Interoperability of applications in the Smart Grids context

María Lerena Rubio, *Student*, Universidad Pontificia de Comillas, ICAI

Abstract — *This work focuses on the study of interoperability in the context of Smart Grids (SGs) and the application of the Industrial Internet of Things (IIoT) in them, with the goal of exploring how to make any application developed in this context work on any node management platform, regardless of the provider.*

In collaboration with Minsait, this work investigates the integration of contemporary information and communication technologies (ICT) in SGs, with the purpose of improving the reliability, stability, and energy efficiency of the system. Key challenges are addressed, such as the incorporation of distributed energy sources, the diversity of communication protocols, and the need for a robust architecture. Given this scenario, it explores how the IIoT architecture, together with the implementation of node management platforms, promotes data processing at the edge. It describes how these platforms enable applications and analytics capabilities to be deployed directly to the nodes, optimizing real-time decision making, and reducing the load on the core infrastructure.

The work underlines the essential role of the Web of Things (WoT) in standardizing protocols and APIs to achieve uniform communication between devices and services from different manufacturers and platforms. Interoperability is exemplified through the implementation of an application with its Thing Descriptor (TD), facilitating integration and communication between systems.

Index Terms — Smart Grids, IIoT, Interoperability, Edge Computing, Edge Nodes, Applications, Virtualization, Docker

I. INTRODUCTION

Smart Grids (SGs) are modern electricity distribution systems, vital for the transition of the energy market towards sustainability and efficiency. Key features of an SGs include optimizing asset use, integrating distributed generation and storage, ensuring power quality, anticipating, and responding to disruptions, protecting against physical and cyber-attacks, and enabling consumer participation.

Effective network architecture design is crucial for managing the large number of connected devices. In SGs, the bidirectional flow of energy is favored thanks to technologies such as control systems. By collecting and transmitting consumer data, SGs resemble telecommunications networks, benefiting both consumers and operators. This energy and communications infrastructure is shown in *Figure 1*. IoT devices in SGs optimize management, detection of faults and enable user-network interaction, maximizing energy efficiency. To avoid data overload, the application of decentralized processing through Edge Computing is proposed.

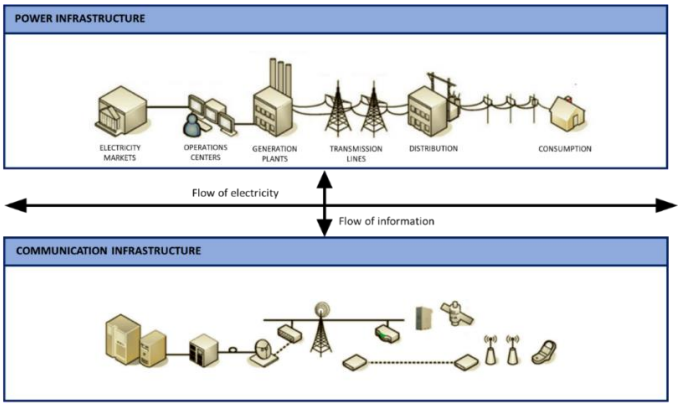


Figure 1. Smart Grid

Figure 2 proposes a new vision of information exchange in the Secondary Substation (SS) of a SG. The pyramid model illustrates the hardware, Edge Nodes, and centralized IoT platform for data exchange and control.

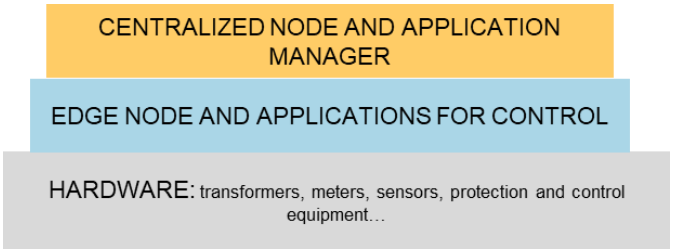


Figure 2. New vision of information exchange in a SS of a SG

The key challenge of SGs lies in the integration of components, systems, information, and applications. Interfaces and functionalities must ensure interoperability to enable high-level processes. The connection of all the components of an electrical network gives rise to an interconnected network in which the flow of information and its analysis will be carried out in real time.

The Web of Things (WoT) standardizes protocols and APIs, promoting device interaction and data exchange. Thing Descriptors (TDs) define device capabilities, properties, and interactions, fostering interoperability and integration within applications.

The main objectives of this project are to investigate the role of application management platforms and edge nodes in the vision

of SGs, explore the role of application virtualization in the search for interoperability, study WoT standards to achieve application interoperability and create a TD specification for interoperability between applications in the domain of a SS.

II. ANALYSIS

A. IoT in Smart Grids

Current distribution networks were not designed to manage bidirectional energy flows. As demand for energy grows and distributed generation increases, technical limits on supply lines can be exceeded, leading to power quality issues. This causes an imbalance between generation and demand at the local level, with slower responses and quality losses.

Faced with these problems, it is necessary to find measures that allow us to act more quickly in situations of energy imbalance. An additional challenge is the monitoring of distribution networks. Traditionally, it is done in HV and MV networks, leaving a lack of information in BT, where most consumers are connected. An alternative to address this is to connect the transformers of the distribution network to the control center, which implies high investment costs and workload. Another option is the implementation of the Industrial Internet of Things (IIoT), which installs sensors and gateways to collect and process data locally, monitoring the network more deeply.

The adoption of the IIoT represents a significant change for utilities, which traditionally use OT systems, such as SCADA, isolated from other systems and the internet. The process of adopting IoT in SGs can start by monitoring the LV network, capturing data from physical devices and processing information locally to generate alerts for parameter deviations. This reduces the load of data transmitted through gateways, decreasing wireless communication costs.

The adoption of IoT in SGs presents revolutionary opportunities as shown in *Figure 3*. But it also brings security, privacy and interoperability challenges that must be carefully addressed.

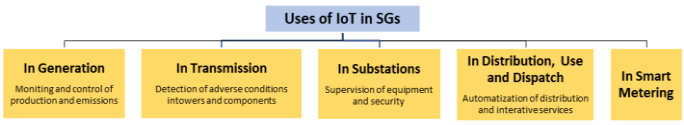


Figure 3. Uses of IoT in SGs

1) Focus on Interoperability

Focusing on interoperability, there are different categories that describe the different levels of interoperability that can be considered in the context of communication between systems and platforms in IoT technologies.

In this work and within the study of interoperability, we will focus on the network interoperability layers, syntactic and semantics, to study their application in the context that any developed application can work on any platform, regardless of the provider.

- Network interoperability: It is responsible for transporting information between different devices that interact through different communication networks.
- Syntactic interoperability: It is the agreement on format and structure rules used when encoding the information that is exchanged between devices.
- Semantic interoperability: It is responsible for ensuring that the transmitted data is interpreted in a common way between the different systems and applications. All actors involved in the transmission of information, both senders and receivers of data, must be under a standardized information model that will be used as a reference. The most common way to get systems to understand the same language is through data ontologies, which are a kind of maps that clearly explain what each thing means, helping devices and systems understand and share information correctly.

Through ontologies, the information that explains the data shared by IoT devices can be transformed into a structured sequence, which helps to make sense of the information coming from the devices and to understand it correctly, since a clear context is established for its subsequent interpretation. Ontologies can also function as a channel for sending different types of instructions to devices. To do this, they provide guidelines to ensure that commands are executed correctly, including security and access to information, as well as device operations.

2) IoT communication protocols applicable to SGs

In the context of IoT, it is critical to explore in depth the communication protocols that enable efficient integration between different devices and systems. In particular, it is convenient to delve into the MQTT protocol in the IoT context, especially for this work, due to the following:

- It is designed for IoT applications with low latency and limited bandwidth, essential in SGs where fast and efficient communication between IoT devices and the electrical grid infrastructure is crucial.
- It is ideal in scenarios with unstable connections and low power consumption, ensuring reliable communication even in challenging conditions, relevant in SGs with devices in remote areas or adverse environmental conditions.
- Its publish-and-subscribe topology benefits SGs where multiple devices need to monitor and receive data from multiple sources, enabling efficient distribution of messages to multiple subscribers.
- It is suitable for real-time monitoring and remote-control applications, important aspects in SGs where fast transmission of data from sensors and IoT devices enables effective real-time network management.
- In terms of security and privacy, MQTT simplifies message encryption using TLS and client authentication using modern protocols such as OAuth.

How MQTT works is simple and follows a publish-subscribe pattern, shown in *Figure 4*.

- The editor creates the message and publishes it to a specific topic.
- The subscriber receives messages relevant to the topic to which he has subscribed.
- A broker communicates with clients through a local network or internet connection.

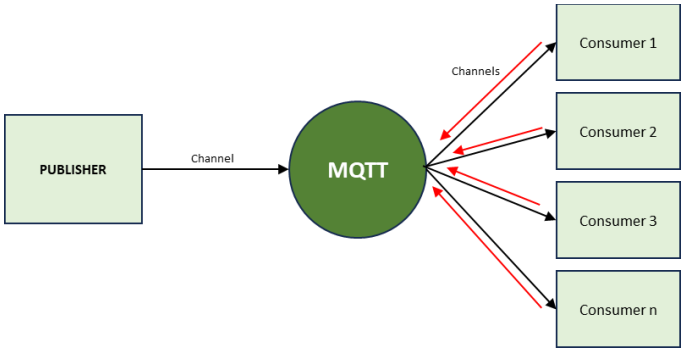


Figure 4. MQTT Structure

B. *Importance of "Edge" in Smart Grids*

Accompanying the IoT, the term Edge is common. Applied to SG, an edge node is located at strategic points where data is generated, transmitted, or consumed in the different stages of generation, transmission, and distribution of electrical energy. The primary function of an edge node is to process, analyze, and make decisions about data in real time before transmitting it to a centralized location, such as a data center or the cloud.

In this way, basic and advanced measurement data concentrators and supervisors in LV enable the local collection and processing of energy consumption data and supply monitoring in the LV network. This makes it easy to detect and resolve issues in real time without relying on the connectivity and latency associated with sending data to a centralized location.

C. *Application Virtualization*

Application virtualization allows applications to run in isolated and virtualized environments, separate from physical infrastructure and other applications on the system. Containers or virtual instances are created that encapsulate applications and their dependencies, providing a consistent and isolated environment for their execution.

In IoT and SGs, application virtualization has benefits such as resource efficiency, simplified management, minimal disruption, and rapid provisioning. Hypervisors and containers are prominent virtualization techniques. Containers, especially Docker, stand out for flexibility, portability, and agile performance. Figure 5 shows a container-based virtualization structure.

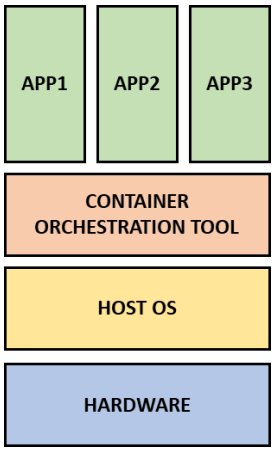


Figure 5. Container-based virtualization

Docker is chosen for its ease, portability, and scalability. In IoT Edge Nodes, Docker provides advantages such as application portability and fast, lightweight performance compared to virtual machines (VMs). Docker allows you to allocate and limit CPU, memory, network, and disk resources, ensuring an equitable distribution. Its architecture is based on containers with features such as "cgroups" and "namespaces", allowing to manage, limit system resources, and isolate processes. These technologies prevent interference and ensure an isolated environment for each process.

D. *Centralized node and application manager*

IoT platforms facilitate the development and deployment of IoT systems, allowing users to focus on what is most important to them, the operation of their businesses. With the increase in the number and geographical dispersion of IoT device deployments, the need for a centralized management system that provides basic information about the status of deployed devices and allows software updates remotely becomes evident. Applications running on devices must be deployed remotely and securely. Operational robustness is essential in any IoT platform, as well as cybersecurity, especially in the industrial area.

When selecting an IoT platform, it is useful to distinguish between generic platforms and more specific products. The former, such as Azure IoT Hub, AWS IoT Core, Google Cloud IoT, and Oracle IoT Cloud, offer a broad ecosystem of tools, but often have a steep learning curve and are not tailored to specific needs. On the other hand, specific platforms, such as Minsait Onesite Phygital Edge, offer more limited capabilities but better user support and shorter development times.

IoT nodes act as intermediaries between data capture devices and centralized processing systems. These nodes focus on efficiently collecting and transmitting data to core systems or the cloud for deeper analysis. The implementation of IoT nodes involves physical installation and configuration with the logic and algorithms necessary for their purpose. These nodes are where the operating system and applications run, either natively or in Docker containers.

1) Onesite Platform

Minsait Onesite Platform is comprehensive and offers modules that allow organizations to develop customized solutions, from data management and analysis to the creation of IoT applications and solutions. The platform offers capabilities such as asset and data management, advanced analytics, visualization, application creation, and more.

2) Phygital Edge Solution

Onesite Phygital Edge is a component of the Onesite Platform that focuses on managing IoT devices and systems at the network edge. This platform is based on a processing system close to the devices on the network. A virtualization and container architecture are deployed on these nodes to efficiently distribute intelligence in the form of microservices.

The architecture of the Phygital Edge platform is based on three key components, which work together to enable complete device management in the field and offer functionalities through a specialized agent:

- Field devices: These devices are connected to the platform to allow remote management.
- Application and Edges Management System: This distributed global management center has as its main function to commission, access, configure, update, and manage edge devices. This component is shown in *Figure 6*.

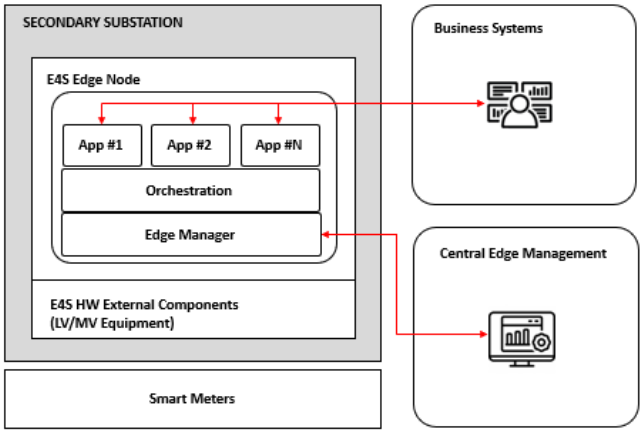


Figure 6. Application and Edges Management System

The figure of the agent is important, since it facilitates the interaction between the central administrator and the edge node, guaranteeing the functionalities of the module. Agent management involves controlling and managing the software agents installed on the edge node, which can execute commands, collect data, and transmit it. In addition, the Edge Node and Application Management System must enable remote installation, uninstallation, upgrade, and downgrade of agents on the edge node, and monitor their status and performance. You must also provide an API for agent management, used by the graphical interface and command line.

• Edge Engine: This set of capabilities deployed on nodes through containers enable remote control of nodes in the field, local information acquisition and processing, as well as connection to the enterprise cloud to scale and perform deeper analysis.

In terms of security, the Phygital Edge platform employs an identity record that stores information about devices and modules authorized to connect. The devices are authenticated in the Management System using credentials stored in the registry, ensuring secure and reliable connections.

3) Application to the Secondary Substation

Various applications of Phygital Edge have been identified in the SS, and the use cases and requirements to make them feasible have been summarized in Table 1.

TABLE I
USE CASES OF PHYGITAL EDGE IN A SS

Use Case	Description	I/O Requirements
Data Concentrator for Smart Meters	Collects data from smart meters using PLC PRIME technology, and sends it over the WAN connections available in the Data Hub	It requires a PRIME Bse Node connected to the 3 phases of the neutral of the LV network in SS
Advanced LV Supervision	Monitors the LV panels on the SS. Communicates via DLMS with LV I/O cards	Requires the installation of I/O cards in each phase of the LV panel, to measure instantaneous values (V, I, P and Q) of each phase
Network Topology Identification	Calculates the LV topology, identifying the line and phase for each smart meter connected to the LV panel	Does not require additional I/O as it uses data from previous applications
Interruptions Identification	Identifies outages in the LV network, estimating the size of the outage and its impact on customers	Does not require additional I/O as it uses data from previous applications
Transformer Regulation Monitoring and Control	Monitors the LV parameters of the transformer and regulates the output voltage to adapt the power quality to the state of the grid. Provides an interface with the transformer controller to modify the output of the transformer	Requires an I/O module that can operate the transformer controller
Video Analysis for visual monitoring	Cover several use cases based on the video signal coming from one or several digital cameras that capture different areas of the substation	Requires one or more video cameras
Substation Load Balancing and Load Profile	Calculates SS load statistics by phase, circuit, and transformer to suggest better customer distribution	Does not require additional I/O as it uses data from other applications
Generation and Demand Prediction Analysis	Calculates demand and generation predictions based on historical data from smart meters and LV panels	Does not require additional I/O as it uses data from other applications
Distributed Energy Resources Management (DERMS)	Operates distributed energy resources to respond to specific grid situations	Requires an I/O to communicate with the DERs

The container orchestration system, an essential component of the platform, enables efficient management of applications and resources in complex distributed environments.

The designed communication model seeks three types of information exchange between the software components deployed in the substation.

- Message Type 1 - On-Demand: Application sends MQTT to request information.
- Message Type 2 - Scheduled: Application uses MQTT to schedule task or request info, with details and URI to access later.

- Message Type 3 - Spontaneous: Application publishes data in MQTT topic; others subscribe to receive data.

These communication flows must be compatible and are achieved using the MQTT protocol, which is combined with the WoT architecture to achieve interoperability and efficient communication between applications.

WoT allows applications deployed on different nodes to communicate seamlessly and efficiently. The TDs define the IoT devices present in the SS, incorporating their properties, actions, and events, facilitating the integration of new devices, and offering a unified view of the substation's assets. TDs components are mentioned in the following *Figure 7*.

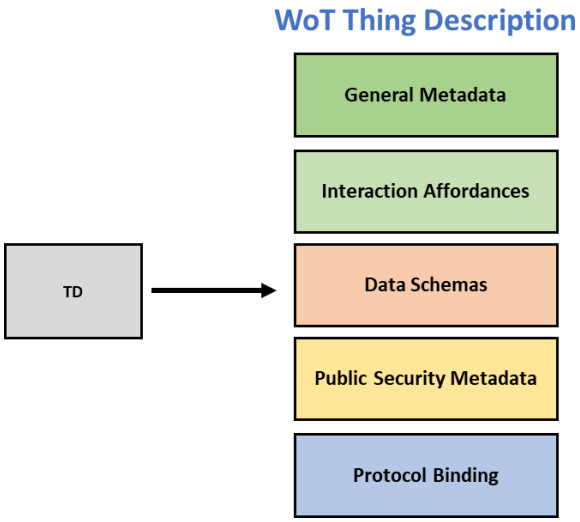


Figure 7. Thing Descriptor components

The SS-specific WoT-A architecture treats applications as "Things," each with a full set of functionalities including information reading, status updates, notification subscriptions, and features. The WoT Directory is a fundamental part of this architecture, where applications hang their ID cards. Communications between applications occur directly without intermediaries, and an efficient communication protocol, such as JSON-MQTT, is used to ensure improved interoperability.

This combination uses JSON for structured data and MQTT for efficient communication, a combination that helps improve interoperability between applications and data in SS. The common use of JSON in this context and specially in TDs is due to its readability, simple structure of key-value pairs which facilitates the representation of information, recognition and support in multiple platforms and languages, flexibility to describe capabilities and details in a coherent way, fluid integration with web technologies and APIs, efficiency in data transmission in IoT networks with limited bandwidth, and a wide support of tools and libraries for manipulation in various programming languages.

III. DEVELOPMENT OF SPECIFICATION

In this work, an application has been deployed together with its corresponding TD, based on the WoT standard. The programmed TD defines the capabilities, properties, interactions, and other relevant information about the Thing, which in this case represents a hardware element, allowing interoperability and seamless integration with WoT applications. This TD facilitates integration and interaction with IoT devices or services by other applications or platforms.

The final objective of the application is to receive signals from IoT devices deployed in a Minsait laboratory environment, fill a TD of a Transformer Temperature Sensor with the information of those messages coming from this type of device. Then another app reads that TD and creates alerts when the temperature exceeds a previously determined value.

The hardware device selected to fill the proposed TD and whose measurements are subsequently monitored is a wireless temperature sensor. These compact sensors can communicate without physical wires and use protocols such as Bluetooth, Wi-Fi and Zigbee, facilitating fast and reliable transmission of temperature data. In addition, wireless technology offers flexibility and scalability, as sensors can be easily deployed and moved, adapting to different monitoring scenarios. In the context of SGs, these sensors are very useful for monitoring temperature variations in electrical equipment, transmission lines, substations, and other critical grid components. This continuous monitoring allows operators to detect potential faults or problems in the network caused by overheating, taking preventive measures to improve the reliability and security of the network.

The practical part of this work establishes several fundamental objectives, including efficient communication between devices and systems using the MQTT protocol, management of data generated by IoT devices, extraction of relevant information from the data received, and interoperability to facilitate integration and compatibility between devices and applications in the SG. Their application

Strengths include the use of the MQTT protocol, widely adopted and supported by various devices, the JSON format for data exchange, and custom TDs to structure and standardize information about devices and capabilities. In addition, in the context of SGs, these strengths enable greater energy efficiency, electrical load management, real-time monitoring and the integration of renewable energy sources.

Table 2 presents the main tools used for the development of the specification, along with their application in the field of SGs.

TABLE II
TOOLS AND APPLICATION IN SGS

Tool	Description	SGs application
Python	Versatile and easy to use programming language	It facilitates the development of IoT applications in SGs
	Extensive community and libraries for specific tasks	It allows to process data and communicate with devices in the context of IoT
	Multiplatform, compatible with various operating systems	It ensures that the application works in different environments
	Flexibility to develop a variety of applications	Adaptable to different requirements in the context of SGs
	Json: Facilitates data processing in JSON format	Structure the TD and temperature data in an organized way
	Paho.mqtt.client: Implements MQTT communication	Establishes efficient communication between the thermostat and the central system
	Datetime: Provides tools for working with dates and times	Records the date and time of temperature data for analysis and tracking
Docker	Container platform that makes it easy to create, deploy, and manage applications	It provides an isolated and consistent environment for development and execution
	Portability and consistency in different environments	It ensures that the application works consistently across various platforms
	It facilitates teamwork and avoids configuration conflicts	It streamlines the development and deployment of the application in different stages
	Isolation and security in the execution of the application	Improves security by running applications in isolated containers
MQTT Explorer	Open source MQTT visualization and debugging tool	Verifies MQTT communication between the thermostat and the central system
	Provides monitoring and debugging of MQTT messages	Facilitates the identification and resolution of communication problems
	Intuitive graphical interface to represent MQTT communication	It allows to visualize in real time the messages and topics sent and received

A. *Model*

In the context of a SS, a wireless temperature sensor is used to capture the outside temperature of the transformer. This device communicates via Zigbee with the Zigbee Gateway.

The Zigbee Gateway is a device that acts as an intermediary between Zigbee devices and other communication systems. Basically, it allows Zigbee devices to communicate with systems and applications that use other communication protocols, such as MQTT. Information collected from Zigbee devices, such as sensors or actuators, can be transmitted via MQTT to the central system.

Once IoT applications and devices have performed their calculations or carried out their actions, it is crucial that the results and data generated are shared and communicated effectively. To achieve this, a Node and Application Management Platform is used, which acts as a control and coordination center for the IoT infrastructure. However, this part of the communication with the platform is not included in the practical development of this work.

Figure 8 represents the model proposed for the development of the practical part of the work.

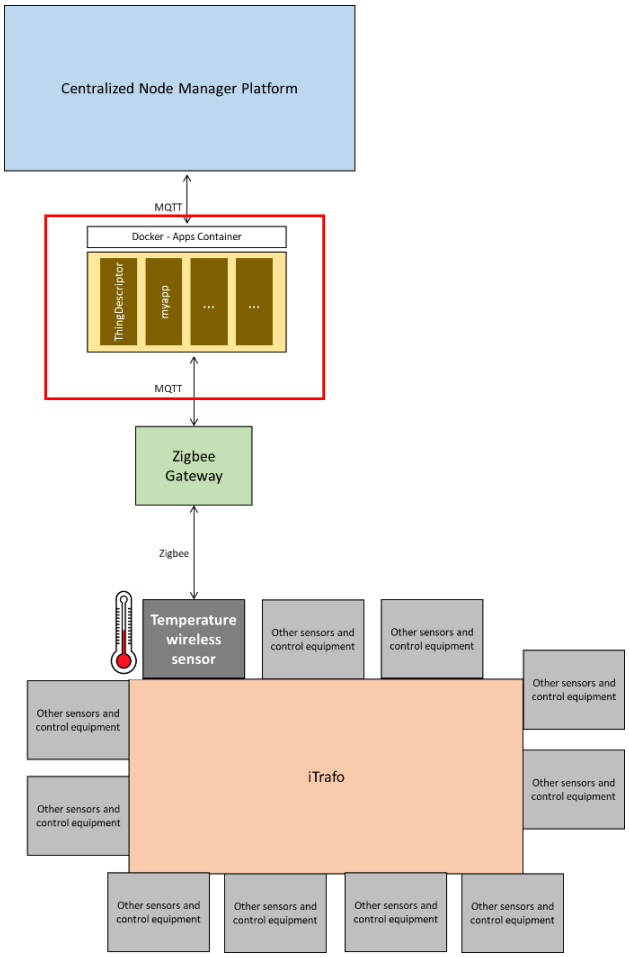


Figure 8. Model of the practical part of the work

B. *Results*

Test #1: A message published by MQTT is read from a device. The deviceType is a temperature sensor of the SS, and whose temperature does not exceed the limit value of 20ºC, as shown in *Figure 9.*


```
[{
  "profile" : "zigbee",
  "deviceId" : "000D6F00040DA3D6",
  "signalId" : "0201A0000",
  "signal" : "LocalTemperature",
  "description" : "Report",
  "value" : "220",
  "timeStamp" : 1689070624187,
  "timeStampInNanos" : 1689070624187000000,
  "deviceType" : "THERMOSTAT",
  "number" : 18
}]
```


Figure 9. Test #1: Message Read from a Thermostat

From it, a message containing the TD filled with the information read has been published. In another MQTT topic, the corresponding message for temperature measurement within the limits has been received. These are shown in *Figure 10.*

Topic 1

Topic 2

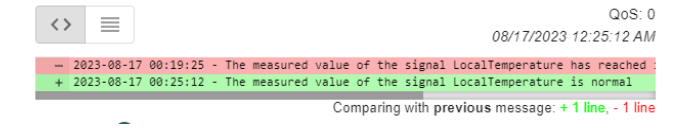


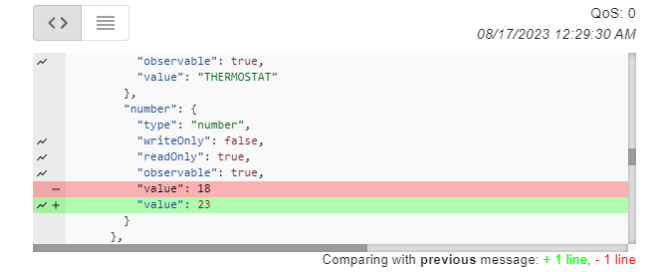
Figure 10. Test #1: Published messages (on the top the TD, on the bottom the limit message of T[a])

Test #2: An MQTT-published message is now read from the same device type, but exceeding the 20°C limit, as shown in *Figure 11*.

Figure 11. Test #2: Message Read from a Thermostat

Again, the corresponding messages are published with the information read, as shown in *Figure 12*.

Topic 1



Topic 2

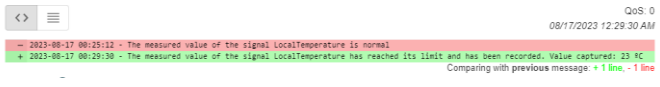


Figure 12. Test #2: Published messages (on the top the TD, on the bottom the limit message of T[a]

Test #3: The correct containerization of the developed applications is checked. Three containers have been created on the system and all of them are running correctly.

```
1.  CONTAINER
    ID   IMAGE
              COMMAND                  CREATED          STA
    TUS          PORTS
      NAMES

2.  dd2473d82c99   myapp_image:latest
                        "python myapp.py"        13 seconds
    ago   Up 12
    seconds

    myapp_container

3.  6e72ccb33867   thing_descriptor_app_image:latest
                        "python ThingDescrip…"   4 minutes
    ago    Up 4
    minutes
    thing_descriptor_container

4.  fb35bfe64c8a   emslab.onesaitplatform.com/onesait-
    things/edge-mqtt:2.0.0   "/docker-entrypoint.…"   2
    weeks ago      Up 2 weeks      0.0.0.0:1883->1883/tcp,
    :::1883->1883/tcp   edge.mqtt
```

The developed specification marks a milestone in the integration of IoT devices within the WoT framework. With the addition of TDs, interoperability with a wide range of IoT devices is ensured. Although initially conceived for a Transformer Temperature Sensor, its inherent adaptability makes it a versatile solution for different devices. This flexibility enables agile integration into the WoT ecosystem, which in turn streamlines deployment and management when containerizing applications. Ultimately, this approach enriches the usefulness of IoT devices by encouraging the exchange of data and alerts, promoting interoperability across diverse applications and platforms, regardless of vendor.

IV. ECONOMIC IMPACT

The economic impact analysis focuses on distributed computing and interoperability in this project.

The incorporation of edge nodes into the BT network seeks to modernize it and has significant economic implications. These investments are distributed in key areas, such as the installation of Edge equipment, the management platform, interoperability certifications, licenses and costs of applications, necessary components, and operation / maintenance. This translates into operational efficiency and savings, such as reduced inspection costs, repairs, and compensation for interruptions, among others. The adoption of distributed systems also optimizes resources and reduces technical and non-technical losses, with a flexible focus on new functionalities.

The interoperability of applications under a standard offers advantages, such as reduced development costs and efficiency in data exchange, encouraging competition between suppliers and avoiding dependencies.

V. CONCLUSIONS

- The importance of application management platforms and edge nodes in SGs for management and monitoring is confirmed, with edge nodes allowing preliminary on-site analysis. Centralized management becomes crucial with the expansion of IoT devices, enabling remote updates and an integrated environment for development and debugging. The Onesait Phygital Edge platform, based on virtualization and containers, is ideal for distributing information in the energy value chain.
- Virtualization technology promotes efficiency, simplified management, and portability, with Docker as a choice backed by its ease of use, scalability, and adaptation to the changing demands of SGs.
- The implementation of WoT concepts allows fluid and efficient communication between applications, facilitating interoperability. TDs and WoT directories promote collaboration and communication pattern and JSON-MQTT combination improve interoperability and data exchange.
- Through the development of the specification, the seamless integration of applications with hardware in secondary substations is demonstrated, creating applications to build TDs of temperature sensors and generate alerts. This flexibility facilitates adaptability to different IoT devices and enriches data exchange and cooperation between applications and platforms, regardless of the provider.

VI. RECOMMENDATIONS FOR FUTURE WORKS

The integration of IoT in Smart Grids, just with the research in the standards defined by WoT to achieve interoperability of applications in this context, has marked a milestone in the evolution of electrical infrastructures. Now, by exploring the possibilities of AI, new doors are opening to further drive efficiency, sustainability, and automation in the power grids of the future. In the following publication [100], information of great interest is presented to make some recommendations for future work in the field of SGs and AI.

Considering the increasing complexity of power grids, future work could focus on the implementation and optimization of AI techniques, such as artificial neural networks, genetic algorithms, and reinforcement learning, to improve the efficiency and automation of distributed resource management in SGs.

Future work could also address how to implement fully automated and self-learning systems in SGs, using machine learning algorithms. That would allow the network to adapt and learn from past operations for planning and failure prevention.

As SGs become more susceptible to cyberattacks, therefore, another idea would be to explore how artificial intelligence techniques, such as machine learning and anomaly detection, can strengthen cybersecurity and privacy in these networks, ensuring the confidentiality and integrity of critical data and operations.

Based on the analysis of WoT as an essential part for interoperability in SGs, it could be explored how AI techniques can improve communication and collaboration between IoT devices from different vendors, investigating how to facilitate the translation and adaptation of data between communication formats and different protocols.

Exploring how AI can be employed in optimizing resource management, security, automation in the context of SGs could lead to significant advances in the efficiency and reliability of the power grids of the future.

REFERENCES

[1] O. VERMESAN, ADVANCING IOT PLATFORMS INTEROPERABILITY. NEW YORK: RIVER PUBLISHERS, 2022.

[2] A. GOUDARZI, F. GHAYOOR, M. WASEEM, S. FAHAD, Y I. TRAORE, “A SURVEY ON IOT-ENABLED SMART GRIDS: EMERGING, APPLICATIONS, CHALLENGES, AND OUTLOOK”, ENERGIES, VOL. 15, NÚM. 19, P. 6984, 2022.

[3] ‘PHYGITAL MINSAIT’, MINSAIT.COM. [ONLINE]. AVAILABLE: https://www.minsait.com/es/aceleradores/phygital.

[4] “GRUPO DE TRABAJO DE CT INTELIGENTE - FUTURED. PLATAFORMA ESPAÑOLA DE REDES ELÉCTRICAS”, FUTURED. PLATAFORMA ESPAÑOLA DE REDES ELÉCTRICAS, 24-MAR-2021. [ONILINE]. AVAILABLE IN: HTTPS://WWW.FUTURED.ES/GRUPO-TRABAJO-CT-INTELIGENTE/.

[5] “MQTT - THE STANDARD FOR IOT MESSAGING”, MQTT.ORG. [ONLINE] ONLINE IN: HTTPS://MQTT.ORG/. W. SHI, J. CAO, Q. ZHANG, Y. LI, Y L. XU, “EDGE COMPUTING: VISION AND CHALLENGES”, IEEE INTERNET THINGS J., VOL. 3, NÚM. 5, PP. 637–646, 2016.

[6] S. SINGH Y N. SINGH, “CONTAINERS & DOCKER: EMERGING ROLES & FUTURE OF CLOUD TECHNOLOGY”, EN 2016 2ND INTERNATIONAL CONFERENCE ON APPLIED AND THEORETICAL COMPUTING AND COMMUNICATION TECHNOLOGY (ICATCCT), 2016, PP. 804–807.

[7] “DOCUMENTATION - WEB OF THINGS (WOT)”, WWW.W3.ORG. [ONLINE]. AVAILABLE IN: https://www.w3.org/WoT/documentation/.

[8] “WEB OF THINGS (WOT) ARCHITECTURE 1.1”, WWW.W3.ORG. [ONLINE]. AVAILABLE IN: HTTPS://WWW.W3.ORG/TR/2023/PR-WOT-ARCHITECTURE11-20230711/.

[9] EUROPEAN PLATFORM INITIATIVE, “ADVANCING IOT PLATFORMS INTEROPERABILITY”, IOT-EPI.EU, 2018. [ONLINE]. AVAILABLE IN: HTTPS://IOT-EPI.EU/WP-CONTENT/UPLOADS/2018/07/ADVANCING-IOT-PLATFORM-INTEROPERABILITY-2018-IOT-EPI.PDF.