less than 200 µV within a full-scale input range of ±5 V.

3.1.3 5G Connectivity

The 5G edgePMU was evaluated with three 5G module and a custom-made M.2 carrier board, that provides the communication capabilities and is compatible with any device supporting USB 3. The first tests were done with a SIMCom SIM8200 [5]. A picture of this module is given in Figure 7 (left).

Since the 5G edgePMU is designed to be compatible with different module manufacturers and hardware versions, different modules were acquired and evaluated. A more detailed description is given in the following section. In addition to the evaluation of different modules, a custom carrier board was developed. The main goals are to prepare the hardware for a PPS output and provide a small form factor USB-C based board that is compatible with the different M.2 module modules. More details are given in the 5G M2 USB-C Carrier Board subsection.

3.1.4 5G Module evaluation

For the 5G edgePMU different M.2 based modules were evaluated in a 5G SA and 5G NSA environment. The main goal is to determine if all modules achieve the expected latency numbers and to understand if a specific module needs additional software to run properly. The modules are shown in Figure 7 and a list of the modules is given in Table 1. The Quectel RM520N-GL is the module with the lowest price and also the lowest measured latency. In addition, a RedCap module was acquired. These types of modules, with a reduced capability set, are lower cost but based on the data sheet, still meet the requirements for the energy vertical. Unfortunately, the modules are not yet supported by the local 5G SA network and need to be tested within the 5G-ICE network.

Figure 7: Different M2 module for 5G connectivity with onboard antenna connectors that are evaluated.

Name	Price	Notes	5G Relase	Downlink (to Modem)	Uplink (from Modem)	Capabilities
SIM8202G-M2	307€		R15	720 Mbit/s	100 Mbit/s	
SIM8200EA-M2	365€		R15	510 Mbit/s	105 Mbit/s	
Quectel RM520N-GL	303€	lowest latency	R16	780 Mbit/s	130 Mbit/s	
Quectel RG255C-GL	118€	not yet evaluated	R17			RedCap, URLLC

Table 1: List of evaluated 5G modules.

Most important for the use in power systems automation is latency. Therefore, latency depending on the package size was tested in a lab environment. For this test, a 5G SA network based on an Ericsson EP5G product was used. The module under test was the only active device in the network and the latency was measured to the network router, which is a pfSense firewall appliance. The results for different packet sizes are given in Figure 8. The Quectel RM520N-GL was found to provide a lower latency across all package sizes. It will thus be used in the further development and deployment of 5G edgePMUs until RedCap can be finally evaluated and RedCap mobile networks become widely available.

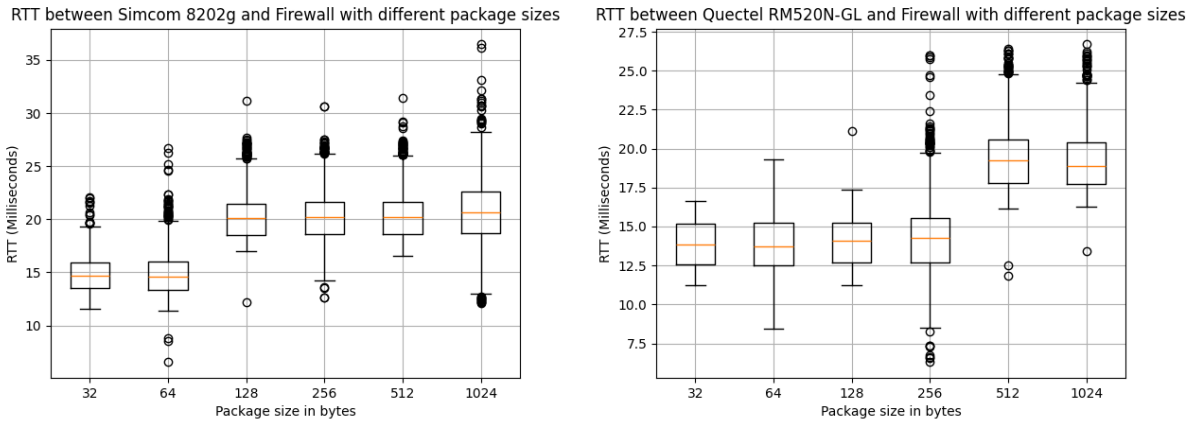

Figure 8:Latency evaluation of 5G Modules depending on package size.

The different modules were tested on the one hand with the EP5G 5G SA network but also at the robotics testbed with the 5G NSA network. In both networks connectivity was established and communication with our custom 5G carrier board was working.

All considered modules support the Qualcomm MSM Interface (QMI) [28] for sending commands and establishing the connection. Consequentially, this protocol is used during provisioning to setup a 5G

connection. The developed scripts, based directly on the standard Linux QMI implementation, allow for independence from the used network stack and therefore the general software platform. The Quectel RG255 Module required further intervention, as it is not directly supported by the Linux Kernel. Patched kernel drivers had to be procured from Quectel and integrated into a self compiled kernel. Further software challenges were encountered during evaluation in a 5G NSA context. While all Modules show no problem connecting to a 5G SA environment, a 5G NSA environment needs to be handled slightly different, and the module needs to be forced to connect to this specific network type.

RedCap

Besides the standard 5G modules also a RedCap module was acquired. Already the engineering samples are only a third of the price of a normal 5G module, and in addition the datasheets claim URLLC capabilities. It is planned to evaluate these modules at 5G-ICE on campus Melaten in Aachen, Germany. For testing the RedCap module, a software update of the 5G campus network is needed. Testing is planned to take place between the end of 2024 and early 2025.

3.1.5 5G M2 USB-C Carrier Board

To connect the M.2 modules to the edgePMU, a carrier board is needed to adapt the M.2 interface of the modules to a suitable USB Connection. Current products on the market regularly use a normal USB A 3.0 connector instead of the modern and recommended USB-C connector. Because the older USB A connector does not support the power requirements of most modules, an external power supply is often required. They also often come in physically big enclosures compared to the small module. The previously mentioned modules have potential support for 5G-based PPS signal generation to use in timing applications. These features are not yet implemented in software, but advertised by the module manufacturers and the hardware is capable of it. However, the existing carrier boards do not have support to connect to the PPS signal coming from the module. Without access to this signal from the 5G module, an additional GPS module is required for time synchronization of the edgePMU, increasing total cost. For these reasons, a new carrier board was designed to satisfy the requirements.

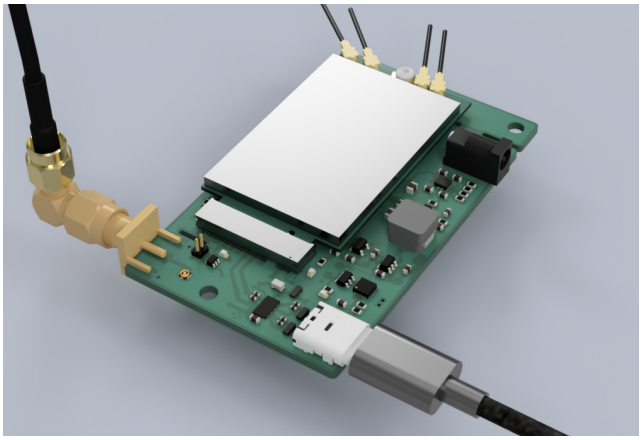

Figure 9: 3D rendering of the carrier board.

The design, made with open-source software KiCad 0, is 44 mm wide and 74 mm long (Figure 9). It supports multiple sizes of M.2 modules. This is important, as 5G modules with M.2 connection most commonly are either 40 mm or 52 mm in length. Data and power for the carrier board is provided using a USB-C connector. The connector supports the USB 3.2 Gen 2x1 with a maximum transfer speed of 10 Gbps. Due to the higher power-carrying capabilities of USB-C, all tested 5G modules did not require an external power supply. For better versatility, an external power connector is still present if the power capabilities of the host's USB port are limited.

The previously mentioned PPS signal is now available through a coaxial connector. Additionally, a regular SIM-card connector is available and a MFF2 embedded SIM can be used as the secondary SIM card. A MFF2 SIM card is integrated into a chip, which can be soldered to the Carrier PCB directly. Depending on the IoT provider, these can hold multiple carrier profiles and can allow to change the used carrier through an Over-the-Air update.[29] While PPS functionality could not yet be tested due to pending firmware modifications by the module vendors, 5G connectivity was tested successfully. Power Delivery either through the USB-C port or the power connector with the 5G modules attached works. The implementation of the reverse side plugging feature of USB-C is challenging and only works partially in the current hardware revision. Modifications were made to the design and future revisions can also make use of this feature. The carrier board with a mounted SIMCom module is shown in Figure 10.

Figure 10: The assembled carrier board with module attached.

3.1.6 Time Synchronization

Time synchronization is critical for every PMU device. Up to now, the synchronization for the 5G edgePMU has been achieved by sampling the PPS signal. This allows for a simple time synchronization that can be achieved based on software running in a non-real time operating system environment (Figure 11 top). The downside of this type of synchronization is that the precision is directly related to the sampling rate. For a sampling rate of 10 kSamples/s, a time precision of 100

µs can be achieved. Based on this approach, achieving a time precision of higher than 100 µs on a Raspberry Pi is challenging due to the high data throughput needed at higher sampling rates. Further errors are introduced by the clock precision of the actual sampling clock. The resulting error can be observed in Figure 12 (the orange phase estimation result).

A better solution is a hybrid approach for sampling. In that approach, the PPS signal is used as a time reference for a Phase Lock Loop (PLL). The PLL then generates a secondary PPS signal as well as a sampling clock for the acquisition device (Figure 11 bottom). This allows for much more precise timings without losing the easy-to-use USB interface and Raspberry Pi approach for the device.

Based on a development board for the chosen PLL chip, a first test with the 5G edgePMU was already done. A comparison is shown in Figure 12. For the direct PPS sampling approach (blue) the phase is jumping periodically, whereas in the case of the hybrid sampling, the phase is estimated much more precisely (orange). In addition, the measurements are compared with commercial PMUs. It can be seen that the commercial PMUs (yellow and purple) provide a phase estimation that is in line with the new PLL based estimations. A picture of the two commercial PMUs is given in Figure 13. Based on these findings and requirements, a custom circuit board has been designed that integrates the PLL chip and can easily be used with the 5G edgePMU.

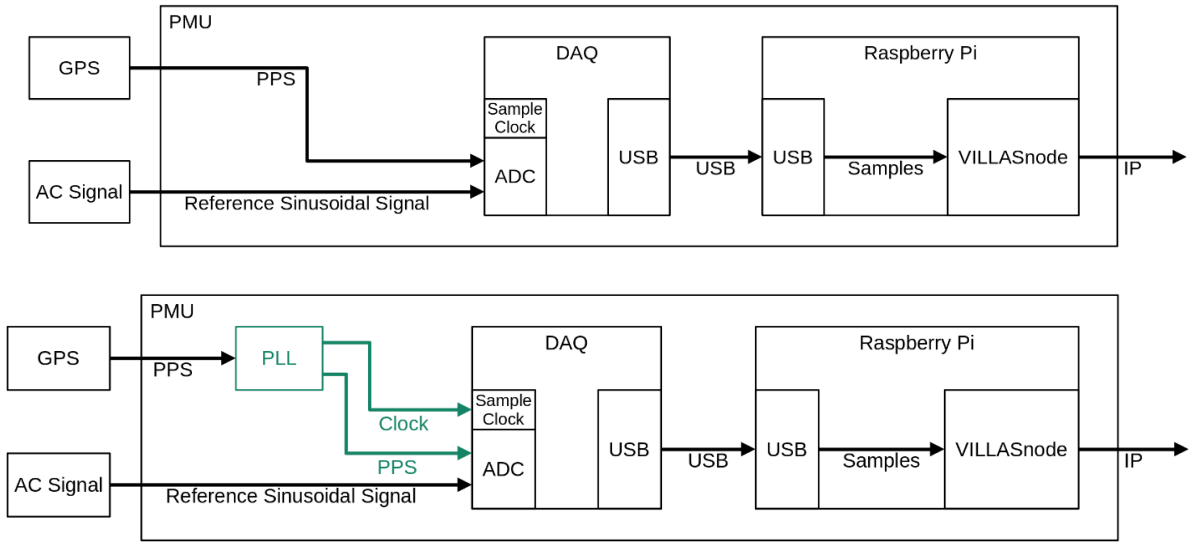

Figure 11: PPS based synchronization (Top) and Hybrid synchronization (Bottom), usable for a PMU device. In this case, this approach is adapted for the 5G edgePMU.

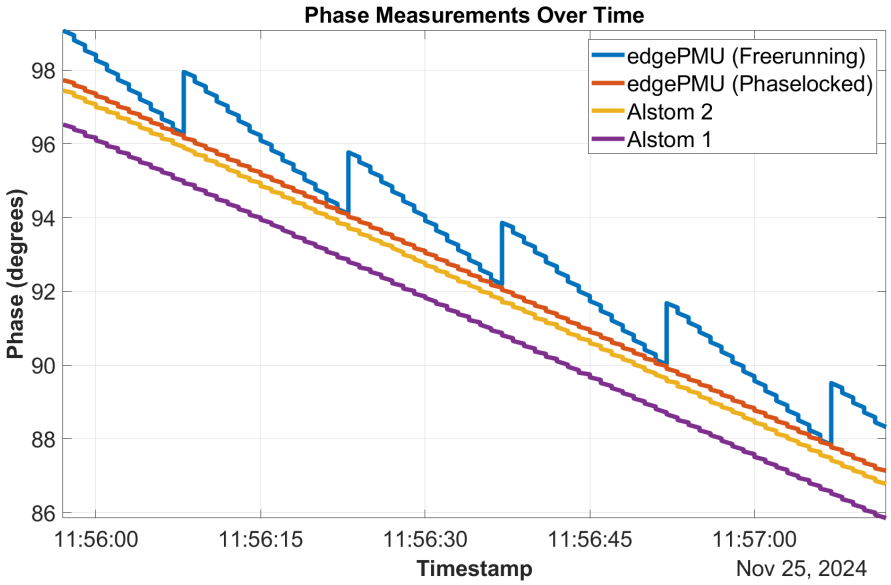

Figure 12: Comparison PPS synchronization (edegPMU Freerunning) vs. Hybrid time synchronization (edgePMU phaselocked) to commercial PMUs.

Figure 13: Alstom P847 commercial PMUs used in testing.

Development boards for PLLs from their manufacturers are often costly, have a big footprint, and include features which are not required for the 5G edgePMU. Due to this, the main design goals for a custom printed circuit board are to reduce size and cost while delivering good performance in terms of stability of the PLL. The offset and Jitter of the generated clock from the PLL, compared to the PPS signal from the GPS receiver, results in an error to the phase estimation, which makes a good synchronization of the PLL to the PPS signal important.

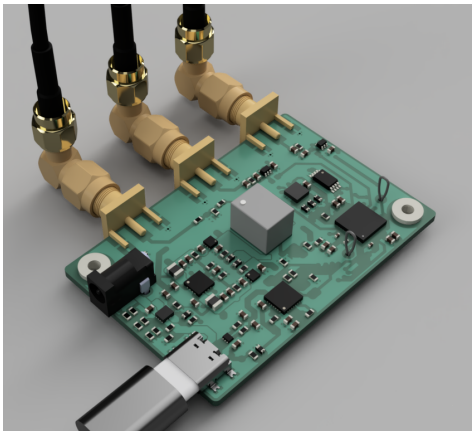

Figure 14: Rendering of the custom PLL circuit board.

The resulting design is a 40 mm wide and 60 mm long circuit board (Figure 14). The input signal and output clocks are available with SMA coaxial connectors. In total, two output clocks are available, one for the regenerated PPS signal and another for the sampling clock. Both are synchronous to another, which solves the sampling clock problems seen in “Freerunning” edgePMU (Figure 12). Connection to the edgePMU is established using a USB-C Connector. An intermediate chip is used by the edgePMU to control the PLL, allowing for fast deployment of the PCB. An alternative power connection is also available. The total BOM cost for a low quantity manufacturing order is around 150 €.

Figure 15: Performance measurement of generated PPS signal of the PLL. (Green: GPS PLL, Yellow: PLL PPS)

After assembly and configuration, the performance of the PLL was determined. The delay of the PLL’s PPS signal was measured to be 20 ns compared to the original GPS receiver’s PPS signal (Figure 15). This constant delay offset can be compensated for with the PLL. Besides the static offset, there is also clock jitter to be evaluated. Due to the nature of this error, it cannot be compensated for but

has a direct impact on the phase estimation performance. After 120 clock cycles, the maximum jitter is measured at 15 ns (Figure 16). This means that when measuring a 50 Hz sinusoidal signal, we have an uncertainty of 4.7 microrad.

One key measure for a PMU is the “total Vector Error”, as defined in the IEEE C37.118.1 [23] standard, which evaluates the accuracy of the phasor measurement. It quantifies the error of the estimated phasor to the real phasor and is dependent on the magnitude and phase estimation. The standard requires a maximum TVE of 1%. Assuming no magnitude error, as that is not dependent on the timing error induced by the PLL, we calculated the total vector error to be only 0.00047 %. This is multiple orders of magnitude better than the required 1 % max. TVE by the PMU standard, indicating that the PLL will not be a significant source of timing error for the resulting estimation.

Figure 16: Jitter measurement of the PLL after compensation using a high quality oscillator. (Green: GPS PPS, Yellow: PLL PPS, 120 captures)

This result can also be used as a reason to reduce the BOM cost of the custom board by using a lower-grade oscillator, while still maintaining good performance for the phase estimation task. In a test, one such lower-grade oscillator (temperature-compensated instead of temperature-controlled) has been used to achieve 100 ns jitter (Figure 17). This is equivalent to an uncertainty of 31.4 microrad when measuring a 50 Hz sinusoidal or a TVE of 0.00314 %, which is still well within the 1 % TVE target of the IEEE standard.

Figure 17: Jitter measurement of the PLL using a lower quality oscillator. (Green: GPS PPS, Yellow: PLL PPS, 120 captures

While the time until the phase-lock was achieved has increased, the price was greatly reduced. The previously used oscillator has a cost of 36,90 € to 29,15 €, depending on the quantity. The tested lower-grade oscillator has a cost of 10,57 € to 10,06 €, while also having a better availability. Power consumption is also reduced, as the low-cost oscillator does not heat up to maintain a steady temperature. More testing in performance of the oscillator is still needed.

3.2 Measurement Box (Meter-X)

For the deployment of the above-mentioned hardware in the construction testbed, a weatherproof enclosure is needed. Furthermore, the device should be built in such a way that no electrician is needed for deployment. Therefore, it was decided to plan an outdoor-capable measurement box that can be connected in line with the device under test. In the case of the construction site, that could, for example, be a crane. More information about the construction site is given in D5.1 [4]. A first mockup of the device is shown in Figure 18. The internal setup of the device is shown in Figure 20.

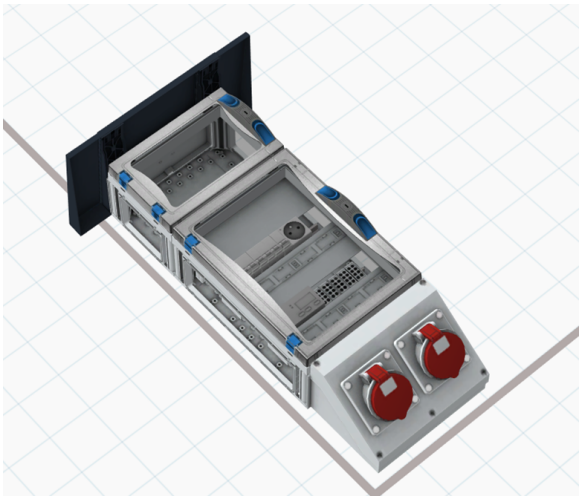

Figure 18: Mockup of measurement box for outdoor inline measurement of voltage current and energy.

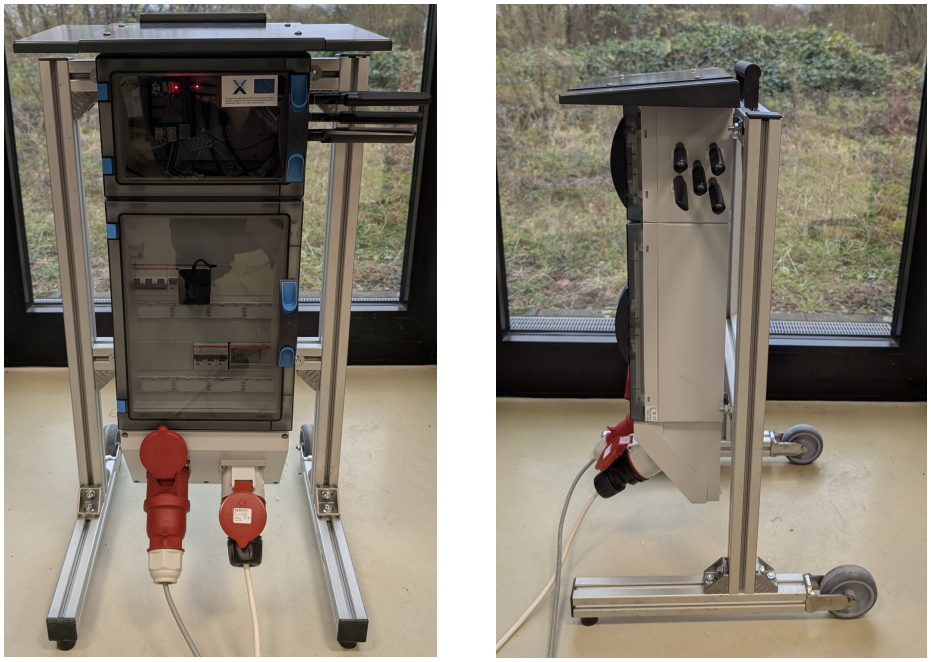

Figure 19: Measurement box for outdoor inline measurement of voltage current and energy, with adapters for one phase measurements.

The architecture of the measurement box comprises three-phase components like plugs, fuses and metering equipment, as well as low-voltage equipment like the Raspberry Pi, 5G module and ADC (Analog-to-Digital Converter), depending on the selected configuration. In Figure 20, all connections and components marked in blue have to be able to handle three-phase grid voltages and up to 63 A of current. The connection is done via outdoor-capable three-phase 32 A or 63 A connectors, referred to as CCE 32A in the figure. All black connections are low-voltage connections and are part of

communication or analog acquisition. The connections marked green are for current sensors and in yellow are digital connections.

The current version of the measurement box is built for a power of 22 kW (32A). The casing and Power meter are rated for powers up to 43.5 kW (63A), and the other components, like fuses and CEE connectors, are available in suitable variants. The box is also designed in a way that it can be supplied with a single phase for single phase loads. Therefore, it can be used with common household sockets (Schuko sockets) as well. The version depicted in Figure 19 is equipped with the needed adapters.

A first test on a construction site has shown that the measurement box is well usable, as it uses standard connectors which the workers are familiar with and can connect to the local 5G SA or 5G NSA network. For better transport, handles and wheels have been added. The Meter-X prototype is depicted in Figure 19. Currently, two Meter-X devices are available. One is used at the construction site and the second one is used at the robotics testbed. Further information on the preliminary deployment is given in section deliverable D3.5 [31].

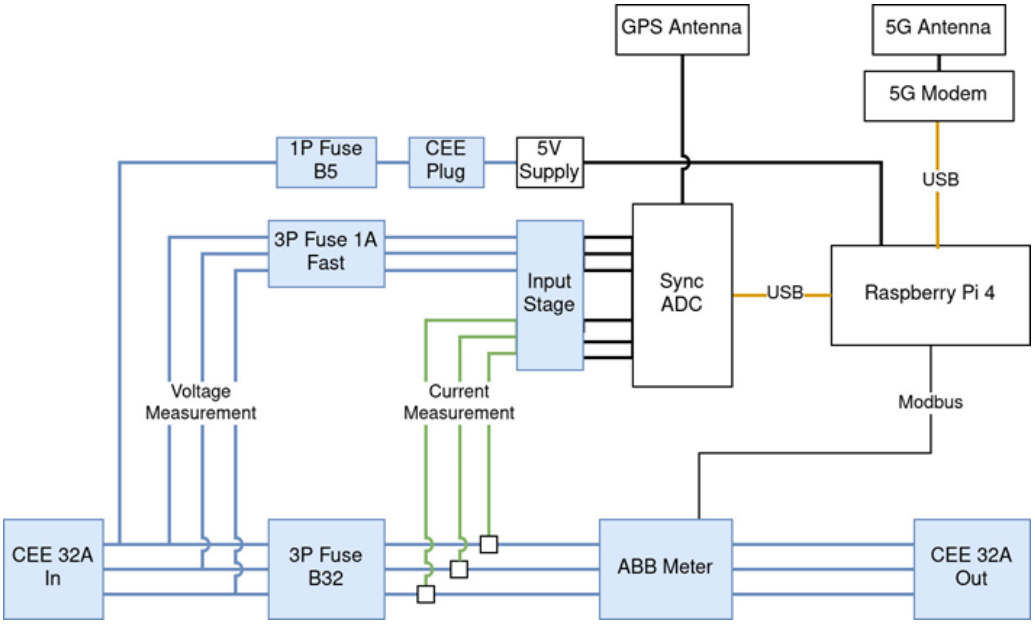

Figure 20: Architecture measurement box with the power components in blue, the low voltage components in black and the USB connections in yellow.

3.2.1 Metering Device

For energy metering, the Raspberry Pi is not well suited. Mainly, the Raspberry Pi is an operating system-based device. This means it can only handle metering while running, and metering can be interrupted due to software or hardware errors. Additionally, the Raspberry Pi would depend on a numerical integration of the energy. Especially for energy measurement, the impact of an integrative error is most important. Therefore, specialized metering hardware will be deployed. This hardware will, on the one hand, provide validated energy metering and, can, on the other hand, be used for validation of the high sampling rate acquisition running on the Raspberry Pi.

For the metering, two devices were validated. The Schneider iem3155 and the ABB B23 metering devices. Both devices were evaluated against an ZES Zimmer LMG671 power analyzer. The test setup can be seen in Figure 21. On the top, the two metering devices are connected with the ZES Zimmer reference device to a Chroma AC/DC load. The Chroma 63804 (bottom) is used to provide a defined power consumption. For the test, different power set points were validated. The range was limited by the used electronic load and started at 5 W up to 780 W. The resulting measurements can be seen in Figure 22. It can be seen that the Schneider metering device has a higher relative error in the lower power range. Furthermore, the absolute error is also higher over most of the tested range. Therefore, the ABB device is selected for the metering application.

Figure 21: Energy meter validation with the two metering devices at the top, the Zimmer power analyzer in the middle and the Chroma AC/DC load on the bottom.

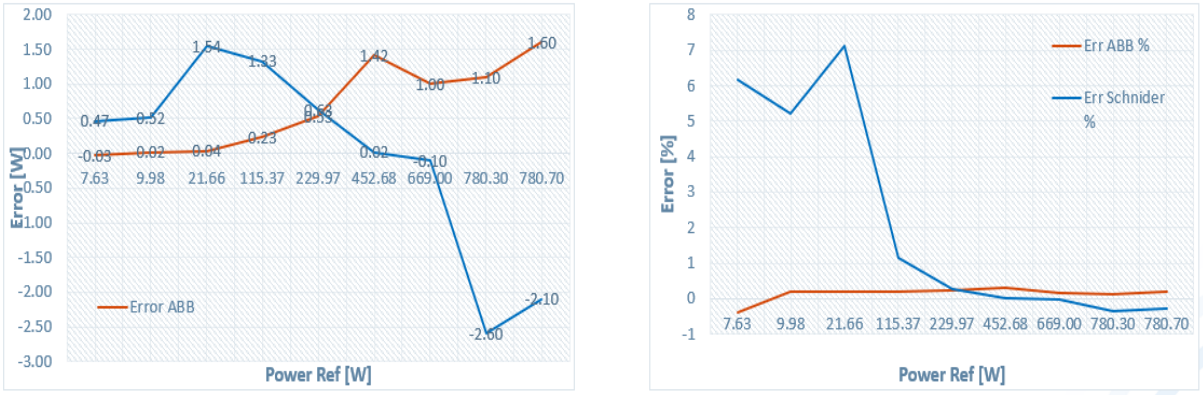

Figure 22: Measurement results for ABB and Schneider power meter with the absolute measurement error (left) and the relative measurement error (right).

3.3 Measurement Box (5G edgePMU)

The above-described Meter-X device is mainly targeted at mobile deployments and use cases where long term energy metering is of high interest. This is important for energy awareness use cases. In contrast to that, the edgePMU is mainly targeted at grid monitoring applications in stationary applications. A major difference is that with Meter-X an inline measurement is done, where in the case of grid monitoring the currents and voltages are often to high to do a direct measurement within the Meter-X device. Thus, the edgePMU allows for flexibly connecting sensors to analog inputs, and it is defined by the user what kind of transducer is connected to the analog channels. Currently, input voltages of ±10 V are supported and compatible voltage and current transducers can be selected.

Nevertheless, the edgePMU needs to be built for rough environments and easy deployment, which is why the whole unit is enclosed in a waterproof enclosure and all connectors are selected accordingly. The number of holes for cabling was minimized to reduce the ingress points for water, dust and insects. A mockup of the box can be seen in Figure 18. The box has a footprint of 25,4cm x 18cm x 9cm which is much smaller than the footprint of the Meter-X device. This is mainly since only low voltage components are part of the edgePMU device, whereas the Meter-X device also includes fuses and high current cabling and connectors.

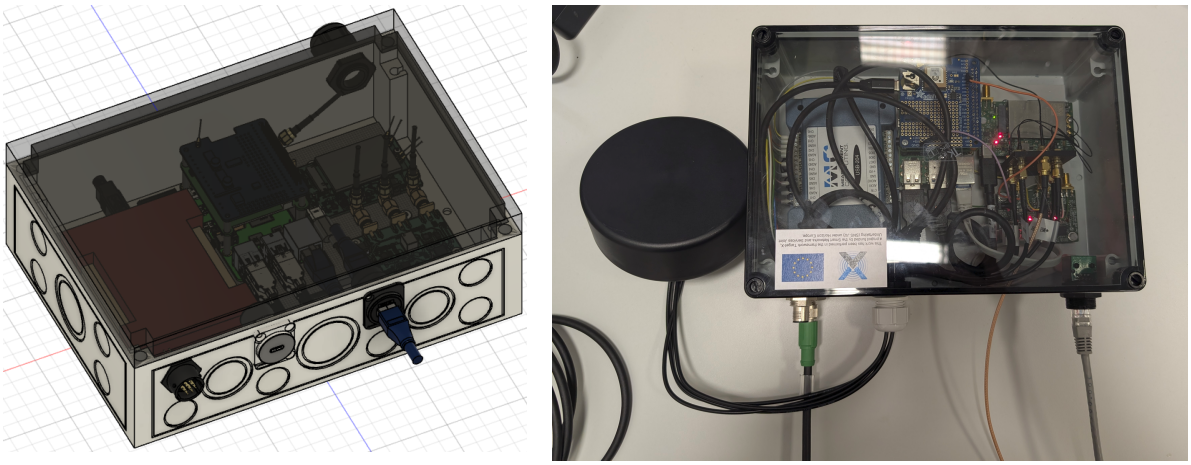

Figure 23: Digital render of 5G edgePMU (left) and functioning prototype (right).

Connections to the voltage and current transformers are made using a rugged and modular connector, connected to an ADC inside. The ADC is synchronized to GPS using the PLL custom circuit board. The same 5G carrier and Raspberry Pi setup can be used, allowing for multiple use cases of the built hardware.

3.4 VILLASframework

This section will introduce the different components used from the VILLASframework [10]. The VILLASframework is an open-source software set with different components that allows for data exchange between different data sources as well as the control of such nodes with a web-based UI.

3.4.1 VILLASnode

VILLASnode [6] is the core software component of the 5G edgePMU as depicted in the architecture overview in Figure 1. It allows for data translation based on so-called nodes and for data manipulation based on so-called hooks. For this project, both nodes and hooks are needed. An input node for communication with the USB data acquisition device is used and then after processing, an output node is needed to forward the data to an upstream service like a database or a secondary VILLASnode as shown in Figure 24.

The processing is done within a hook. An example for that could be Root Mean Square (RMS) calculation for voltages and currents or phasor estimation. For upstream communication, different types of nodes are available. The edgePMU currently allows for use of MQTT, IEC60870 [25] and IEC61850 [27] Generic Object Oriented Substation Event (GOOSE) and more. In addition, the implementation of the PMU protocol C37.118.2 [23] has started. This will allow for a PMU compliant communication and makes the edgePMU compatible with all standard compliant Supervisory Control and Data Acquisition (SCADA) and Phasor Data Concentrator systems (PDC).

For phasor estimation, an interpolated discrete Fourier Transform 0 algorithm is used. This algorithm was implemented within the edgeFLEX project [7], where the edgePMU was deployed in the distribution grid, to acquire measurements for voltage control service. The algorithm will now be extended with error compensation to allow for higher estimation precision in non-trivial cases like transients and off nominal frequency use cases.

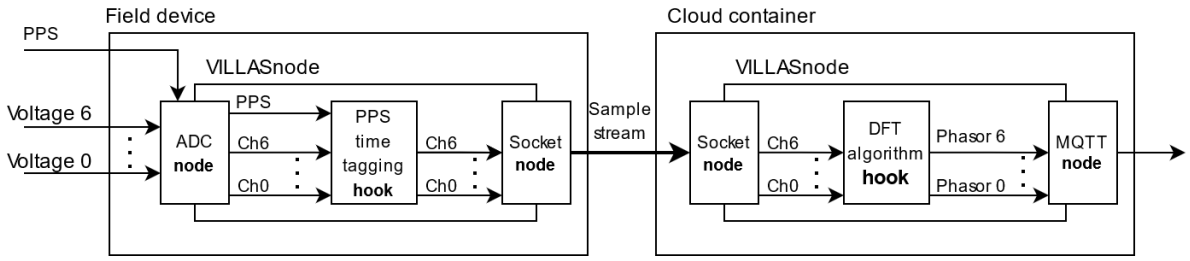

Figure 24: VILLASnode block diagram for the edgePMU with the field device on the left and the edge cloud service on the right.

3.4.2 VILLAScontroller

VILLASnode not only allows for a configuration file-based control, but can also be controlled with its own REpresentational State Transfer Application Programming Interface (REST API). This includes for example start/stop commands or configuration file selection. The tool used for this control is called VILLAScontroller [8]. This Python-based software utilizes a rabbitMQ broker [11] to exchange control data and acquire statistical information. Within this project, it is planned to utilize the VILLAScontroller to remotely control the 5G edgePMU. To do so, the software needs to be extended and integrated into the architecture.

3.4.3 VILLASweb

VILLASweb is a web-based control application for VILLASnode. It includes remote control and visualization features. Different VILLASnode instances can be controlled directly via VILLASweb. The interface utilizes a rabbitMQ interface. The VILLASnodes can transmit statistics to the VILLASweb application as well as samples.

4 Edge Infrastructure

This section will introduce the different components of the edge infrastructure. First, the provisioning system is explained. Then, the VPN and considered broker software as well as the network supervision are discussed. Finally, the storage infrastructure and visualization are described. The developed software stack is available on the RWTH Aachen University Gitlab page [32].

4.1 Provisioning Service

The provisioning system is responsible for installing and updating the various software packages and drivers that are running on the external measurement devices (such as the edgePMU). It further provides the operator with a method of registering new measurement devices in the device database.

Provisioning occurs in two steps. In a first step, an image is generated to be written to an edgePMU Raspberry Pi. This image contains all configurations and software needed for the second provisioning step.

In the second provisioning step, the edgePMU is configured for its measurement operation via Ansible [1]. The edgePMU periodically sends out a request to the Ansible provisioning server, which then constructs an Ansible update package, consisting of a playbook (see Section 4.1.3) and corresponding variables. This package is transmitted to the edgePMU and run locally with the Ansible-playbook command.

Figure 25 shows an overview of the main components that are involved in deploying the software configuration from the management server to the measurement devices. The image generation is discussed in Section 4.1.1. The edgePMU web frontend, which is currently not implemented, will function as a user interface for this step. The edgePMU database holds meta data about all clients, which are explained in Section 4.1.2. Finally, the Ansible provisioning server is detailed in Section 4.1.3.

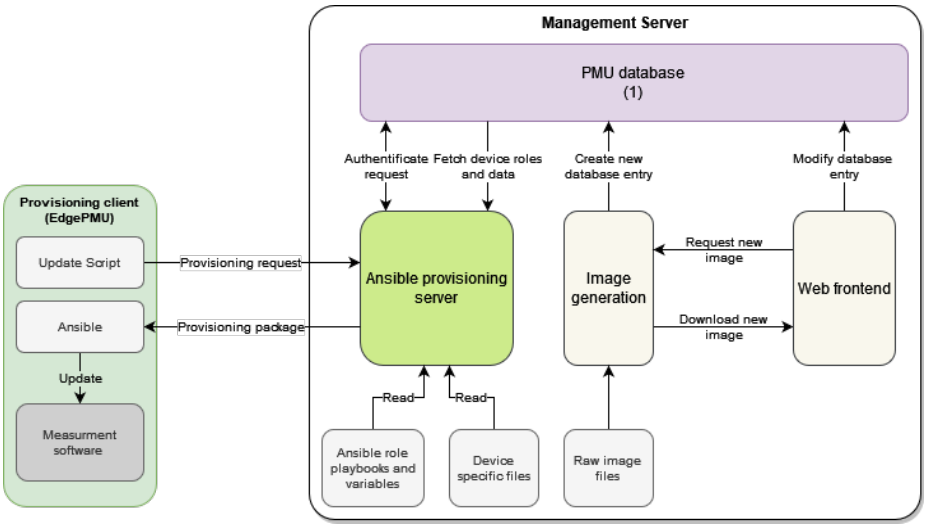

Figure 25: Provisioning Architecture with the edge cloud service for the database and provisioning on the right and the filed device on the left.

4.1.1 Image Generation

For the initial provisioning, a boot image for the edgePMU is created and written to an SD card. This image contains a pre-generated secret which is known only to the management server and the edgePMU. It further contains all software and configuration files to connect to the VPN (see Section 4.2) and perform further provisioning steps. The process is described in more detail in a publication [30]. After completing the initial provisioning, the edgePMU can provide basic functionality that is common to all devices, such as remote access via Secure Shell (SSH), a pre-defined hostname, supervision and Ansible.

4.1.2 Provisioning client

The provisioning client requests a new Ansible provisioning package at regular time intervals. After receiving the provisioning package (see Section 4.1.3), the provisioning client unzips the file. It then runs the Ansible playbook using the Ansible-playbook command. This command downloads or updates the software and executes commands for installation and configuration (see Figure 26).

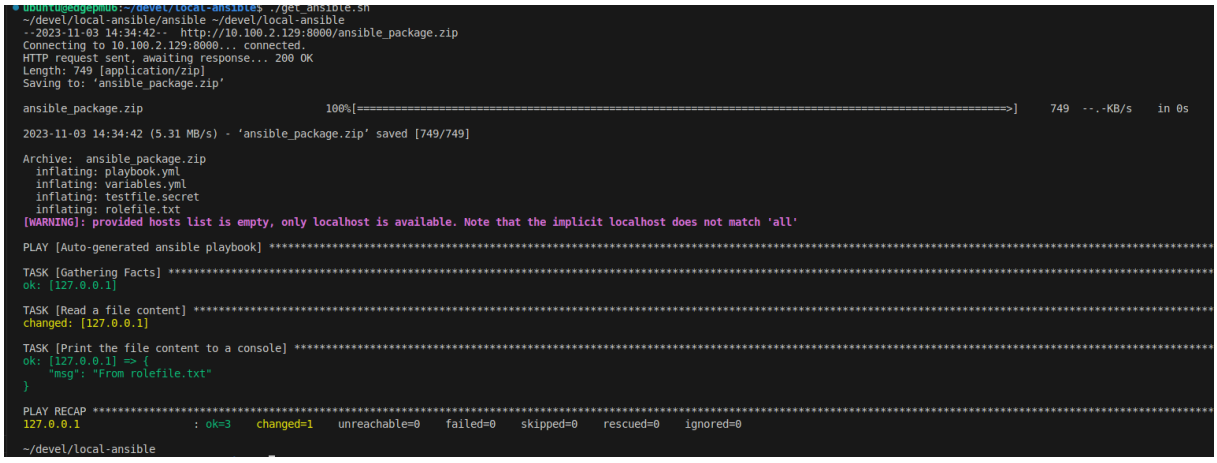

Figure 26: Execution of the Ansible playbook on a 5G edgePMU.

4.1.3 Ansible Provisioning Server

To update the configuration of the field device the Ansible provisioning server receives a request from the provisioning client. The provisioning client, which is an external untrusted source, connects via Hypertext Transfer Protocol Secure (HTTPS). This request contains the edgePMUs Universal Unique Identifier (UUID) and the pre-defined secret from initial provisioning. After authenticating the request with the database, the provisioning server fetches the relevant configuration variables and roles from the database to construct an Ansible provisioning package.

Each Ansible provisioning package is constructed from roles. These roles are a set of tasks and variables defined in .yml files on the management servers. To amend this, it is possible to attach a file, i.e. a configuration file, to a role which is then also transmitted to the edgePMU. Finally, to enable the configuration to be device-specific, variables can be defined on a per edgePMU level. Role-specific variables as well as edgePMU specific variables are then combined and transmitted within

the provisioning package.After gathering all files, the data is encrypted using the edgePMUs secret which was generated during initial provisioning. The encrypted data is then compressed and transmitted to the client.

```
INFO:     Application startup complete.
2023-11-03 14:40:20 root[8230] INFO Building Playbook for edgepmu6
2023-11-03 14:40:20 root[8230] INFO Fetching roles for edgepmu6
2023-11-03 14:40:20 root[8230] INFO Using roles: ['testrole']
2023-11-03 14:40:20 root[8230] INFO Writing variables to variables.yml
2023-11-03 14:40:20 root[8230] INFO Creating zip file at /home/meister/devel/ansible-http/temp/edgepmu6.zip
2023-11-03 14:40:21 root[8230] INFO Deleting working directory /home/meister/devel/ansible-http/temp/edgepmu6_2023-11-03T13:40:20.996587
INFO:     10.100.2.172:40122 - "GET /ansible_package.zip HTTP/1.1" 200 OK
```

Figure 27: Proof of concept screenshot from provisioning server, with an 5G edgePMU request.

4.2 Virtual Private Network

Using a Virtual Private Network (VPN) is a common technique to secure communication and allow for access to devices that are running in the field and cannot be reached via a public IPv4 or Ipv6 address. In the case of the 5G edgePMU, the VPN is used to connect the 5G edgePMU securely to the edge cloud infrastructure and allows for remote control of the field device. This supports manual intervention in case of software errors or for testing during deployment.

In the current version of the software stack, OpenVPN [12] is used as a VPN solution. OpenVPN is an established VPN solution and client for most operating systems, including Linux, Windows, macOS as well as Android and iOS. The main disadvantage is that OpenVPN is running as a user space application and thus is not the most performant solution. Furthermore, the configurations can be complex and are not fully downwards compatible. Therefore, it is planned to evaluate more recent VPN implementations like WireGuard 0 WireGuard stands out as it is implemented as a kernel module and thus much more performant than OpenVPNb [12]. Even though WireGuard is relativity new (initial release 2015), it is already well established and implementations for Linux, Windows or Android and iOS are available.

4.3 Broker (MQTT/Kafka)

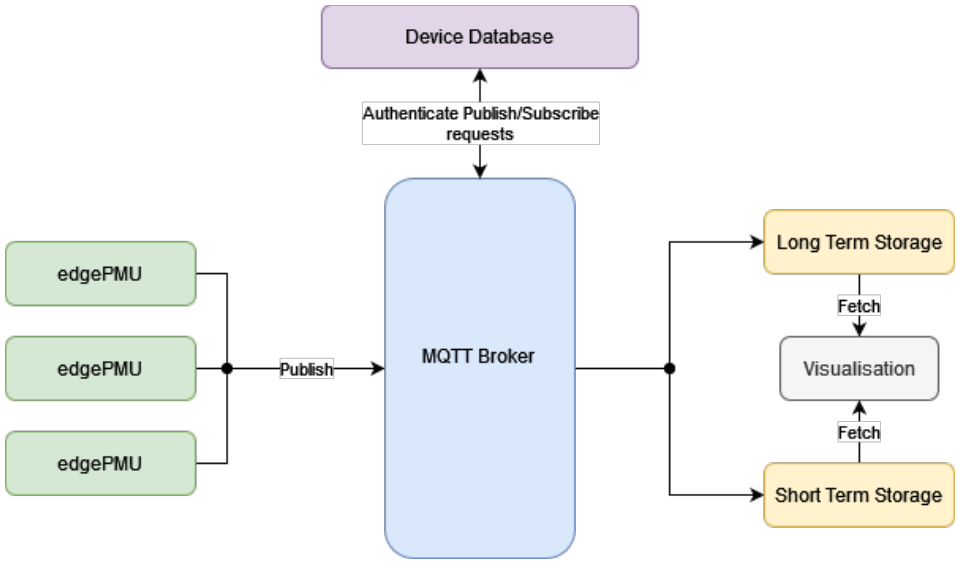

Figure 28: High-level data transmission architecture with multiple field devices (edgePMU) on the left and edge cloud services with broker, database and storage services on the right.

The broker is a server that controls the data exchange/distribution between PMUs and other devices. In our case of the 5G edgePMU, as shown in Figure 28, the broker is used to provide a secure data flow from the 5G edgePMUs to the data consumers, i.e. data flow to the long-term and short-term data storage and device database. For more information on the storage solution, see Section 4.5. Currently, both MQTT Broker and Kafka are considered to be used. Both design options will be examined in more detail in the following subsections.

4.3.1 Design option Broker 1: MQTT Broker

MQTT is a lightweight messaging protocol that is designed for limited bandwidth, high latency and low reliability networks [14], [15]. For real-time monitoring of edge devices like PMUs, it is important to have low-latency messaging by the protocol which can be fulfilled by MQTT. MQTT brokers are simple to set up and to use while handling numerous devices, which makes it a good choice for data transmission from edge devices to a storage or device database. However, MQTT brokers typically do not retain messages for an extended period, which may be a disadvantage if we need data history in some cases for analysis.

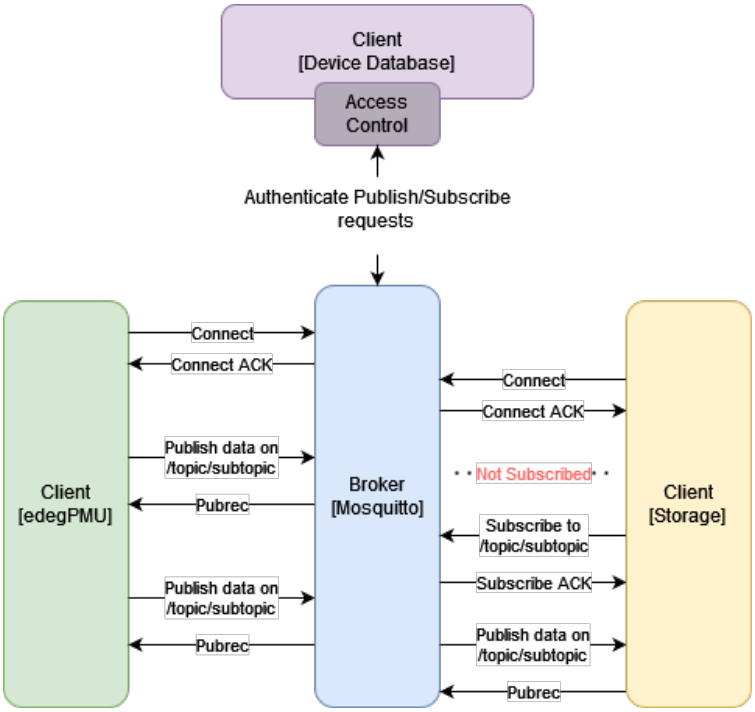

Figure 29: Overview of Mosquitto broker architecture for edgePMU data transmission case with the different states of communication.

In this case, the MQTT protocol can be used for communication between edgePMU devices and data storage or device database. MQTT brokers follow the Publish/Subscribe architecture, where the Publisher/Client publishes the messages on a topic, and the Subscriber/Client can receive those messages by subscribing to that topic [12].

In our case of 5G edgePMU data transmissions, Mosquitto is used as MQTT broker as shown in Figure 28. The edgePMU, storage and device database as clients create a connection with the Mosquitto Broker. After creating a connection, Mosquitto replies with a connection acknowledgement (Connect ACK) to each of them. Then, as per requirement, Mosquitto sends a data publishing request (Pubrec) to the Publisher edgePMU. In result, edgePMU publishes data to Mosquitto on a topic. On the other side, Subscribers like storage and device database as a client subscribe to the specific topic on Mosquitto. So, in return, the Subscribers get the subscription acknowledgment (Subscribe ACK) from Mosquitto. Further, as per need, Subscribers request for edgePMU data publishing (Pubrec) from Mosquitto. And as an outcome, Mosquitto publish the required data from edgePMU with that specific topic to the Subscribers.

4.3.2 Design option Broker 2: Kafka

Kafka is an open-source distributed data store and streaming platform, optimized for ingesting and processing large amounts of streaming data in real-time. Kafka 0 is based on a number of servers, called brokers, and clients communicating through a TCP network protocol. Kafka is designed as an event streaming platform, capable of handling high-throughput, fault-tolerant data streams. It is well-suited for scenarios where data must be ingested, processed, and retained over time. Key-value

messages, which come from many random processes, known as producers, are stored by Kafka. Here the data can be divided into different partitions within various topics. Kafka processes data in both publish-subscribe and log-based methods. PMUs can publish data to Kafka topics, and consumers can subscribe to these topics. Within Kafka, data can be retained for a specified duration. Kafka provides built-in data transformation and processing capabilities through Kafka Streams and Kafka Connect, allowing the manipulation of data as it flows through the system. However, Kafka can be more complex to set up and manage as compared to MQTT, and it typically requires more resources. For the time being, Option 1 with an MQTT broker is selected and used in the Meter-X and 5G edgePMU device. But still, Kafka is a possible solution in the future if the limits of MQTT are reached.

4.4 Infrastructure Supervision

Enabled by the distributed nature of the infrastructure, an efficient monitoring of the components is key to detecting faults and ensuring stable operation. This includes keeping track of the health status of the network and all connected clients. It further remains important to keep metadata about the connected clients, such as location, contact person or organization.

The Simple Network Management Protocol (SNMP) is a widely used protocol to supervise the state of multiple agents from a single central manager by polling. The manager periodically sends a request which is executed by the agent and a value is returned. The returned value is then logged by the manager in the Management Information Base (MIB) to be displayed in a graphical user interface. These requests can be amended by custom data, such as the agent's location.

For the supervision of the edge-cloud infrastructure and the deployed edgePMUs, Observium has been selected. Observium [19] is a mostly automatic SNMP data visualization tool, based on open-source software. Figure 30 shows the dashboard with an overview of the device locations. Features of this software include:

- Automated detection of available SNMP device features
- Visualization of historic data for the different features
- Alerting in case of errors (e.g. resource allocation maximum)
- Alerting in case of unreachable device
- Device grouping
- Geographical representation of devices

To add a device in the supervision infrastructure, the SNMP role can be assigned to the device during provisioning. This will install the SNMP daemon on the device and configure the device to respond as an agent to requests from Observium. To automatically register a device with Observium, further development is needed.

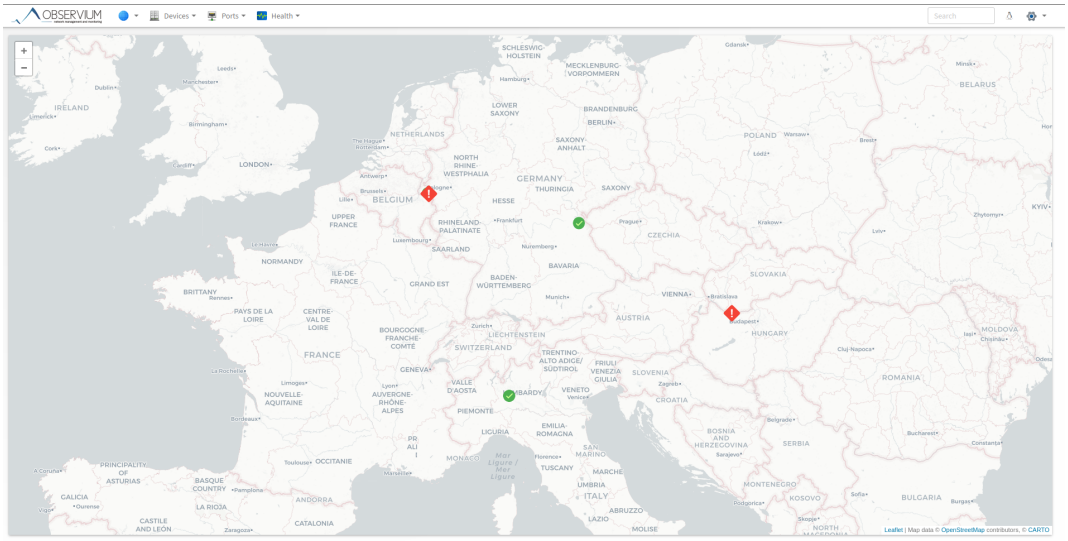

Figure 30: Observium Interface showing the different devices on a map of Europe.

4.4.1 Graphical representation of a network structure

Visualizing the information associated with a device is crucial for debugging in the lab. Data like IP address, port connection or location of a device can be accessed through the user interface of Observium. However, the data is scattered across different parts of the User Interface (UI) and categorized within different devices, which makes collecting information of a device cumbersome and time-consuming. Therefore, a UI is needed where relevant device data can be accessed all at once.

By using the data from Observium, a visual representation of the network structure can be realized, which results in a more user-friendly way of accessing device data. To achieve this, the data is processed and visualized with the help of the open-source JavaScript project d3.js [26] , which is a library specialized in data visualization. In this case, a force simulation of the network structure is used.

The network visualization enables users to retrieve information of devices from one spot (i.e. host name, domain, IP address, MAC address, port, location, VLAN). Furthermore, the graphical representation of the network includes the switches to which the devices are connected, allowing users to check the online status and connection of a device. These features are especially beneficial for lab testing and increase the user experience.

The network visualization is the bridge between the virtual and physical structure of a network, making it an indispensable tool that can be modified to fit the current need for device information. The visualization can be seen in Figure 31.

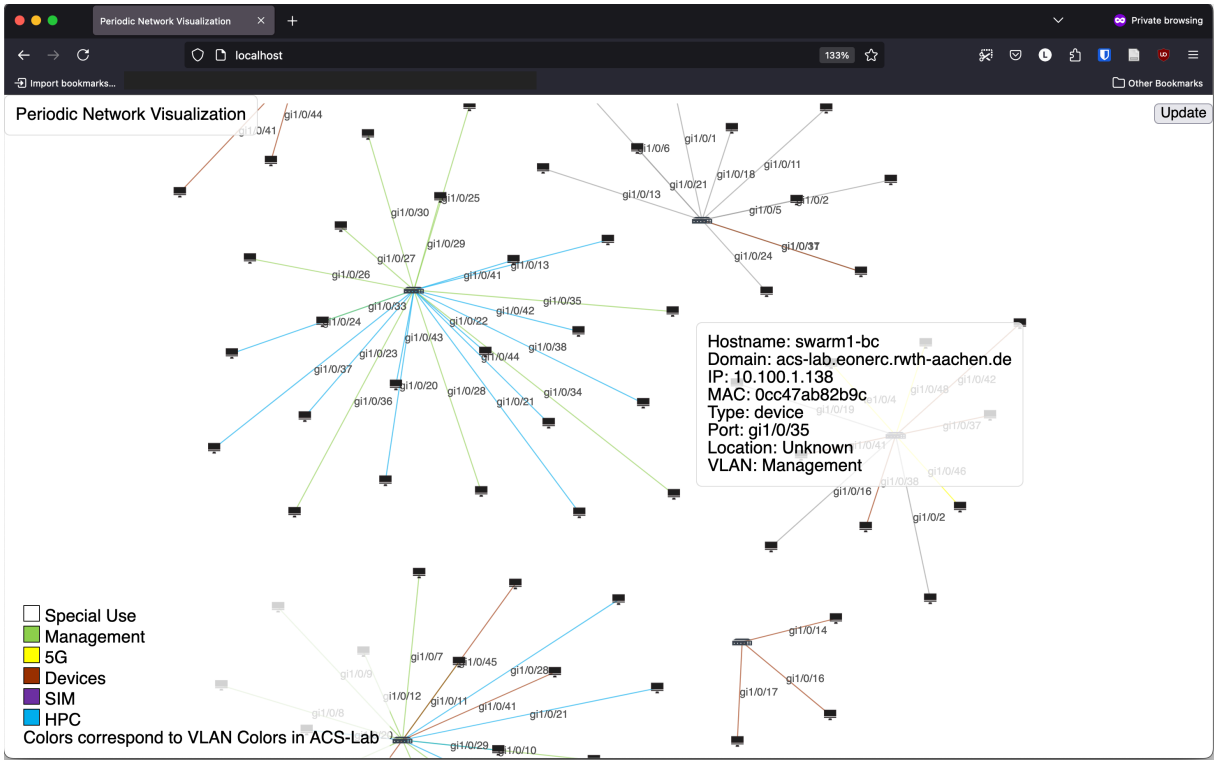

Figure 31: Network Visualization displaying device information of the ACS Lab.

4.5 Storage

The storage solution is a key element of making energy data available to the project partners but also allows for later reuse, for example in machine learning applications. To achieve the data storage, a storage server was procured. It was decided to buy a DELL PowerEdge R550 server equipped with 8x16 TB hard drives. As operating system TrueNAS Scale [20] was chosen. This enables a RAIDz2 0 with a usable storage capacity of 82 TB. It is expected that the actual storage capacity is higher since Zettabyte File System (ZFS) allows for native LZ4 compression, and it is expected that the energy data will allow for a high compression ratio. A screenshot of the installed TrueNAS Scale is given in Figure 32.

The above-mentioned storage server will be utilized for short as well as long-term storage. To allow for future extension, an S3 storage solution will be evaluated. This will allow for upscaling in case the available storage is not sufficient. The two topics of short-term and long-term storage will be further discussed in the following subsections.

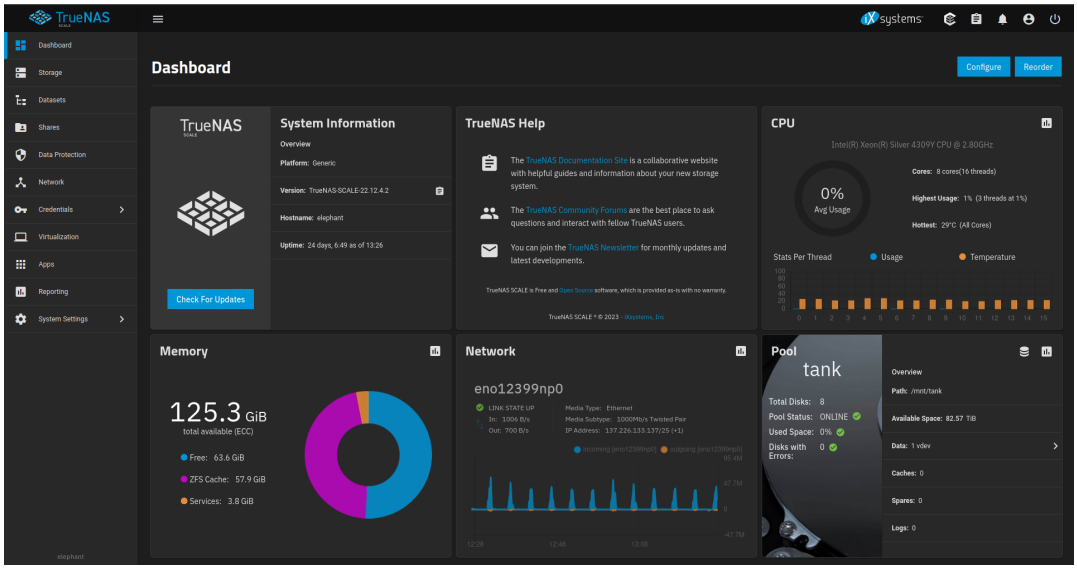

Figure 32: Screenshot of TrueNAS Scale installation on storage server.

4.5.1 Short-Term Storage

Short-term storage allows the user direct access to the data via a web User Interface (UI). It is planned to provide access to data for the past weeks or months. This is highly dependent on the chosen database infrastructure. For the current software stack, InfluxDB was utilized. It was seen that scalability for this approach is limited by the performance of InfluxDB. Therefore, it is planned to evaluate other time series databases for the short-term storage. The major requirements are fast access to data and the capability to automatically remove data after a certain time span.

4.5.2 Long-Term Storage

The long-term storage server, functions as an archive for previous measurement data as well as an easy access point for further processing of the acquired data. All operations of the long-term storage server take place decoupled from the management server's operations. For this reason, the long-term storage is located on an entirely different server than the main management tasks. While running, all configured broker topics are continuously logged into the file system. This has been demonstrated with a proof of concept, logging 100 topics, each with a data-rate of 100 kBps (see Figure 33), with further improvements possible.

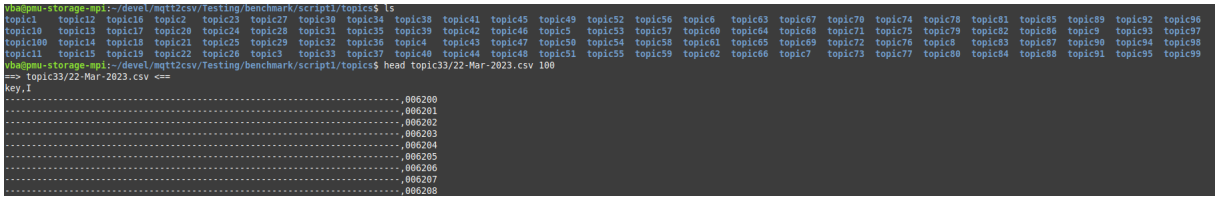

Figure 33: Long term storage proof of concept showing 100 topics and the running benchmarking application.

The active log files are rotated daily with old log files being compressed to reduce disk usage. Currently, two file formats are being explored for the final storage.

CSV: Comma Separated Values are a textual representation of a table format. Each column is separated by a separator character, usually a comma, and rows are separated by a new line. This simple representation makes CSV files easy to parse and human-readable. For this reason, CSV files require more disk space than a binary representation but also allow for easy processing.

HDF5: The Hierarchical Data Format has been developed to store large amounts of data in a file system like container. Values are stored in data frames in a binary format and accessed through a library. This decreases the file size drastically, while increasing the effort to extract the data for subsequent users. In addition, HDF5 can store metadata alongside the actual data frames.

For the current version of the platform, CSV is selected since it is very easy to share and evaluate.

4.6 Visualization

The visualization is the final puzzle piece that allows the user to interact with the proposed architecture. In the above-mentioned architecture, three different types of visualization are needed. The visualization of historical measurement data, configuration and control information and real-time measurement data. The three aspects will be discussed in the following paragraphs.

4.6.1 Historical Measurement Data

For visualizing historic data, Grafana is a well-established choice. For the current revision of the edgePMU architecture, Grafana is already used. In the updated deployment, this Grafana interface will be extended with an access policy to only allow for access to specific datasets. Additionally, it is planned to set up a template-like approach for the dashboards. This would allow for easy generation of dashboards for different field devices. An example of the Grafana UI is shown in Figure 34. This is an example already showing PMU measurements, but all plots had to be configured manually. This is not feasible when deploying multiple devices in the field.

Figure 34: Example Grafana UI showing PMU measurements, currently with a manually configured interface.

4.6.2 Visualization of Configuration and Control

This aspect of the visualization is not yet implemented but will provide a management interface for the databases that control the deployment of the edgePMU device. This includes setting the configuration for the field devices and registering new devices.

4.6.3 Visualization of Real-Time Measurements

Grafana is well suited for the visualization of data points from a Database. Due to the underlying concept, Grafana relies on loading all data points of a visualization every time the visualization is updated. This is especially inefficient when visualizing large amounts of real-time data. Furthermore, the main use case for Grafana is reading data, not sending an action to a client application. It is possible to enable such features, but the integration could raise issues. Due to these reasons, VILLASweb [9] is developed. VILLASweb. is an open-source project that is part of the VILLASframework [10]. It allows for web socket-based streaming of data and provides seamless integration with VILLASnode [6]. Thus, it is well suited for this architecture. Further development will be done to extend VILLASweb within this project.

5 Conclusions

In conclusion, this deliverable provides insights about the software and hardware architecture. Furthermore, the different aspects of the 5G edgePMU and Meter-X hardware development have been described, including the development of custom PCBs and the testing of different 5G modules. Since the hardware and software stack are still in a prototype state, further development and learnings from the field will be continuously integrated into the hardware and software stack. The developed hardware prototypes already show promising results, especially since the same Meter-X hardware can be used for three-phase measurements of a lifter (construction testbed) that consumes power in the kW range as well as for a single-phase robot arm (robotics testbed) that consumes less than 100 W. A more in-depth analysis will be done in the upcoming deliverable D3.6.

For the edge cloud software platform, the prototype software has been developed and described. The major components of the architecture are the automated deployment and management of the field devices, the storage and supervision infrastructure, and finally the visualization. The proposed architecture provides a holistic approach to the general question of measurement acquisition and archival. The software is provided on a public repository, which enables the extension by third parties and provides a continuously updated version for the deployment.

6 References

[1] Ansible, "Ansible is an open source IT automation engine" [Online]. Available: https://www.ansible.com/

[2] N. Alguacil, H. Herrero and C. Solares, "Observability Analysis and Restoration for Distribution Grids: A Robust Greedy Numerical Algorithm," in IEEE Transactions on Smart Grid, vol. 14, no. 6, pp. 4542-4552, Nov. 2023, doi: 10.1109/TSG.2023.3256488

[3] D3.1 Pilot Implementation Plan

[4] D5.1 Roadmap for the 5G 6G empowered deconstruction robotic platform

[5] SIM8200, "SIM8200EA-M2 product page" [Online]. Available: https://www.simcom.com/product/SIM8200EA_M2.html

[6] VILLASnode, "VILLASnode is a modular gateway for simulation data" [Online]. Available: https://villas.fein-aachen.org/docs/node/

[7] edgeFLEX "Managing future grids with the new VPP Concept" [Online]. Available: https://www.edgeflex-h2020.eu/

[8] VILLAScontroller, "VILLAScontroller provides a unified API for controlling simulation equipment from different vendors" [Online]. Available: https://villas.fein-aachen.org/docs/controller/

[9] VILLASweb, "A web interface for managing distributed simulations" [Online]. Available: https://fein-aachen.org/en/projects/villas-web/

[10] VILLASdocs, "Documentation of the VILLASframework" [Online]. Available: https://villas.fein-aachen.org/docs/

[11] RabbitMQ, " One broker to queue them all" [Online]. Available: https://www.rabbitmq.com/

[12] ObenVPN, "OpenVPN Community edition" [Online]. Available: https://openvpn.net/community/

[13] WireGuard, "WireGuard Fast, Modern, Secure VPN Tunnel" [Online]. Available: https://www.wireguard.com/

[14] Soni, Dipa, and Ashwin Makwana. "A survey on mqtt: a protocol of internet of things (iot)." International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017). Vol. 20. 2017.

[15] P. Brogan et al., "MQTT Architecture for Stream Analytics of PMU Data," 2021 32nd Irish Signals and Systems Conference (ISSC), Athlone, Ireland, 2021, pp. 1-7, doi: 10.1109/ISSC52156.2021.9467849.

[16] HiveMQ, " MQTT Publish/Subscribe Architecture" [Online]. Available: https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/

[17] Kafka, " Kafka 3.9 Documentation" [Online]. Available: https://kafka.apache.org/documentation/

[18] A. Hugo, B. Morin and K. Svantorp, "Bridging MQTT and Kafka to support C-ITS: a feasibility study," 2020 21st IEEE International Conference on Mobile Data Management (MDM), Versailles, France, 2020, pp. 371-376, doi: 10.1109/MDM48529.2020.00080.

[19] Observium, " Network monitoring with intuition" [Online]. Available: https://www.observium.org/

[20] Truenas, " Next Generation Unified Storage and Apps" [Online]. Available: https://www.truenas.com/truenas-community-edition/

[21] RaidZ, " RAIDZ Levels" [Online]. Available:

https://www.raidz-calculator.com/raidz-types-reference.aspx
[22] Grafana, " Grafana Open Source Software Docs" [Online]. Available: https://grafana.com/docs/grafana/latest/
[23] A. Derviškadić, P. Romano and M. Paolone, "Iterative-interpolated DFT for synchrophasor estimation in M-class compliant PMUs," 2017 IEEE Manchester PowerTech, Manchester, UK, 2017, pp. 1-6, doi: 10.1109/PTC.2017.7980868.
[24] C37.118, "IEEE Standard for Synchrophasor Measurements for Power Systems" [Online]. Available: https://standards.ieee.org/ieee/C37.118.1/4902/
[25] IEC 60870-5-104, "Telecontrol equipment and systems - Part 5: Transmission protocols - ALL PARTS" [Online]. Available: https://webstore.iec.ch/en/publication/3755
[26] D3 "The JavaScript library for bespoke data visualization" [Online]. Available: https://d3js.org/
[27] IEC61850 "IEC61850 Standard" [Online]. Available: https://iec61850.dvl.iec.ch/
[28] QMI, " Qualcomm Linux Modems by Quectel & Co" [Online]. Available: https://projects.osmocom.org/projects/quectel-modems/wiki/QMI
[29] eUICC, "What is an eUICC SIM card?" [Online]. Available: https://whereversim.de/euicc
[30] M. Pitz, F. Wege, N. Eiling, S. Vogel, V. Bareis and A. Monti, "Automated Deployment of Single-Board Computer Based Grid Measurement and Co-Simulation Equipment," 2024 Open Source Modelling and Simulation of Energy Systems (OSMSES), Vienna, Austria, 2024, pp. 1-6, doi: 10.1109/OSMSES62085.2024.10668996.
[31] D3.5 Pilot sites energy
[32] Open energy data platform "Giltab repository of the open energy data platform" [Online]. Available: https://git.rwth-aachen.de/acs/public/open-energy-data-platform