

10632**B5 PROTECTION AND AUTOMATION****PS1 - Practical Experiences and new Developments of Process Bus****Implementation of an IEC 61850 MMS interface for Centralized Protection and Control (CPC) virtualized platforms****Carlos ALBERO
CASTILLON****Miguel Angel
OLIVAN MONGE****Yasmina GALVE
PASTOR****Carlos RODRIGUEZ DEL
CASTILLO***CIRCE Research
Centre, Spain**CIRCE Research
Centre, Spain**CIRCE Research
Centre, Spain**ELEWIT, Spain*

calbero@fcirce.es

maolivan@fcirce.es

ygalve@fcirce.es

calrodriguez@elewit.ventures

SUMMARY

The digitalization of substations has become a prominent trend in the rapidly evolving technological world. This transformation has brought to a change in the way substations are run, with Centralized Protection and Control (CPC) platforms becoming more and more popular in place of conventional Intelligent Electronic Devices (IEDs). Out of all the CPC topologies, virtualized software-based platforms are becoming more and more popular due to their ability to offer innovative possibilities for the development and enhancement of protection algorithms. To address the evolving needs of this digital era, the EPICS (Edge Protection and Intelligent Control System) project has undertaken the development of an innovative solution. The EPICS solution leverages the concepts of edge computing, microservices, and virtualization to effectively decouple hardware from software in protection and control systems. The underlying architecture of EPICS is based on virtualized microservices powered by Docker containers. These containers run on a generic hardware platform, enabling the creation of a highly flexible, manageable, replicable, and scalable solution suitable for deployment in diverse application environments.

This work focuses on the development of an IEC 61850 communication interface, which plays a crucial role in standardizing the external communications of the EPICS platform on the station bus. Acting as an intermediary, this interface facilitates seamless data exchange between the EPICS platform's real-time database and external systems via the Manufacturing Message Specification (MMS) protocol. As a result, users can conveniently access real-time data via MMS, receive timely updates through MMS reports, and execute MMS commands that prompt the server to update the relevant data stored in the database referencing the IEC 61850 standard. The MMS interface also possesses the ability to communicate with the EPICS SV-APM (Sampled Values-Analog Processing Module), which is responsible for receiving and analysing Sampled Values streams from the process bus. Its primary function is to extract valuable

information from the Sampled Values flows as, e.g., phasors and grid frequency. Furthermore, the MMS interface stores the data received via GOOSE and publishes the corresponding values in its data model. By implementing this interface as a Minimum Viable Product (MVP), extensive testing and validation have been performed in a laboratory environment to ensure its reliability and effectiveness.

In addition to the development of the server-side interface, EPICS also encompasses the creation of an IEC 61850 client, which provides simultaneous connections to multiple servers, enabling efficient data querying and real-time updates from the data model. Furthermore, the client facilitates subscription and reception of reports, allowing users to send MMS commands and to visualize changes both through reports and directly on the underlying database. The versatility of the client implementation makes it a convenient tool for testing purposes, as it offers a generic and highly scalable solution.

The interfaces within EPICS have been developed using C/C++ programming languages, with the aid of the libiec61850 open-source library. Furthermore, the deployment of these interfaces has been streamlined through the utilization of Docker containers. Leveraging containerization technology not only enhances scalability but also provides a seamless configuration experience, making it easier to adapt and adjust the microservices architecture to meet specific requirements.

In conclusion, the EPICS solution represents an advancement in the field of control and protection systems for substations. By embracing the latest trends in software technology, it offers a highly adaptable, scalable, and easily configurable platform. The integration of virtualized microservices, Docker containers, and standardized communication interfaces offers significant advantages for managing the digital substation of the future. This powerful combination enables system operators to efficiently monitor, control, and protect their assets, ultimately leading to an enhanced service quality.

KEYWORDS

Virtualization, Digital-Substation, Edge-computing, IEC61850, MMS

INTRODUCTION

This article presents the work carried out in the context of the EPICS (Edge Protection and Intelligent Control System) project, focused on the virtualization and digitalization of protection systems in the electrical substation. The main point is an MMS interface that connects the EPICS platform to the outside world via the IEC 61850 standard. To understand how it works and its complexity, the complete platform will be first presented.

The power grid is now more dynamic than ever before, and newer tools are increasingly developed to manage the grid better. Renewed attention is required on protection and control strategies that build upon the available and emerging technologies such as centralized protection and control (CPC) technologies [1]. Within this new paradigm, EPICS fits in as a microservices-based virtualized software platform, designed to operate on a general-purpose edge server. The idea behind this platform is to have services or applications, running on a field server, executing algorithms, automatisms, and functions from the world of electrical substations and control centres, with the final objective of replacing the IEDs that are nowadays responsible for these tasks. It also has other features such as telecontrol functions, event logging and oscillography, and it allows local level 2 operation. It is a proof of concept that can be deployed in a substation position, and it has been successfully field-tested in a real substation as a Minimum Viable Product (MVP).

The platform runs on a Linux server, more specifically on a Rocky Linux 9.0, and has been implemented following the concepts of the IEC 61850 standard, such as the logical node to define the data model in the system and the SCL files for defining each EPICS scenario. All the services have been developed using the C++ programming language. A compiled language such as C++, is faster and more reliable than interpreted ones such as Python, and this makes it more adequate for critical scenarios as electrical substations. However, it should be noted that, thanks to the nature of the system architecture, services could be programmed in any language, and it would be possible to have services programmed in different languages running and interacting with each other.

Furthermore, the EPICS platform's versatility goes beyond language compatibility to support dynamic adaptability. Its virtualized microservices architecture allows for the dynamic addition and removal of services, promoting scalability and facilitating seamless future expansions and upgrades.

In essence, the EPICS project, with its MMS interface and standardization represents a cutting-edge leap in the modernization of electrical substations. In addition to addressing present operational difficulties, it positions itself as a scalable and adaptable solution for the changing demands on the power industry by embracing a microservices design and employing a virtualized environment. Successful field testing of the platform in a real substation has demonstrated its practical viability. This could change how we use advanced technologies in the essential area of electrical infrastructure.

VIRTUALIZATION

Virtualization significantly impacts digital substations by transforming their architecture and operational capabilities. Here are some key impacts of virtualization in digital substations that protection and control platforms like EPICS provide:

- **Decoupling of Hardware and Software:** Virtualization allows for the separation of hardware from software in protection and control systems. This decoupling means that specific functions are no longer tied to specialized hardware but can be implemented in software running on generic hardware platforms. This approach enhances the flexibility and adaptability of the system.
- **Use of Microservices and Docker Containers:** The EPICS project leverages microservices architecture encapsulated in Docker containers. These containers can run on generic hardware platforms, providing a highly flexible, manageable, replicable, and scalable solution. This architecture is suitable for deployment in various application environments, indicating a significant shift from traditional substation designs.
- **Enhanced Scalability and Manageability:** Virtualization, particularly using Docker containers and microservices, allows for easy scaling of substation functionalities. It facilitates the addition or removal of services, promoting scalability and ease of updates or upgrades. This scalable architecture is crucial for adapting to evolving technological and operational demands.
- **Standardized Communication Interfaces:** In the EPICS platform, the development of an IEC 61850 Manufacturing Message Specification (MMS) interface standardizes external communications. This interface enables seamless data exchange between the platform's real-time database and external systems, ensuring consistency and reliability in communications.
- **Cost-Effectiveness and Innovation:** By using generic hardware and software-based solutions, virtualization can reduce costs associated with specialized hardware. Additionally, it opens opportunities for innovation, as new functionalities can be developed and deployed more rapidly and cost-effectively.
- **Enhanced Security and Flexibility:** Virtualization can enhance the security of digital substations by isolating different functions and services. This isolation can prevent the spread of potential security breaches. Furthermore, it offers greater flexibility in system configuration and adaptation to changing requirements.

In summary, virtualization transforms digital substations into more flexible, scalable, and efficient systems, capable of adapting to the evolving needs of modern electrical grids. This transformation is a key factor in the ongoing modernization of the electrical infrastructure.

ARCHITECTURE

The platform architecture follows the microservices concept thanks to the containerization software Docker, a platform for developing, shipping, and running applications in containers. Containers are lightweight, standalone, and executable software packages that include everything needed to run a piece of software, including the code, runtime libraries, and system tools. It simplifies the deployment process, enhances scalability, and promotes a microservices architecture, making it easier to manage and scale complex applications. Docker uses containerization technology to isolate applications, ensuring they run consistently regardless of the underlying infrastructure.

One big advantage of Docker is that it facilitates independence from specific hardware and algorithm vendors, allowing to open the platform for the publication of services by third parties. This not only enriches the ecosystem but also enhances collaboration and innovation within the broader energy landscape.

Additionally, Docker and the microservice-based architecture contribute to maintaining a high standard of quality of service, particularly in terms of keeping constrained both delays and latency. Two studies have been published on this subject testing EPICS scenarios.

The first one [2] examines the deployment and performance of a software-defined approach to analogue processing within digital substations inside the EPICS platform, focusing on the crucial aspects of signal processing for control and protection functionalities. Leveraging a comprehensive test framework, using a Real-Time Digital Simulator (RTDS), the research measures the performance of the Sampled Values Analogue Processing Module (SV-APM), demonstrating its ability to meet stringent latency requirements. The documented results, with more than 99% of processing tasks executed within 250 μ s threshold and a maximum latency of 1.1ms, significantly below the IEC 61850-5 standard's stipulated 3ms, underscore the viability and efficiency of software-based centralized protection and control systems.

The second study [3] covers the implementation of several key protection microservices, including two and three-terminal line differential protection and distance protection, and the SV-APM previously mentioned. Extensive testing (1915 functional and performance tests) was carried out using the RTDS in a Hardware in the Loop (HiL) configuration to evaluate the performance and reliability of these microservices. The tests covered various scenarios, including steady-state and fault conditions, to assess the platform's response times and the correct operation of the protection functions. The line differential protection microservice, for example, was tested across a range of fault types, locations, and resistances, demonstrating its ability to operate correctly with tripping times of less than 30 ms, which is within the expected range for differential protections. Similarly, the distance protection microservice underwent a comprehensive test battery, showing reliable operation with an average tripping time of 35.71 ms, aligning with the standards for distance protections.

Although further testing is needed as new functionalities and applications are added, the positive outcomes from these studies, such as minimal latency (worst-case latency of 1 ms) and swift tripping times, demonstrate the platform's capability to perform at the same level as traditional protection relays, validating the architectural design options of the EPICS platform, as well as its functionality and effectiveness.

To ensure the persistence of data in the system, the Redis database engine has been chosen. Redis is an open-source, in-memory data structure store that can be used as a cache, message broker and key-value database. In this case it has been implemented as a key-value database, using the IEC 61850 reference as the key, and a custom structure formed by the data, the quality, and the timestamp for the value. One of its powerful features is the support for the publisher/subscriber model as it can be observed in Figure 1. Redis channel scheme

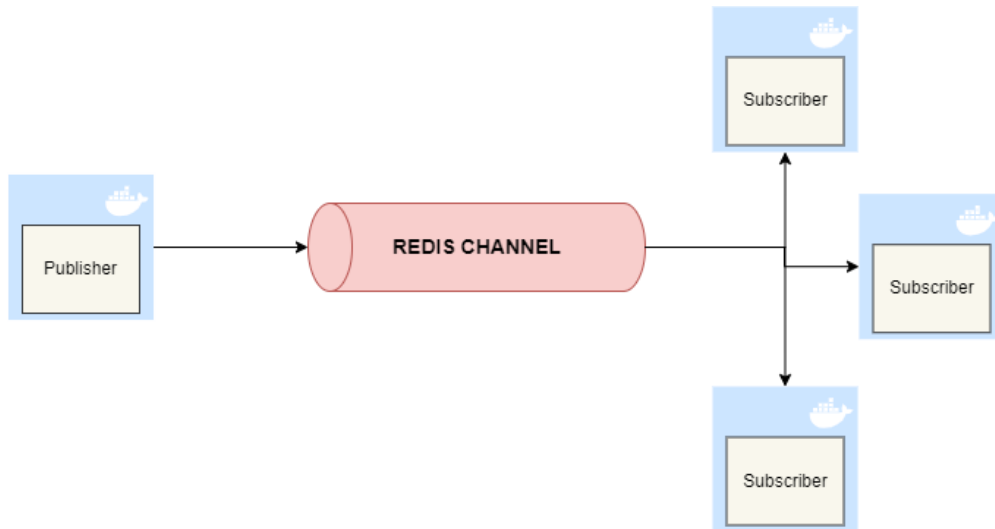


Figure 1. Redis channel scheme

In Redis, clients have the capability to subscribe to specific channels and consequently receive messages that are disseminated to these channels, enabling a decoupled communication mechanism. This model facilitates real-time communication between different components of an application. It is lightweight, fast, and scalable, making it suitable for scenarios where real-time communication or event notification is essential, such as our platform inside electrical substations.

Figure 2. EPICS platform architecture shows the architecture of the whole EPICS platform where many different services can be seen running inside Docker containers. Not all services have been drawn because there are too many, but this scheme is enough to get a clear idea of how the ecosystem has been built.

The bottom of the image shows elements of the electrical substation such as the station bus and the process bus. The two services above (i61SVs and i61GOOSE) act as ETL (Extract-Transform-Load) services. They are responsible for obtaining data from the buses, processing and transforming them to the format specified in the platform and storing them in the database. The data are processed by the Analog Processing Module (APM) to extract information such as phasors and grid frequency. The output APM data and GOOSE data are managed by a low latency microservice named BDTR in order to distribute the information across the EPICS platform.

At the top of the figure the control and protection services are displayed. Only three of them are represented but there exist additional ones. The platform is also open to third party manufacturers to easily add their own services. This gives the platform great scalability and flexibility. These services are fed with the data that has been introduced into the system by the ETL services, and they use them to perform their calculations and algorithms to generate the outputs.

In the centre of the picture, the Redis database service can be observed, which can be considered as the heart of the platform. This service is the reflection of the system's data model in memory and allows to have a real-time data flow throughout the entire ecosystem. It is responsible for keeping the in-memory data model up to date and available for real-time update using channels. The MMS interface functions as a translator between the data stored in the database, which has a specific format, and the MMS IEC 61850 communication protocol.

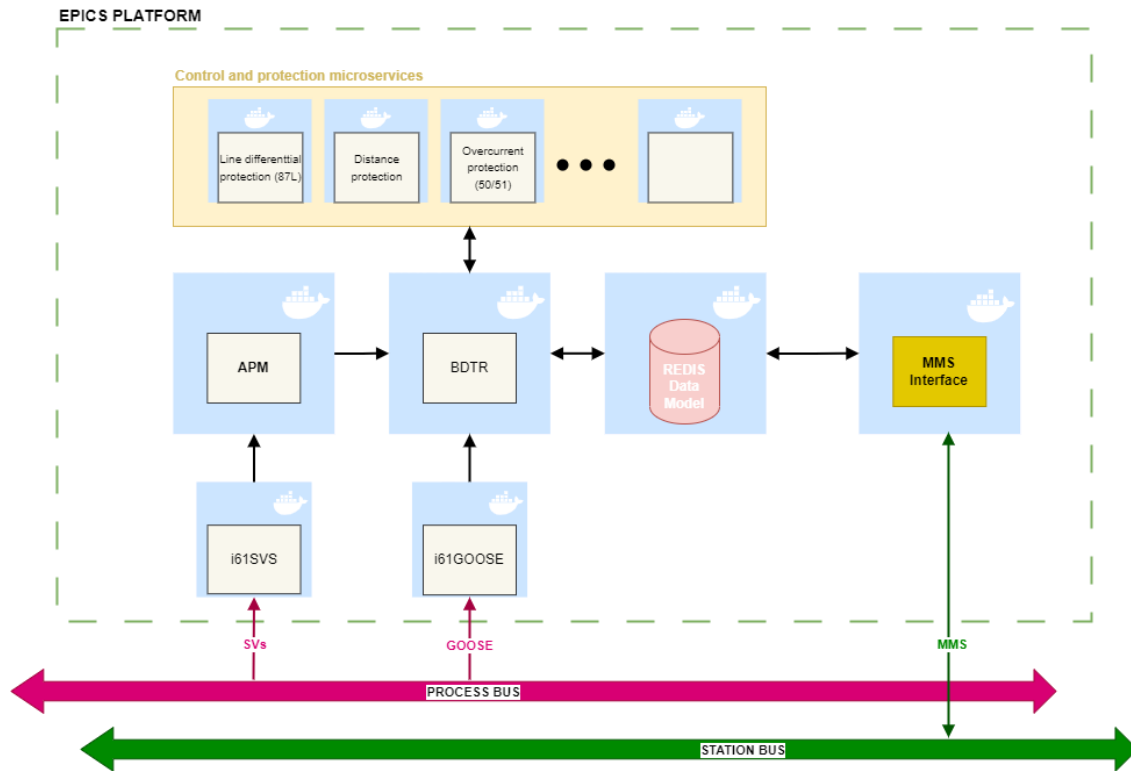


Figure 2. EPICS platform architecture

All the services represented in the figure run independently and in parallel, each one inside a different Docker container. They share the same internal network, so the connection between them does not require any additional dependency. To easily deploy and stop the whole stack of services, the orchestration tool Docker Compose is used. It only needs a configuration file (.yaml) where the services with their configuration and dependencies are specified.

MMS INTERFACE

The MMS interface service is employed to expose the live data model to elements outside the platform. It uses MMS for communication and data handling, it integrates with Redis for real-time data processing, and ensures security and file handling capabilities. Figure 3. MMS Interface scheme shows the interfaces of the MMS server.

At startup, it connects to the Redis database and updates its data model with the data retrieved from it. After that, it subscribes to a channel to keep up with incoming changes to the model. That is how the interface can maintain the data model updated in real-time. This way anyone can connect to the interface via MMS protocol to view the data model and subscribe to the MMS reports to be notified when a change has occurred.

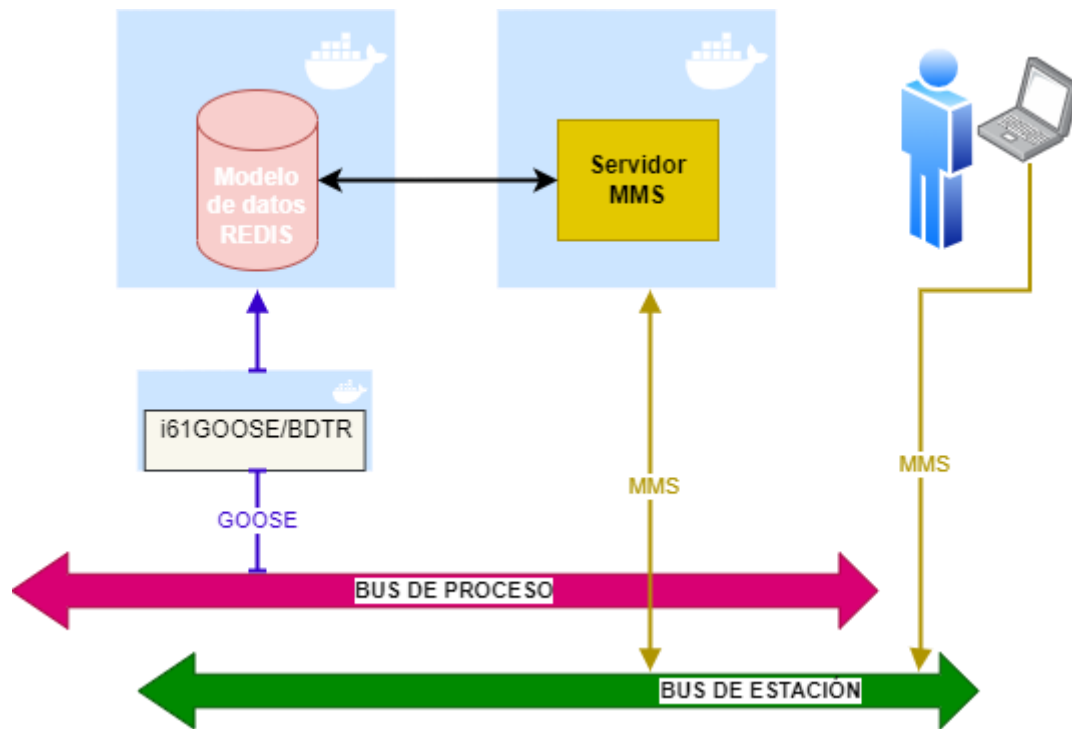


Figure 3. MMS Interface scheme

This service has been developed as a standalone service inside a docker container programmed with the C++ programming language, using the open-source library libiec61850 written in C. This library provides an implementation of IEC 61850 on top of the MMS (Manufacturing Message Specification) protocol in standard C. It is designed according to edition 2 of the standard series and allows the use of TLS for encryption of the data sent over MMS and ACSE authentication for the establishment, termination, and control of associations between the two communication parties, as mandated by the IEC 62351-4 standards. The current scope of this project, being a proof of concept, does not use the implementation of these security measure. However, it could be implemented in a next stage of the project.

The architecture and a summary of how the MMS Interface has been designed is presented in the following paragraphs.

It operates by using a configuration file, generated from a .cid one, to maintain an in-memory data model consistent with the platform's database. This data model is managed by a *Server* class, which holds the model and contains the functions needed for its maintenance and update.

Additionally, there is a *ControlRedis* object within the system. This object is crucial for managing the server's connection to the database. It subscribes to a specified channel and retrieves data from it. This subscription process is executed in a dedicated processing thread. Meanwhile, the main thread of the server is engaged in a continuous loop, actively monitoring

for any changes occurring within the server. The classes of the service can be observed in the UML diagram in Figure 4. MMS Server UML DiagramFigure 4.

A key feature of this server is the ability to handle multiple client connections simultaneously. This capability has been tested with various commercial clients, ensuring a robust and versatile connectivity.

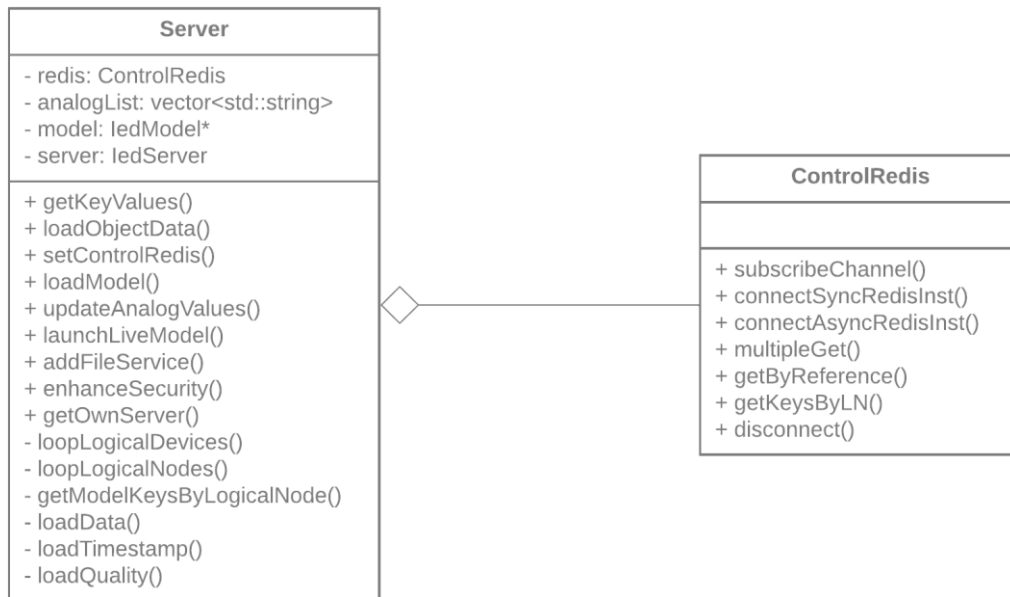


Figure 4. MMS Server UML Diagram

Connection between different commercial clients and our server has been tested, as shown in Figure 5, which shows the successful connection between the platform and a simulated IED. The result is that we have the same information and ways of operating as if we were connecting to a normal IEC 61850 device (IED). The whole data model, with all logical nodes, logical devices, data objects and data attributes, can be observed. We have the possibility to subscribe to the defined reports and see how they change. Finally, the server is also prepared to operate with commands in case the system supports this operation.

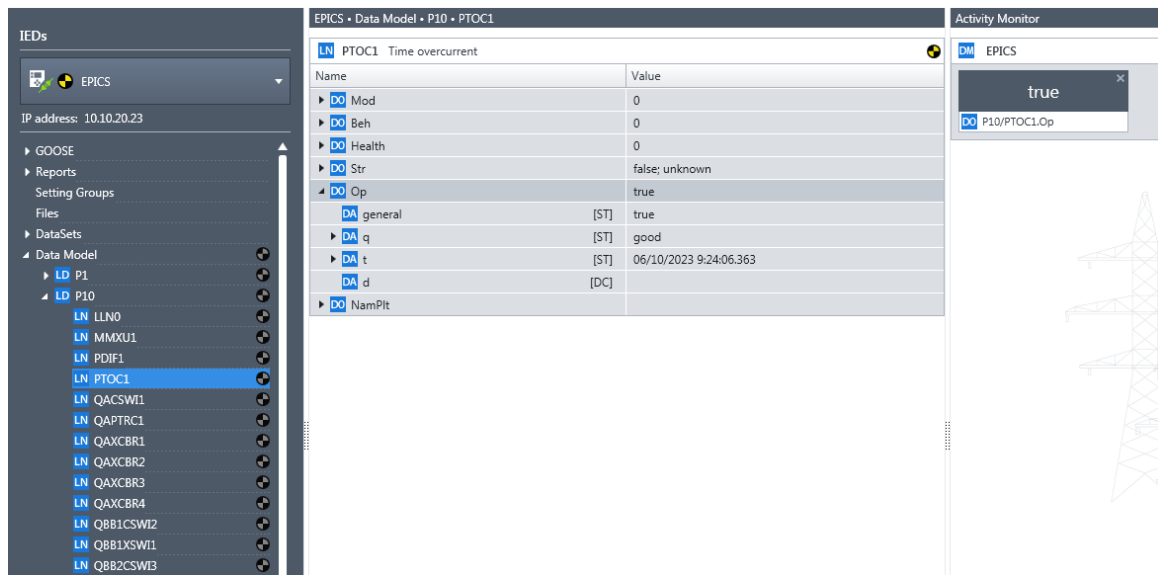


Figure 5. MMS Server connection with an external client

MMS CLIENT

As a necessity to test the entire EPICS infrastructure, an MMS client has been developed, which has become a powerful tool within the project. It allows to execute commands, query the data model via MMS, modify values externally via MMS, subscribe to reports and to connect to multiple IEDs or MMS servers simultaneously. It is developed in C++ using also the open-source C library libiec61850 in another standalone Docker container.

The potential of this client is that it allows to import data from other MMS servers or IEDs from the outside to the EPICS platform. In other words, it is a way to synchronize other IEDs in our model.

The client starts with a configuration file coded in xml format like the following one:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<connections>
  <connection inst="1" hostname="10.10.20.23" port="107">
    <ied name="ICMA12" cid="ICMA12.cid" />
    <rcb rcbRef="ICMA12CTRL/LLN0.BR.BR_ICMA12_CCS01"
dsRef="ICMA12CTRL/LLN0.ICMA12_CCS" />
  </connection>
  <connection inst="2" hostname="10.10.20.23" port="102">
    <ied name="ICMB12" cid="ICMB12.cid" />
    <rcb rcbRef="ICMB12Application/LLN0.BR.BR_ICMB12_CCS01"
dsRef="ICMB12Application/LLN0.ICMB12_CCS" />
  </connection>
</connections>

<bdtr hostname="10.10.20.23" port="6379" channel="EPICSCOMMAND" />
```

From this configuration file, the program obtains the necessary data to start. For each connection indicated, it tries to connect to the specified IP address and port in a different thread. It performs a parse of the indicated .cid file to obtain the structure of the system data model, following the hierarchy defined in the IEC 61850 standard. The classes in charge of this are those represented in the UML diagram in Figure 6.

This class structure starts from the IED label of the model and goes down through all levels (Logical Device, Logical Node, Data Object and Data Attribute), performing a deserialization and forming the in-memory data model. This allows to make queries to the model maintained in the Redis database, to subscribe to MMS reports and to visualize the changes in the model in real time.

To test the client, tests were carried out such as receiving changes via report message and verifying that the same value has been received from the database through the Redis channel. The sending of commands and the query of model data were also tested with satisfactory results.

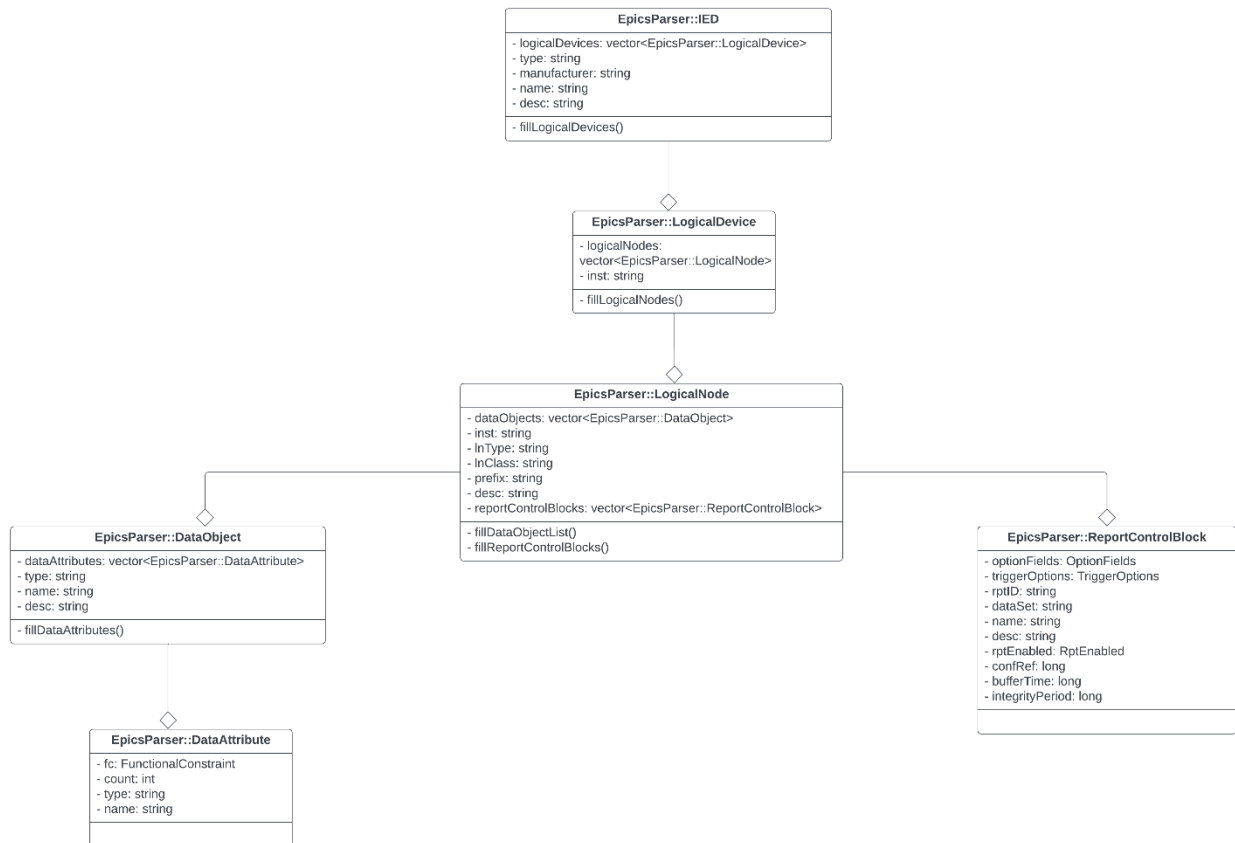


Figure 6. MMS Client UML diagram

CONCLUSIONS

To summarize the article, the work presented here represents a new form of control and protection in electrical substations, using the latest trends in software technology, such as virtualization, Docker, Redis...

In the end we have obtained a scalable, flexible, and easily configurable solution which provides interesting possibilities such as the integration of algorithms or services by third parties in the platform.

It is also an interoperable solution, providing an MMS interface to access and manage data using the above protocol and is also capable of integrating other IEDs from outside the platform through the MMS client service.

It has been successfully tested in the laboratory and in the field in a real substation.

All this allows us to envision future work opportunities, such as the application of machine learning algorithms that enable predictive maintenance, improve operational efficiency, and increase the accuracy of load forecasts. This would involve analysing large datasets to identify patterns and predict potential system failures before they occur. For a more standard configuration, the MMS Client connection parameters and references could be integrated in the global SCL file instead of the current xml document.

Another possible development could be a SCADA-like graphical interface from which to provide relevant information to operators as well as the possibility of real-time means of action on the platform.

To complement the IEC 62351-4 security standards for MMS communications, the full spectrum of security features (TLS and ACSE authentication) has been evaluated and will be implemented in the frame of the HE eFORT project (GA No 101075665), devoted to the analysis and testing secure countermeasures in digital SAS.

BIBLIOGRAPHY

- [1] IEEE Power System Relaying Committee, “Centralized Substation Protection and Control WG K15”, Dec. 2015. Available: <https://resourcecenter.ieee-pes.org/publications/technical-reports/PESTR0055.html>
- [2] M. T. Villen Martinez et al. “Software-defined Analog Processing based on IEC 61850 implemented in an Edge Hardware Platform to be used in Digital Substations”, Accepted for publication in IEEE Access, January 2024.
- [3] M. T. Villen Martinez et al., “Description of an edge computing solution to be used in digital substations”, presented at the ISGT IEEE Europe 2023 conference, Grenoble, 26 October.