Journal of Internet Services and Applications, 2022, 12:1, doi: 10.5753/jisa.2022.2374

Towards Automating the Integration of Legacy IEDs into Edge-Supported Internet of Smart Grid Things

Wanderson Modesto [**Federal University of Rio Grande do Norte** | ***wanderson@ppgsc.ufrn.br***]
Lucas Bastos [**Federal University of Pará** | ***lucas.bastos@itec.ufpa.br***]
Augusto Venâncio Neto [**Federal University of Rio Grande do Norte** | ***augusto@dimap.ufrn.br***]
Denis Rosário [**Federal University of Pará** | ***denis@ufpa.br***]
Eduardo Cerqueira [**Federal University of Pará** | ***cerqueira@ufpa.br***]

Received 30 November 2021 • **Accepted** 19 July 2022 • **Published** 01 November 2022

Abstract
The prominence of the SG-to-Cloud continuum will pave the way towards advanced Smart Grid (SG) ecosystems and will enable cutting Edge applications and servers into the power energy vertical at unprecedented innovation levels. During the design of future Smart Grid ecosystems, legacy Intelligent Eletronic Device(IED) cannot be left behind, whereby their full integration into the Internet of Smart Grid Things(IoSGT) reveals itself as a continuous issue. In an attempt to tackle this challenge, we are introducing the Legacy Smart Grid to IoT Integration Approach(SG2IoT), which automates the integration of multiple legacy IEDs in a scalable and flexible environment made possible by the IoSGT. Aside from that, the SG2IoT establishes an SG-to-Cloud continuum for provisioning architectural modular components for running in a distributed approach at Cloud facilities spread in Edge and central datacenters. Finally, the SG2IoT impact estimation was made up of harnessing a prototype running atop a lab-premised testbed that features real-world technologies. Outcome analysis proves the viability of the SG2IoT lightweight approach by establishing an SG-to-Cloud continuum to afford low times responses and affordable IoSGT scalability.

Keywords: Smart Grid, Legacy IED, IoSGT, SG-to-Cloud Continuum.

1 Introduction

Energy systems are being modernized by Smart Grid (SG) in the form of an ecosystem that integrates modern Information and Communication Technologies (ICTs) to establish an automated, intelligent, and widely distributed energy network (Pfeiff *et al*., 2020). Internet-of-Things(IoT) and Cloud Computing head the list of mature ICTss to pave the way for the next generation of SGs Barja-Martinez *et al*. (2021). In the former, IoT-based Smart Grid Systems (Internet of Smart Grid Things (IoSGT)) harness a set of Intelligent Electronic Devices (IEDs) that are equipped with transmission protocols to deliver measurement data to the operations center of the power operator Rouhani *et al*. (2020).

The modern data center network domain uses edge server interactions to provide a collaborative computing approach, striving to outperform classical centralized SG systems on several advantages (depending on application workflow) Ramalho and Neto (2016). In the IoT environment, the edge-cloud computing continuum becomes the efficient and seamless integration of SG specific services and applications (including third-party solutions) across multi-vendor computing and network convergence-grade platforms. As a result, the deployment of applications for running at edge data centers premises foresees monitoring and controlling operations with greater flexibility, allowing to avoid security and privacy threats while leveraging the scalability and energy efficiency of central cloud data centers at the same time and many other reported benefits Modesto *et al*. (2021).

At the operations center, analytic applications process the measurement data employed for protection estimation, power quality assessment, network monitoring, energy metering, and other purposes Saleem *et al*. (2019). In the case of the latter, the expansion of the Cloud domain to the network Edge premises (a.k.a. Edge Computing Negash *et al*. (2018)) allows service applications to be introduced close to IEDs so that achieving latency rates at unprecedented low levels.

The aforementioned modern SG systems, covering the list of cutting-edge ICTss, are now referred to as SG-Cloud-IoT Mota *et al*. (2019). It can be claimed that the full realization of the SG-Cloud-IoT has the potential to lay the foundations for future benefits by leveraging a truly advanced set of features. Moreover, new opportunities can emerge in the SG market and add value through intelligent innovations that can significantly boost both capital expenditure (CAPEX) and operational expenditure (OPEX). However, legacy IEDs should not be left behind during the evolutionary period of a conventional electrical system for the SG-Cloud-IoT, as they might prevent the changing process from being viable.

A legacy Intelligent Electronic Device (IED) stands for a conventional device (power sensor and/or actuator) that has already been integrated into an electrical system for years. A networked legacy IED can be accessed through a standard bus protocol that leverages the means of establishing a logical communication between IEDs in electrical substations, primary and secondary devices, and Supervision and Data Acquisition (SCADA) systems coexist in the same power system operators. Both DNP3 IEEE Boakye-Boateng *et al*. (2021), and IEC 61850 IEC61850 (2020) represent the main legacy SG protocols for automated substations and can be found in practically all of the SCADA application sys-

tems Tightiz and Yang (2020) Njova *et al*. (2020).

If full interoperability between legacy IEDs and other energy systems are ensured, this can play a significant role in achieving IoSGT. Within the scope of the present research study, the incorporation of legacy IEDs into the IoT is of paramount importance for the big energy players by enabling them to move towards the full realization of IoSGT. Through ubiquitous telemetry through power data, it is possible to conduct a value-added analysis of the system as a whole Farri and Ayubi (2022). Most of the available solutions that address the question of integrating legacy IEDs into the IoT adopt a hardware-based gateway approach, in which network additional nodes encapsulate incoming power messages into another outgoing protocol in a centralized manner. Firstly, solutions requiring the addition of new hardware-based gateway nodes are intrusive and impact a heavyweight computing approach by concentrating all incoming messages for being processed to encapsulation afterward. Finally, flexibility and scalability are issues in this kind of centralized framework since it regularly supports a single SG protocol (without updating the means of enhancing and supporting technologies) atop a resource-restricted hardware platform Silva *et al*. (2013).

Moreover, the IoT platform has emerged as the central point service of the IoSGT since it shares the benefits of the open Internet with a system that, classically and strategically, is proprietary by definition Cheruvu *et al*. (2020). APIs are commercially available for large-scale use; for example, the NGSI adopted by the FIWARE platform employs strategies to enable communication throughout the IoT. Hence, it has become cost-effective to devise different applications for the managing and controlling of IEDs that coexist in the underlying SG without the need to take account of the intricate details of their heterogeneous hardware and software designs Saxena *et al*. (2017).

In this article, we design the Legacy Smart Grid to IoT Protocol Integration Approach (SG2IoT) to tackle the challenges mentioned above. The SG2IoT employs a holistic solution to establish the SG-to-Edge-to-Cloud continuum within the IoSGT infrastructure, operating in a distributed approach across Edge- and Cloud-supported nodes. The SG2IoT has progressed beyond traditional gateway-based protocol-translation solutions by handling the orchestration of the legacy IEDs in the IoSGT, enabling architectural components to run and interwork at different computing layers of the IoT-to-Edge-to-Cloud continuum in a fully automated manner. The SG2IoT functional architecture comprises modular components with different containerized nodes spread across the IoT-to-Cloud continuum, foreseeing providing enhanced agility to handle high granular IEDs simultaneously by on-premise functions.

This article extends our previous work described in Modesto *et al*. (2021), and apart from the SG2IoT system, it seeks to make the following contributions: (i) establishing the SG-to-Cloud continuum by introducing capabilities at nodes spread over the IoSGT ecosystem. The purpose of this is to achieve a performance-enhanced and flexible system while obtaining low latency and conducting analysis in an on-premises environment (i.e., at the extreme Edge); and, (ii) an evaluation of the system by prototyping in a lab-premised testbed consisting of real-world devices and techniques to ensure highly accurate outcomes and new perspectives. The outcome analysis suggests that the lightweight approach of the SG2IoT prototype enables low response times in the testbed, complying with stated SG requirements (IEC61850 (2020) and by Kuzlu and Manisa (2013)) while offering a reasonable level of scalability.

The rest of this article is structured as follows. Section 2 examines the principal studies related to the background of this research. Section 3 describes the SG2IoT system designed for this work. Section 4 provides the results of proof-of-concept prototyping in an actual laboratory testbed. Finally, Section 5 concludes the article and makes recommendations for future work in the field.

2 Background

In this section, we introduce details about some concepts and modules of the FIWARE platform and also talk about the two main smart grid protocols (IEC 61850 and DNP3) covered in this work.

2.1 FIWARE

FIWARE is an open-source platform that is used in the development of smart applications, and IoT systems fiware.org (2022). FIWARE offers custom features for developing application components that can be used for free. It uses open-source components called Generic Enablers (GE), which are supported by various FIWARE platform partners, making them an alternative to proprietary solutions. One of the many advantages of this model is that the GEs can be used independently, having the option of being used in a local or remote environment and being activated according to the application's needs.

FIWARE uses the Next Generation Service Interface (NGSI) specification standard developed by the Open Mobile Alliance (OMA). This model provides an interface for various web services through a Representational State Transfer (REST) API. All interactions between platform components take place using FIWARE's RESTful API; thus, it manages the entire lifecycle of context information, all the subscriptions and registrations, also updates and queries FIWARE-NGSI (2018). The NGSI API establishes specific terminologies of its concepts and has its information structured in entities, attributes, and context elemenThe architecture of the FIWARE platform and its main components are divided into modules or chapters FIWARE-CATALOGUE (2019): Core Context Management, Interface with IoT, Robotics and Third-Party Systems, Data/API Management, Context Processing, Analysis and Visualisation, and Context Data/API Management, Publication, and Monetization. Its main component is a FIWARE Context Broker Generic Enabler called Orion.

2.1.1 Orion Context Broker

The Orion Context Broker GE is the central component of the FIWARE platform and is based on the Publish/Subscribe

model. Orion is responsible for managing the entire lifecycle of context information. Its main features include updates, queries, registrations, and subscriptions FIWARE-ORION (2019). Context-producing devices register themselves, send updates to Orion, and then store them in the database. Context-consuming applications can consume the data and be notified through subscriptions about context changes of certain attributes of an entity or receive notifications with a certain frequency, for example, every minute or every hour. Orion uses a MongoDB[1] database to store context data in JSON format. Context data is not stored persistently, as context is based on each device's last measurement of the state of real-world objects at a given time.

2.1.2 IoT Agents

An IoT Agent FIWARE-IoTAgent (2019) is a component of the FIWARE architecture that allows groups of devices to send their data and be managed from an NGSI Context Broker (Orion) using native protocols. IoT Agents must also be able to handle the security aspects of the FIWARE platform (authentication and authorization) and provide other services common to devices. Within the FIWARE architecture, IoT Agents operate as protocol-translating gateways that bridge the gap between traffic sent/received by the devices and traffic sent/received by the internal components of FIWARE that uses the NGSI standard to communicate.

Each device will be mapped as a context entity associated with Orion when using an IoT Agent. The device 'id' will be mapped by default to the same entity identifier, and the IoT Agent will select the entity type, whereas the name and type are user-configurable. Each value received from the device is mapped to a different attribute. This way allows queries or subscriptions to the status changes of device parameters to be performed through queries to subscriptions to the attributes of Orion's context entities.

2.1.3 Cygnus

The Cygnus GE FIWARE-CYGNUS (2019) is one of the components in charge of connecting Orion with a persistent database so that creating a historical view of such data. Cygnus will register to listen for context changes and save them to a database. It can work with a database like PostgreSQL, MySQL, and MongoDB. To collect and store device data or interact with devices, it is first necessary to connect IoT devices to the FIWARE ecosystem through an IoT Agent.

2.2 Smart Grid Protocols

The SG systems utilize various specialized standards, protocols, and technologies for intercommunication between systems within substations in the energy sector. A protocol for SG plays a key role in providing a bus process between IEDs and application software that coexist in the same ecosystem as the power operator.

[1] https://www.mongodb.com/

2.2.1 IEC 61850

IEC 61850 is an international standard for Substation Automation and Communication Systems and Electrical Resource Management. It is considered one of the standards that facilitate the development of the SG. The IEC 61850 standard brings the idea of an interoperability key, which promotes communication between equipment from different manufacturers and makes IEDs from different manufacturers communicate with each other IEC61850 (2020).

The three main types of protocols/messages that the standard defines for the exchange of information between devices in a substation are:

- Generic Object Oriented Substation Event (GOOSE): multicast messages sent in the data link layer. These messages carry information between the IEDs and are responsible for exchanging data, and transmitting control and status messages, in addition to having delivery time restrictions between IEDs within the substation itself;
- Sampled Value (SV): it is responsible for the traffic of the analog readings of the substation. Sampled values, such as transformer current and voltage, are sent in SV messages and also have time restrictions;
- Manufacturing Message Specification (MMS): messages contain reports and information that were not time-constrained. They are used to exchange information that solely indicates the status of a particular device.

2.2.2 DNP3

The Distributed Network Protocol version 3 (DNP3), or IEEE 1815, is an industrial automation protocol commonly used to monitor and control equipment in the electrical energy sector. DNP3 is an open-source protocol that was developed to include the best features of other protocols used at the time of its creation and is more reliable than any previous standard IEEE (2012).

DNP3 works with both master and outstation devices. This type of configuration can be understood as a Master/Slave type and works as follows IEEE (2012):

- Master: device that periodically initiates requests to either collect data from other devices or perform control settings;
- Outstation: device that produces data, and variables, among other information, that other devices may intend to gather. It stores and sends data when requested by master devices. A remote station can be several different devices and systems, for example, an IED.

3 Related work

The integration of legacy IEDs into the SG infrastructure makes a constant demand on energy operators who pursue modernizing their power ecosystems. In response to the challenges raised by the SG-Cloud-IoT ecosystem, this section examines outcomes of the most significant research endeavors both in scientific and commercial domains. Our research

Table 1. Summary of Related Work

Works	Solutions	Techniques	SG	Cloud/Edge Scenario
Araújo *et al*. (2018)	Middleware	Integrate legacy electrical equipment into the infrastructure of the SG through wireless sensor networks.	Yes	No
Shin *et al*. (2017)	The CoAP protocol	Illustrating how the CoAP protocol, (which is based on REST services), could work in conjunction with the IEC 61850 data model in an SG environment.	Yes	No
Nugur *et al*. (2019)	A gateway-based architecture	Offering an IEC61850/DNP3 message encapsulation/translation service in packets of other networking protocol.	Yes	No
Our System	Controlling and monitoring the IEDs of legacy architecture that can be found in SG-Cloud-IoT ecosystems.	The method employed by the SG2IoT for integrating these physical devices into the SG-Cloud-IoT ecosystems.	SG protocols and SG-Cloud-IoT ecosystems	Yes

stands out within the scope of measures that aim at enabling interoperability between legacy SG systems and the Cloud-IoT environment while taking account of the integration of IEDs that are made accessible by the main SG protocols IEC 61850 and DNP3.

The authors in Araújo *et al*. (2018) created a middleware designed to integrate legacy electrical equipment into the infrastructure of the SG through wireless sensor networks. The middleware runs on the sink node of a ZigBee-based wireless sensor network, following the IEEE 802.15.4 standard. It is designed to address problems related to integrating legacy electrical equipment into an SG that uses a standardized communication protocol (e.g., IEC 61850, DNP3, or Modbus). The main feature of this middleware resides in its ability to translate messages that IEDs of the power substation send to the control center SG facility. However, this solution is not IoT compliant since it does not use any of the CoAP, MQTT, or HTTP transport protocols. Furthermore, it does it employ any Cloud-native functions.

In work carried out by Shin *et al*. (2017) the IEC 61850 model was encapsulated into the CoAP protocol. The authors achieved a robust but straightforward mapping from CoAP to IEC 61850 and compared CoAP and MQTT with the SOAP protocols to demonstrate the validity of its use. In this paper, the authors mainly focused on illustrating how the CoAP protocol harnessed REST services to work in conjunction with the IEC 61850 data model atop an SG environment. Finally, this solution is not designed with Cloud-native functions.

The market is mainly concerned with integrating legacy SG to IoT systems employing a gateway-based architecture. The gateway acts at the central boundary between the legacy systems network and the mobile operator's connection for backhaul by offering an IEC61850/DNP3 message encapsulation/translation service in packets of other networking protocols. Companies such as Cisco, Intel, Dell, and others, provide enterprise-scale intelligent IoT (gateways) such as Dell Edge Gateway[2]. However, these solutions were strictly manufactured to be IoT gateways and are only proprietary and compatible with the vendor's Cloud IoT platforms. This type of solution causes several problems in an SG system; for example, i) a part of the communication infrastructure might be owned and managed by third parties, ii) there might be privacy issues, iii) there are acquisition costs, and iv) increases the complexity of the SG Nugur *et al*. (2019).

Table 1 summarizes and compares the main characteristics of the most important works analyzed. It also shows the main requirements used in this work: the architecture or solution, the SG standards or protocols, the IoT communication protocols, and Cloud or Edge scenario.

To the best of the authors' knowledge, the literature review on the related work and commercial solutions reveals that some efforts have been made in science and commerce to integrate legacy SG protocols with SG-Cloud-IoT ecosystems. However, there was no available means of integrating different legacy SG protocols into a complete SG-Cloud-IoT environment that takes account of IoT communication protocols and the Cloud or Edge scenario, which underlines the innovative features and significance of our research. The following section describes our system in detail.

4 The SG2IoT System Proposal

The is designed to control and monitor the IEDs of legacy architecture that can be found in SG-Cloud-IoT ecosystems. The method employed by the SG2IoT for integrating these physical devices into the SG-Cloud-IoT ecosystems provides a more flexible, monitorable, and adaptable environment to accommodate new services and applications without causing significant changes in the neighboring scenario. The solution results from the research and development project IoT-based Centralized Energy Monitoring System directed to the CEA Network under the auspices of the Federal University of Pará (UFPA) in collaboration with the Federal University of Rio Grande do Norte (UFRN).

The architecture of the SG2IoT and the main components of the architecture that have been implemented are shown in Figure 1. It consists of a set of modular components that devote to integrating the IEDs with the components of the Cloud IoT platform. Generally, the components of the SG2IoT solution can be deployed on different server nodes spread over the SGIoT, but for the sake of simplicity, they have been displayed and deployed in a single instance.

The components of the SG2IoT architecture were developed with the aid of the Python language and were designed to be lightweight so that to run on System on Chip (SoC) devices that raise both a restricted memory and processing capabilities (e.g., the widely used Raspberry Pi) as well as on powerful computational nodes (such as data center servers).

The *Data Handler* includes a component called IED-DG that is the core and main component of the SG2IoT. It can communicate directly or indirectly with all the other architectural components. The *Data Handler* receives the parameters and settings of the IEDs data through the IED-M module. It sends broadcasting messages to discover new IEDs on its network at the time interval set by the system administrator. In addition to automatically facilitating the insertion of new devices, it also checks if the IED-M has sent new configurations, including new IEDs, or the exclusion of monitored IEDs. Then, immediately after the *Data Handler* receives the configuration information needed to add new IEDs to the list of monitored devices, it creates an instance of the specific

[2]https://www.dell.com/en-us/work/shop/gateways-embedded-computing/sf/edge-gateway

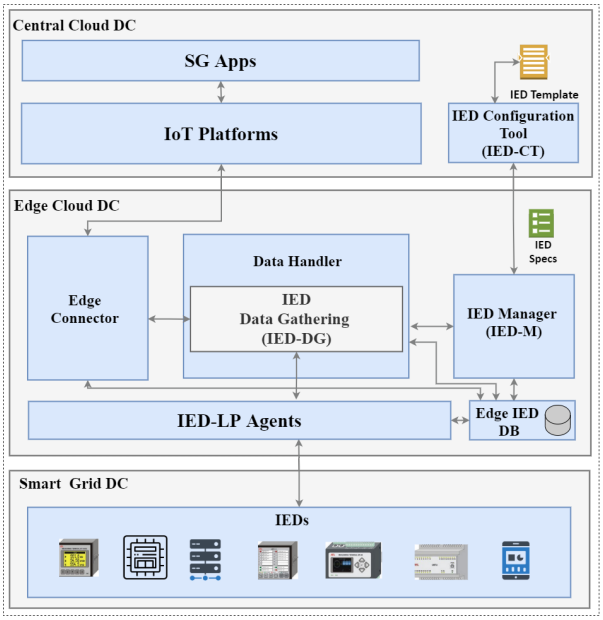


Figure 1. Architecture of the SG2IoT

SG *IED-LP Agent* protocol type, IEC 61850 or DNP3 of the IED in question.

The *Data Handler* is used by the IED-LP Agent to convert the data received from a connected IED (IEC 61850 or DNP3 formats) into an outgoing IoT format protocol (MQTT, CoAP, HTTP, and others). Each *IED-LP Agent* uses specific open source libraries, such as libiec61850[3] and pydnp3[4], depending on the device it needs to connect to. This module performs the tasks required to convert SG protocols.

The IED-LP Agent module performs the tasks related to the conversion of SG protocols and creates a specific server for each protocol. When the *IEC 61850 Agent* is executed, for example, it creates a server and waits for data to be sent by monitored devices to arrive on its respective TCP port (port 102 for the IEC 61850 instance). Then it reads and converts the received data, extracting the data and forwarding it to the *IED-DG* which extracts all the necessary information (id, data, source IP address, and other information), builds new packets, and forwards the message to the *Edge Connector*. For the *DNP3 Agent*, the only difference between the *IEC 61850 Agent* is in its respective TCP port (port 20000 for DNP3).

The IEC 61850 translator, from the *IED-LP Agent*, acts as the client and the IEDs connected to it are the servers. The simulations of the IEC 61850 standard protocols were implemented and adapted from *libIEC61850*. *libIEC61850* is an open-source library, in the client/server model, for the communication protocols IEC 61850/MMS, IEC 61850/GOOSE, and IEC 61850-9-2/Sampled Values, written in C. The library had some modifications made for the addition of new functions to perform the translation of IEC 61850 messages by the *IED-LP Agent*.

By default, the IEC 61850 protocol sends multicast messages, this message is sent to all devices on the network. Once an IEC 61850 server is instantiated, it simulates an IED generating GOOSE messages and SVs to simulate a communication environment with IEDs that have been connected to a substation network.

In this way, several scenarios can be simulated according to what is needed to test, such as automatic message execution scenarios, analysis, and diagnostics, faults, virtual analog voltage signals, the current generated in the form of current and voltage waves and simulation of changing the status of the IEDs breakers. IED detection requires prior knowledge of the Ethernet network configured between the IEDs. In this sense, it is important to know the individual IP addresses of each IED so that it is possible to monitor the messages sent by the IEDs.

The DNP3 works with both master and outstation devices. The master is the device that periodically initiates requests to collect data from other devices, and the outstation is the device that stores and sends data when requested by the master devices. The *IED-LP Agent* DNP3 translator acts as the master, and the IEDs connected to it is the outstation.

The simulations of the DNP3 protocol were implemented and adapted from the library *pydnp3*. This library is an adaptation for Python of the library *Opendnp3* opendnp3 (2020) which is a library created in C++ and is available through the Apache license, being the IEEE-1815 reference implementation (DNP3). The *pydnp3* library has a quick and straightforward installation process, but it has a list of dependencies that must be installed before its compilation.

The IED-DG component stores the settings and parameters of the data defined by the system administrator. The IED-DG contains information, such as the communication parameters required by the IED-LP Agent, when collecting new data from connected devices and sending it to applications for viewing or permanent storage.

The IED-M receives all the communications and parameters through a configuration file (*IEDSpecs*) that is dynamically created with the aid of an *IED Template* file received from the IED-CT. Figure 2 shows an example from a *IEDSpecs* file, described in JSON, responsible for holding the information of the devices that will be monitored. The configuration file contains the data that is used by the SG2IoT modules; for example, the "device_ip" shows the IP address of the IED on lines 6 and 2, and the "entity_type," on lines 7 and 23, describes which the type of protocol this IED has, IEC 61850 or DNP3. It also defines a "device_id" on lines 8 and 24. The "address_fiware," "mqtt_topico_fiware," "fiware-service," "protocol," "transport," "fiware-service path," "entity_name," "timezone," "URL" and "Content-Type" are used by the *Edge Connector* to establish a connection between the SG2IoT and the IoT platform, in this case, the FIWARE platform, since the server hosting the FIWARE instance is remote from the SG2IoT solution.

The *Edge-IED-DB* is a module used by the *IED-M* for keeping the log of data received and sent, recording data on active connections, configurations, creation time, IP addresses, data used by IoT applications and other information. It utilizes a MySQL database that can access all the SG2IoT modules. It also stores, for some time, IED data sent to the IoT platforms.

The *Edge Connector* module acts as a gateway and manager so that mediation data that is delivered from the IEDs can be forwarded to the IoT platform. On receiving this information from the IED-DG, the *Edge Connector* proceeds

[3]https://libiec61850.com/libiec61850/
[4]https://github.com/ChargePoint/pydnp3

```
{
  "configuracao": {
    "IED": [
      {
        "id": "1",
        "device_ip": "10.7.229.211",
        "entity_type": "IEC61850",
        "device_id": "IED01",
        "entity_name": "IED01",
        "endereco_fiware": "10.7.229.35",
        "mqtt_topico_fiware": "/4jggokgpepnvsb2uv4s40d59ov/IED01/attrs",
        "url": "http://10.7.229.35:4041/iot/devices/IED01",
        "fiware-servicepath": "/",
        "fiware-service": "openiot",
        "Content-Type": "application/json",
        "protocol": "PDI-IoTA-UltraLight",
        "timezone": "America/Fortaleza",
        "transport": "MQTT"
      },
      {
        "id": "2",
        "device_ip": "10.7.229.210",
        "entity_type": "DNP3",
        "device_id": "IED02",
        "entity_name": "IED02",
        "endereco_fiware": "10.7.229.35",
        "mqtt_topico_fiware": "/4jggokgpepnvsb2uv4s40d59ov/IED02/attrs",
        "url": "http://10.7.229.35:4041/iot/devices/IED02",
        "fiware-servicepath": "/",
        "fiware-service": "openiot",
        "Content-Type": "application/json",
        "protocol": "PDI-IoTA-UltraLight",
        "timezone": "America/Fortaleza",
        "transport": "MQTT"
      }
    ]
  }
}
```


Figure 2. Example of an *IED Specs* configuration file in JSON format

to translate the format of the targeted IoT platform and then forwards it. The *Edge Connector* is the module that sets off an interaction with the IoT platform (FIWARE in the case of this article), creates and adds IED agents in the Cloud, publishes data from a device atop applications, and updates the IED properties accordingly. It also acts as a message broker by providing storage and data processing and allowing local access when a connection is not made up with the Cloud.

The IED-CT component receives an *IED Template* file, which contains the hierarchical structure of the IED-LP Agents. It interprets the *IED Template* and then performs the initial set of validations, creating the *IEDSpecs* and sending the settings to the IED-M. The IED-CT has a WebUI, which enables management tasks to be carried out, such as uploading configuration files or the *IED Template* and sending the *IEDSpecs* to the IED-M address. Figure 3 shows the home page of the WebUI of the IED-CT admin interface.

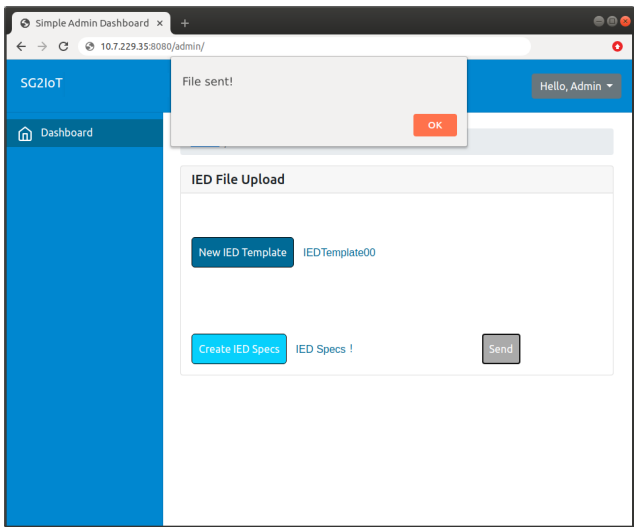


Figure 3. IED Configuration Tool *WebUI* - Admin Interface

The SG2IoT solution considers internal interfaces that aim to allow interoperation between the internal components of the functional architecture and external ones that are intended to expose their functionality to SG applications or receive a response from a request made. It is essential to highlight that, in this research, the internal and external interfaces do not follow a standard recommendation; they were implemented to carry out proof of concept of the functional architecture and its functionalities.

The design of the internal interfaces is based on the Python library *websockets*[5]. The internal interfaces have the following interactions defined: the *IED-M* component has an open interface with *IEC-CT* for receiving *IEDSpecs* on port 5000/TCP; the *IEC 61850 Agent* has port 102/TCP exposed to monitor messages from IEC 61850 type IEDs and port 20000/TCP for DNP3 protocol messages, and the *Edge Connector* uses the FIWARE RESTful API to use its services. This module has port 5015/TCP exposed to receive confirmations of requests made and for local access to data from the IEDs.

Once a DNP3 server is instantiated, it simulates an IED by generating messages from the referred protocol to simulate a communication environment of the IEDs that have been connected to a substation network. The DNP3 server (outstation) was configured to simulate DNP3 objects where the outstation sends an AnalogInput to the master, defining it as being from group 32, variation five, and indexed. Groups provide a means of classifying the type or types of data in a DNP3 message. The DNP3 object group number 32 is assigned to read a changed analog value at the outstation with variation five identifying 32-bit float type numbers. Thus, it was configured so that the DNP3 server supplies voltage magnitudes and values with a 32-bit float format to its client, the *IED-LP Agent*.

The WebUI main page has a button (*New IED Template*) whose function is to open a window that allows one to write the technical specifications of the targeted IEDs and upload the *IEDTemplate*. Then, through the button, *Create IEDSpecs* the *IEDSpecs* are created from the IED Template. Finally, this interface allows the *IEDSpecs* to be sent to the SG2IoT through the *Send* button.

In compliance with the DNP3 and IEC 61850 standards, the SG2IoT actuates as the client that consumes data from various active IEDs (i.e., servers). Among other features, the SG2IoT performs device management and monitoring and thus makes it possible to track IEDs through dashboards. Figure 4 shows a diagram for the workflow sequence of this functionality.

As shown in Figure 4, the device creation procedures start in the IED by broadcasting metering data. On capturing incoming metering data, the IED-LP Agent (the module in SG2IoT responsible for interpreting and converting the incoming data) then forwards the data to the *Data Handler*, which inspects the incoming packet and forwards it to the *Edge Connector* for making a connection to the FIWARE IoT platform. As soon as the connection has been made, the *Edge Connector* invokes the IoT platform to set up a new IED device. Once the device has been created, the IoT platform becomes enabled to receive IED metering data and consequently notifies external software applications (denoted as IoT applications) that subscribed to it (e.g., a given SCADA system). It is worth raising that the subscription of IoT appli-

[5]https://websockets.readthedocs.io/en/stable/intro.html

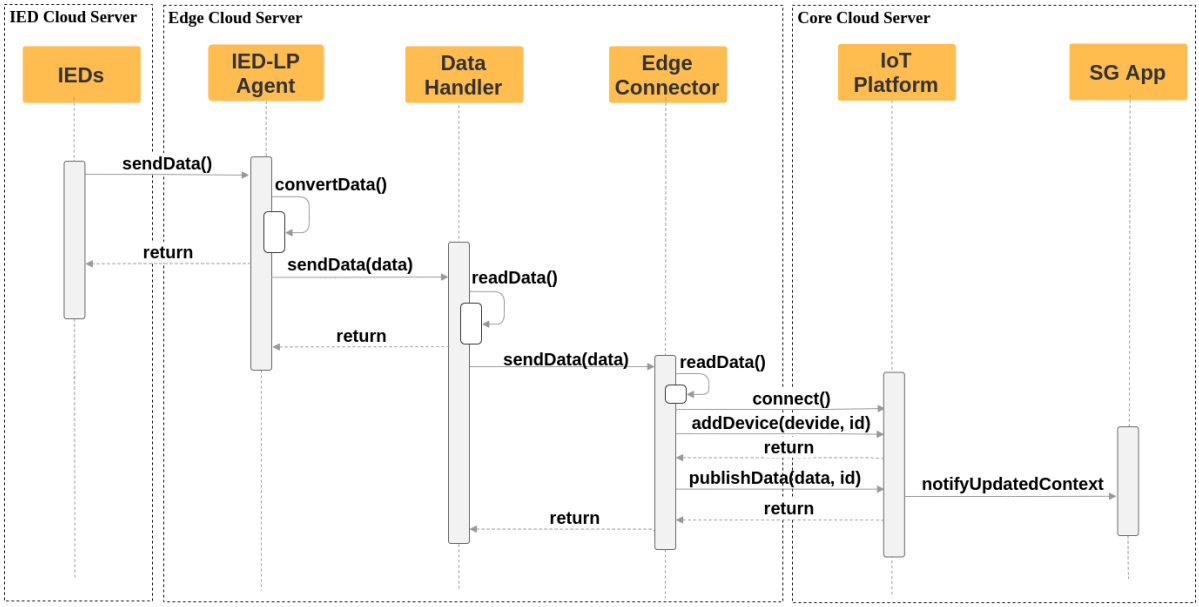


Figure 4. Sequence diagram for creating a device in the SG2IoT

cation is made directly with the IoT platform, being out of the scope of the SG2IoT architecture.

The code repository of the SG2IoT modules is available on GitLab Modesto (2021). An adapted version of the *libIEC61850* and *pydnp3* libraries were used to implement the traffic generators of the DNP3 and IEC 61850 protocols, was also made available in the repository.

5 Prototype Testing and Evaluation

A testbed has been established at the UFRN REGINA-Lab facility to evaluate the SG2IoT system. The testbed comprises real-world devices and technologies that create an environment that comply with the SG-Cloud-IoT approach. In addition to the SG2IoT prototype, an SG monitoring application was made up for running atop the IoT platform so that it can provide an environment suitable for real-time smart metering use cases. The SG monitoring application harnesses the following FIWARE GEs: (i) the Orion Context Broker to settle updates, queries, or subscribe to changes on IED context (i.e., telemetries) publications; (ii) the IDAS GE, which offers IoT Agents to interface with IEDs through MQTT and HTTP; (iii) the Cygnus GE, providing the means IED context history management; and, (iv) MySQL and MongoDB databases for keeping IEC context knowledge.

The IEDs follow a synthetic dataset strategy (given our impossibility to acquire real SG IEDs, which are extremely expensive) to allow legacy IED behavior. Having said this, we implemented applications capable of generating IED legacy SG protocol (DNP3 and IEC 61850) data traffic as contextual entities in the FIWARE NGSI information model. The IED traffic generator application subscribes to the Orion Context Broker so they can have their rate of consumption measured by the smart energy metering applications in real time.

The entire system is virtualized in a Cloud computing environment available on the REGINA-Lab premises. The Cloud environment entails a Dell PowerEdge R740 server, featuring an Intel(R) Xeon(R) Silver 4110 CPU @2.10 GHz processor with 64 GB RAM specifications and 4TB storage. Four (4) virtual machines are set in the Cloud using a hypervisor VM monitor on top of the Ubuntu 18.04 LTS operating system. Each VMs features a virtual Ethernet link in bridge mode to interconnect the VMs while allowing a high-density message rate to be sent simultaneously from multiple IEDs. The remainder of testbed components (namely SG2IoT prototype, SG monitoring application, and IEDs traffic generators) are deployed in Docker containers that run inside a VM accordingly. This scheme allows the density of IEDs to be scaled up flexibly while enabling complete control of CPU and RAM resource allocation, all at running time. Figure 5 depicts the layout of the testbed and its main components in a higher-level view.

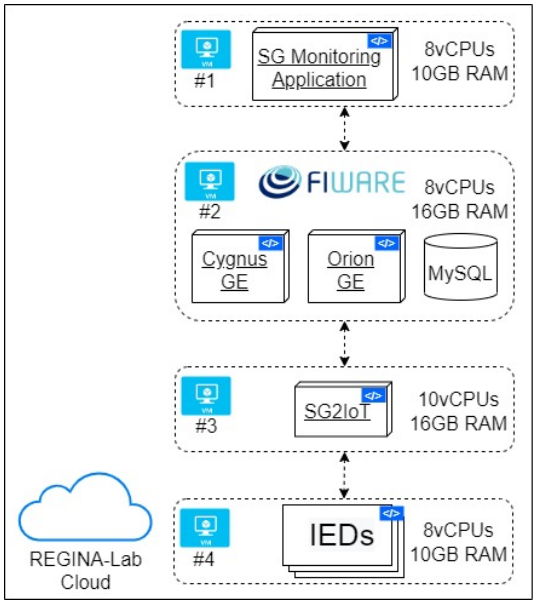


Figure 5. Testbed layout at the REGINA-Lab premises

As Figure 5 presents, the execution of the different IEDs runs at VM4 that is set with eight vCPUs and 10 GB RAM. VM3 executes the SG2IoT prototype, being allocated with

ten vCPUs and 16 GB RAM. The VM2, in turn, is provisioned with eight vCPUs and 16 GB RAM for the IoT platform employing the FIWARE GEs adopted along with the databases. Finally, VM1 employs the container running the SG monitoring application.

The role of the IEC 61850 and DNP3 traffic generators is in simulating electrical parameters compatible with the IEDs, such as phase voltages and currents, date and time, active and reactive powers, and frequency, among others. It was subjected to several stress levels to estimate the maximum capacity of the prototype; these levels were caused by the increase in the scale of simultaneously activated IEDs traffic generator instances on the REGINA-Lab testbed. The maximum density of IEDs in the testbed was 100 simultaneously active when measured in a container restricted to 2 GB and two vCPU (representing a Raspberry Pi-like SoC facility). Any value above this makes the container (in which the SG2IoT was running) restart because of RAM depletion, thus causing the testbed to become inoperative.

As a case study, four (4) groups of experiments were carried out for each of the two protocols, each differentiated by variations in the density of IEDs (namely 1, 10, 50, and 100 of both IEC 61580 and DNP3 types). Simultaneous threads performing uninterrupted requests (1 message every second per IED) to the SG2IoT for 1 hour were set. The experiments were repeated ten times (each under identical conditions) to obtain a 95% confidence interval.

The first experiment for the first scenario involves subscribing the IoT application to a group of active legacy IEDs (IEC 61850 and DNP3) to obtain real-time measurements when considering the SG2IoT running in a lightweight computing device. To achieve this, we scaled the containers of VM3 with RAM of 2 GB capacity and processing power of 2 vCPU, compatible with the widely used Raspberry Pi SoC device. Figure 6 shows a screenshot of the Web dashboard of the actual smart monitoring IoT application.

The IED monitoring application (using the FIWARE platform) was developed and used as an interface that aims to monitor critical information in real-time, such as IED specific data and information about its location, the data model of a particular IED, GOOSE, MMS, SV, and DNP3 messages and the status of the IEDs, among other information.

The application was developed using Node-RED[6]. This tool was chosen because it provides a simple way to connect hardware devices to APIs and online services, such as API-REST, available in the FIWARE platform and, more specifically, in the *Orion* component. Furthermore, it is possible to run Node-RED on a local computer, on an SoC device such as the Raspberry Pi, or in the Cloud, which makes it a versatile and simple tool.

To create the data from the IEDs, IEC 61850 and DNP3 traffic generators were used, in which the values of some of the IED's electrical parameters and quantities are simulated, such as phase voltages and currents, date and time, active and reactive powers of phases, frequency, among others. It should be noted that the IoT application in real-time gives the measurements that originated from the legacy IEDs because of the integration carried out by the SG2IoT. In determining the ability of SG2IoT to integrate legacy IEDs into the SG-Cloud-IoT holistically, we monitored the average flow produced by the active IEDs in the different experimental runs (as shown in Figure 6).

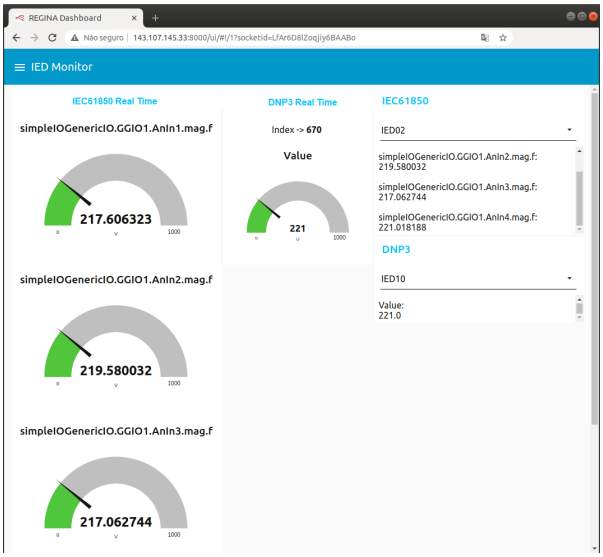


Figure 6. Web dashboard - IED Monitor

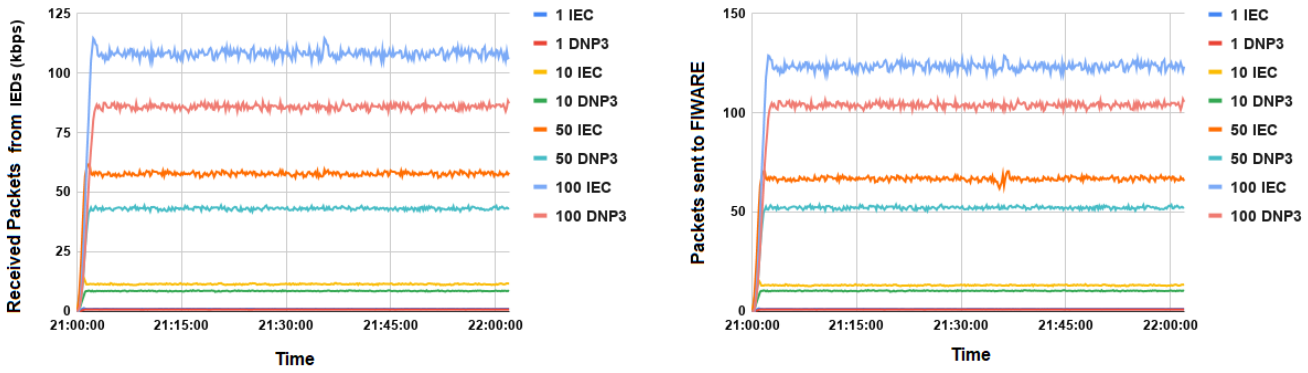
As shown in Figure 7, the SG2IoT prototype is first sent to the packets published by the legacy IEDs in their respective IEC 61850 and DNP3 formats (Figure (7a)). Following this, it translates the incoming data into a coherent, semantic system recognized by the IoT Agent and then forwards the messages to the IoT platform via the MQTT protocol (Figure (7b)). This proof-of-concept demonstrates that the SG2IoT system is functional, that its components conform to the established architecture, and that the internal and external interfaces of the architecture have been successfully validated. Therefore, the results suggest that the effectiveness of the SG2IoT in holistically integrating the legacy IEDs with the SG-Cloud-IoT is confirmed.

Since the ecosystem in question offers the prospect of automating and aligning the connection of the IEDs of different legacy protocols to the SG-Cloud-IoT, a second experiment for the first scenario was conducted to determine its performance by assessing the number of resources consumed by the prototype on the testbed. This type of assessment is critical, as it seeks to ensure that the solution in question works according to plan and, thus, determines whether there is any point that needs improvement. This was discovered by measuring the performance of the prototype concerning CPU utilization and RAM usage behavior in different scenarios while varying the density of active IEDs instances on the testbed. These metrics were obtained employing using the Netdata[7] monitoring tool. In addition, the response time of the prototype in the experiments was analyzed to estimate its scalability. To achieve this, we implemented functions directly in the SG2IoT prototype that records the instant time of execution and completion of each task.

Table 2 shows the average RAM consumption behavior that the SG2IoT solution takes at many stress levels. The information in Table 2 is summarized in Figure 8, which de-

[6]https://nodered.org/

[7]http://www.netdata.cloud/



(a) Data received by SG2IoT from the IEDs

(b) Data sent by the SG2IoT to the FIWARE platform

Figure 7. Average data received by the SG2IoT and sent to the FIWARE platform

picts the average rate of RAM consumption in experiments performed with IEDs at different granularity and publishing telemetries following respective SG legacy protocols while taking note of the fact that the testbed was provisioned with 2 GB of RAM and two vCPU for restricted capabilities (compatible with tiny and affordable hardware architectures, available at low-cost).

As shown in Figure 8, the addition of new IEDs instances impacted an increase in RAM consumption, as expected. The RAM consumption for the IEC 61850 and DNP3 protocols is practically at the behavior of the same rate. According to Figure 8, the RAM consumption is almost identical for both protocols in experiments with 1, 10, and 50 IEDs, and there is a low variation of 15.10 MB in the experiment with 100 IEDs (leading to 0.77% of increasing rate). Moreover, adding a new device to the solution impacts an average increase of 20 MB per monitored IED. Concerning CPU consumption impact, Table 3 shows the outcomes of the tests. The information in Table 3 can be summarized in Figure 9, which provides the averaging CPU consumption (%) during the monitored period. Figure 9 confirms that CPU consumption is also directly dependent on the number of IEDs monitored by the SG2IoT. In other words, the processing performance of SG2IoT degrades exponentially with the increasing amount of IED instances activation in the testbed.

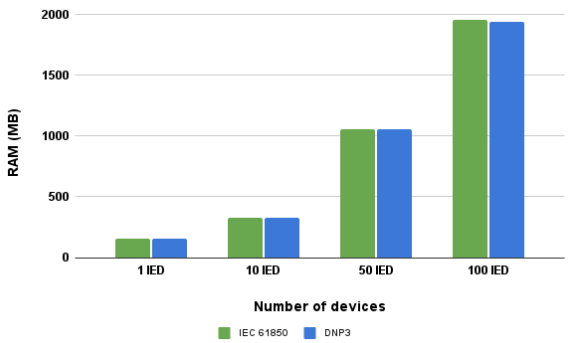


Figure 8. RAM consumption by the SG2IoT - First Scenario

It is clear from the outcomes presented in Figure 9 that the SG2IoT raised the same percentage of CPU consumption in practically all the experiments (1, 10, 50, and 100 IEDs)

Table 2. RAM consumption - First Scenario

SG2IoT	1 IED	10 IEDs	50 IEDs	100 IEDs
IEC 61850 (MB)	155.58	325.19	1.059.53	1,957.38
DNP3 (MB)	155.42	324.48	1,055.19	1,942.28

Table 3. CPU utilization - First Scenario

SG2IoT	1 IED	10 IEDs	50 IEDs	100 IEDs
IEC 61850 (%)	3.41	11.06	48.32	77.43
DNP3 (%)	3.36	11.20	48.43	77.34

when conducted under the same conditions. The CPU utilization behavior was proven to be low (approximately 0.9%, on average, for each monitored IED) and did not use the entire CPU required for the solution - i.e., two vCPUs were used when forming the container in which the solution was obtained. Hence, the results suggest that the SG2IoT approach yields a lightweight solution since the RAM consumption did not exceed the memory threshold (2 GB, compatible with tiny and affordable computing devices, e.g., the widely used Raspberry Pi SoC prototyping solution) that was set when creating the container in which the solution was obtained.

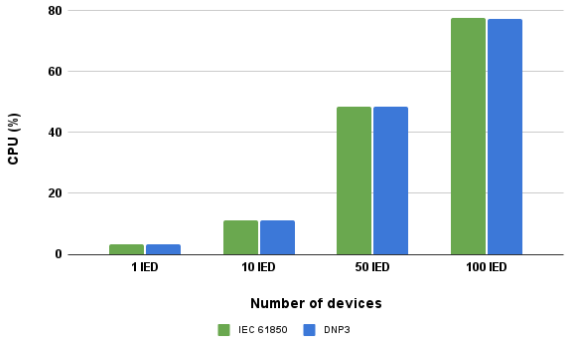


Figure 9. CPU utilization by the SG2IoT - First Scenario

A new round of computational tests was conducted to assess the performance impact of the SG2IoT in a second scenario where the RAM and CPU resources were scaled up to determine the pattern of behavior of the solution when there were variations of these metrics on a larger scale pattern, along with enhanced computational power. For this rea-

son, the settings used in the Docker containers in which the SG2IoT was deployed were set to 8 GB and eight vCPUs (compatible with an affordable desktop). The same test performed in the first scenario was repeated for this second scenario, with a difference in the variation of monitored IEDs, which, this time, ranged between 1, 10, 50, 100, 200, 300, and 400 IEDs density, and involved uninterrupted requests (1 message per IED every second) during 1 hour for the SG2IoT solution.

Table 4 presents the outcome results for the RAM consumption based on the various stress levels for the two types of SG legacy protocols used. The information in Table 4 is summarized in Figure 10, which shows the average consumption of RAM (MB) during the monitored period.

Figure 10 exhibits that the rate in RAM consumption at the second scenario behaved similarly to that of the first scenario, with a difference between the two scenarios of 1.20% of RAM consumption on average on the IEC 61850 IEDs and a difference of 1.42% of RAM consumption in average on the DNP3 IEDs. The rate of increase in RAM consumption (see Figure 8) with the addition of new IEDs was similar to that of the first scenario (20 MB per IED), the second scenario, in turn, impacted an average increase of 19 MB per IED. With the data currently available it can be assumed that the consumption of RAM is directly dependent on the number of IEDs monitored by the SG2IoT. Although in the second scenario, the SG2IoT has more CPUs, this metric did not influence the RAM consumption. However, to prove this theory, further work is needed to be carried out to evaluate how the SG2IoT will behave in a scenario with the same number of IEDs but with a higher workload.

It can be concluded that the addition of a new active publishing instance to the system results in a very similar increase of RAM on average per IED in both scenarios. Finally, the RAM consumption did not exceed the memory threshold (8 GB) set when the solution's container was set. Therefore, the results suggest good prospects in environments where the amount of RAM resources available is not a problem.

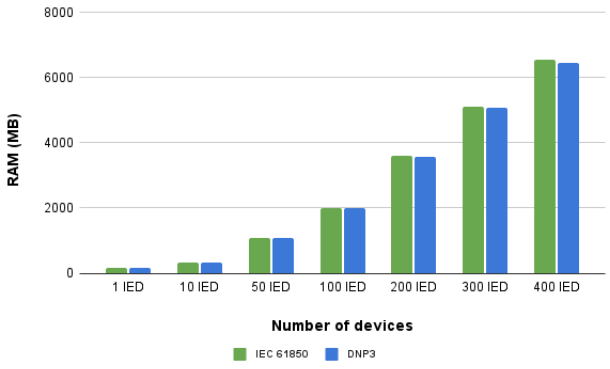


Figure 10. RAM consumption by the SG2IoT - Second Scenario

Concerning the use of CPU in the second scenario, Table 5 shows the results of the tests performed. The information in Table 5 is summarized in Figure 11, which shows the average use of CPU (%) during the monitored period.

Figure 11 confirms that CPU consumption is also directly dependent on the number of monitored IEDs, as seen in the first scenario (approximately 0.9%, Figure 9). The percentage of CPU utilization was similar for the two protocols when measured in the same experimental conditions (approximately 0.2%, on average, for each monitored IED) in the second scenario. Even with the increase of new IEDs monitored by the SG2IoT (at 300% rate), the total processing power did not surpass the eight vCPUs required for the solution and remained below 40% of its total capacity. Also, it can be noted that the values obtained for CPU utilization in the second scenario decreased by approximately 75% on average compared to the first scenario. These results show that adding more computational power influences the processing capacity of the SG2IoT.

The tests carried out in the second scenario aimed to determine how the solution would behave in environments with no excellent restriction on the use of computational power. In the tests, the SG2IoT solution was executed with the use of RAM limited to 8 GB and the use of 8 vCPUs, which suggests that the SG2IoT solution can run smoothly in personal computer architecture to handle a domain enclosing 400 IEDs simultaneously active. In high-dense IED environments, SG2IoT can be deployable in an edge server running in bare metal mode, solely designed for this purpose. In conclusion, it can be stated from the outcomes of the second scenario that a medium-scale computing architecture allows the SG2IoT to support many more than 400 IEDs simultaneously, while the CPU consumption was also within acceptable limits.

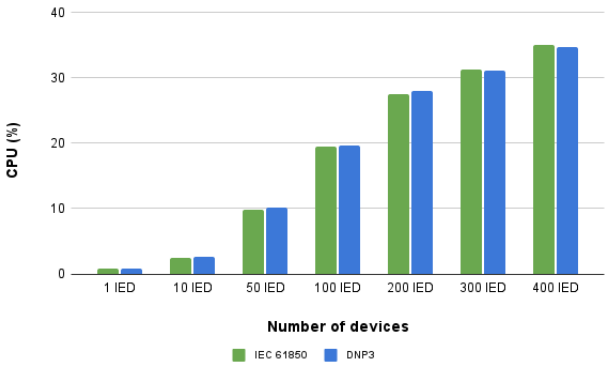


Figure 11. CPU utilization by the SG2IoT - Second Scenario

Finally, but not least, Figures 12 and 13 come up with the average response times that the SG2IoT prototype achieved in the experiments. The end-to-end response time was measured by calculating the difference in time from the moment when the IED sender places the data content at the top of its transmission stack until the moment the receiver (FIWARE, through Orion Context Broker) extracts the data from its transmission stack and confirms the information that has been recorded. A locally available NTP server synchronized the testbed machines' clocks. The response times were divided into two parts: (i) SG2IoT, average response time to receive IEDs messages in legacy SG format, convert them to MQTT, and then forward them to the IoT platform; and (ii) FIWARE, the average response time of the IoT platform that is needed to receive the IED measurements and operate the intelligent energy measurement system.

As it can be observed in Figure 12, the SG2IoT time behaved within a relatively stable response time for handling

Table 4. RAM consumption - Second Scenario

SG2IoT	1 IED	10 IEDs	50 IEDs	100 IEDs	200 IEDs	300 IEDs	400 IEDs
IEC 61850 (MB)	155.91	330.63	1,074.90	1,987.94	3,587.46	5,110.78	6,542.43
DNP3 (MB)	155.70	328.89	1,073.27	1,991.74	3,559.60	5,073.56	6,448.47

Table 5. CPU utilization - Second Scenario

SG2IoT	1 IED	10 IEDs	50 IEDs	100 IEDs	200 IEDs	300 IEDs	400 IEDs
IEC 61850 (%)	0.84	2.49	9.85	19.55	27.54	31.25	34.99
DNP3 (%)	0.75	2.59	10.23	19.69	27.95	31.10	34.75

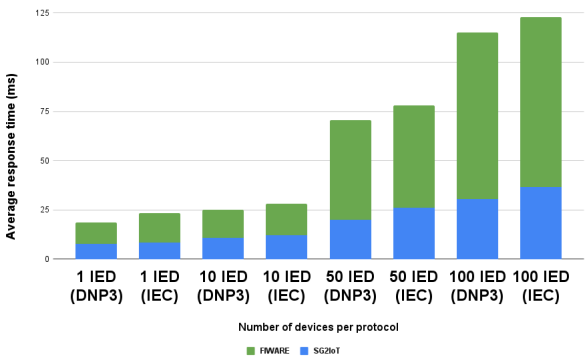


Figure 12. Average response time by the SG2IoT - First Scenario

messages from different IEDs in the first scenario (with a growth rate of 1.4% for the IEC 61850 IEDs and 1.3% for the DNP3 IEDs density settings of each experiment). On the other side, the FIWARE platform (on average 64.88%) yielded a much longer response time than the SG2IoT, with an exponential growth rate as more IEDs are activated, 1.7% and 2% for the IEC 61850 and DNP3 protocols respectively. The high-complex architecture explains this difference (almost two-fold) that the FIWARE GEs (interworking components and database handling operations) impact in the testbed concerning the lightweight SG2IoT approach.

Table 6 shows the SG2IoT and FIWARE average response times rates taken to handle both DNP3 and IEC 61850 protocol messages. Figure 13 summarizes Table 6 and it shows that in the second scenario, the average response time for both legacy SG protocols remained relatively constant during the experiments with 1, 10, 50 and 100 IEDs. There were no significant variations in the experiments with 1, 10, 50, and 100 IEDs when compared with the first scenario. Outcomes reveal that they obtained a similar growth rate, 1.3% for both protocols for the SG2IoT time, whereas it obtained 1.6% and 2% for the FIWARE time to deal with the IEC 61850 and DNP3 protocol messages, respectively.

On the other side, the executions enclosing 200, 300, and 400 active IEDs simultaneously yielded a significant increase in the average response time, by reaching an average of 1,909.40 ms, 3,320.24 ms and 3,487.79 ms behavior for the execution with 200, 300 and 400 IEDs (IEC) respectively. Even in the case of the experiments with a large volume of messages (200, 300, and 400 IEDs), the SG2IoT prototype allowed a relatively stable response time behavior for handling the messages, whereas the FIWARE platform impacted a longer response time (84.87% higher than SG2IoT). Concerning the execution with 400 IEDs, the growth rate behavior of the FIWARE time (1.45% and 1.30%) was two-fold than the growth rate that the SG2IoT time (0.62% and 0.68%) impacted to handle IEC 61850 and DNP3 protocol messages respectively.

The explanation for all these outcomes lies in the architectural complexity of the FIWARE component stack, along with heavyweight workflows employed concerning the SG2IoT solution. This leads us to conclude that in high-dense IEDs scenarios, the FIWARE IoT Agent tends to become a bottleneck processing, thus making Orion delaying to notify the receipt of messages to the SG2IoT in the testbed. Outcomes of existing performance tests conducted on several modules of the FIWARE platform Araujo *et al*. (2019) come up with its main limitations while being a bottleneck. Our outcomes suggest that FIWARE did not impact performance degradation by being a bottleneck in our testbed. Despite this, the results suggest that the prototype was following the standards required by SG applications - 200 ms according to Kuzlu and Manisa (2013). Moreover, SG2IoT was able to meet the performance requirements of the average response time - 500 ms as laid down by the IEC 61850 standard IEC61850 (2020).

Table 6. Average response time - Second Scenario

SG2IoT IEC 61850 DNP3	SG2IoT (ms)	FIWARE (ms)	SG2IoT + FIWARE (ms)
1 IED (DNP3)	5.66	15.40	21.06
1 IED (IEC)	5.41	10.83	16.24
10 IED (DNP3)	8.71	15.54	24.25
10 IED (IEC)	9.29	19.69	28.98
50 IED (DNP3)	9.03	56.31	65.34
50 IED (IEC)	7.15	51.47	58.62
100 IED (DNP3)	22.52	80.90	103.42
100 IED (IEC)	20.75	81.73	102.48
200 IED (DNP3)	51.74	1,511.24	1,562.98
200 IED (IEC)	59.9	1,849.51	1,909.41
300 IED (DNP3)	60.37	2,519.75	2,580.12
300 IED (IEC)	58.04	3,262.20	3,320.24
400 IED (DNP3)	85.85	2,597.08	2,682.93
400 IED (IEC)	66.46	3,421.33	3,487.79

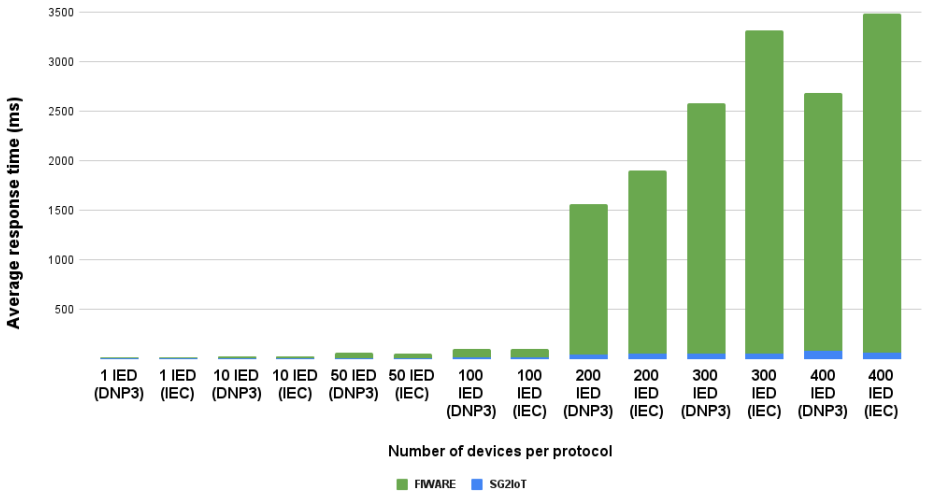


Figure 13. Average response time by the SG2IoT - Second Scenario

6 Conclusion and Recommendations for Future Work

This article introduced the SG2IoT, which establishes an SG-Cloud-IoT ecosystem and automates the integration of legacy IEDs protocols through modular architectural components that run in distributed nodes of the Cloud (including Edge nodes). The SG2IoT provides a flexible, monitorable, and automated infrastructure environment to accommodate new IoT applications and services atop the SG vertical. As a proof-of-concept of the proposed solution, a prototype was built on an actual testbed compatible with the SG-Cloud-IoT. The tests suggest the effectiveness of SG2IoT in integrating legacy IEDs to test SG-Cloud-IoT when there is low - computational cost estimation.

The results obtained for the computational cost assessment show that although the SG2IoT can support scenarios with an increasing density of IEDs, it does not support a large number of simultaneous IEDs when it is instantiated in an environment where computing power is restricted (first scenario). Almost all computing resources are utilized. Because of this, it is particularly recommended for use in experimental environments or for environments that do not have a high demand from concurrent devices and requests. The SG2IoT supports many devices for environments with no processing and memory constraints (second scenario).

Aside from that, the response time achieved in the experiments also offers a reasonable level of scalability when the SG2IoT response time is taken into account in both scenarios. It represented an average response time within acceptable limits for SG systems, following the recommendations laid down in IEC 61850 IEC61850 (2020) and by Kuzlu and Manisa (2013).

In future work, we plan to implement and map more functions from the IEC 61850 protocol standard and DNP3. Also, integrating it with other legacy SG protocols such as Modbus and increasing the use of other IoT communication protocols such as CoAP. In addition, we intend to integrate and evaluate SG2IoT with other IoT platforms to identify inferences or significant changes and assess the performance of the SG2IoT in this integration, aiming to gain a broader view and find areas of improvement for the solution. Finally, another possibility for future work is the implementation of new Smart Grid applications to help validate the use of the SG2IoT tool in different Cloud-IoT scenarios, as a large number of real-world applications making use of the solution can help to identify points to improve and problems that have not yet been detected.

Acknowledgement

This study was financed by the R&D project entitled “Sistema IoT-Cloud de Medição Centralizada de Energia Voltado a Rede CEA - 001/2021 e 50/2022”, and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

Araújo, P. R. C., Filho, R. H., Rodrigues, J. J., Oliveira, J. P., and Braga, S. A. (2018). Middleware for integration of legacy electrical equipment into smart grid infrastructure using wireless sensor networks. *International Journal of Communication Systems*, 31(1):e3380. DOI: 10.1002/dac.3380.

Araujo, V., Mitra, K., Saguna, S., and Åhlund, C. (2019). Performance evaluation of fiware: A cloud-based iot platform for smart cities. *Journal of Parallel and Distributed Computing*, 132:250–261. DOI: 10.1016/j.jpdc.2018.1.

Barja-Martinez, S., Aragüés-Peñalba, M., Munné-Collado, Í., Lloret-Gallego, P., Bullich-Massagué, E., and Villafafila-Robles, R. (2021). Artificial intelligence techniques for enabling big data services in distribution networks: A review. *Renewable and Sustainable Energy Reviews*, 150:111459. DOI: 10.1016/j.rser.2021.111459.

Boakye-Boateng, K., Siahaan, I. S., Al Muktadir, A. H., Xu, D., and Ghorbani, A. A. (2021). Sniffing serial-based substation devices: A complement to security-centric data

collection. In *IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–6. IEEE. DOI: 10.1109/ISGTEurope52324.2021.9640212.

Cheruvu, S., Kumar, A., Smith, N., and Wheeler, D. M. (2020). Iot frameworks and complexity. *Demystifying Internet of Things Security*, pages 23–148. DOI: 10.1007/978-1-4842-2896-8_2.

Farri, E. and Ayubi, P. (2022). A robust digital video watermarking based on ct-svd domain and chaotic dna sequences for copyright protection. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–25. DOI: 10.1007/s12652-022-03771-7.

FIWARE-CATALOGUE (2019). Fiware catalogue. Available at: `https://www.fiware.org/developers/catalogue/`.

FIWARE-CYGNUS (2019). Cygnus. Available at: ://fiware-cygnus.readthedocs.io/en/latest/.

FIWARE-IoTAgent (2019). Fiware iot agent. Available at: `https://iot-platform-docs.readthedocs.io/en/latest/device_gateway/index.html`.

FIWARE-NGSI (2018). Fiware-ngsi v2 specification. Available at: `http://fiware.github.io/specifications/ngsiv2/stable/`.

FIWARE-ORION (2019). Orion context broker. Available at: `https://fiware-orion.readthedocs.io/en/master/`.

fiware.org (2022). About fiware. Available at: `https://www.fiware.org/about-us/`.

IEC61850 (2020). International electrotechnical commission standard - iec61850. Available at: `https://www.iec.ch/smartgrid/standards/`.

IEEE (2012). 1815-2012 - ieee standard for electric power systems communications-distributed network protocol (dnp3). Available at: `https://standards.ieee.org/content/ieee-standards/en/standard/1815-2012.html`.

Kuzlu, M. and Manisa, P. (2013). Assessment of communication technologies and network requirements for different smart grid applications. In *IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6. DOI: 10.1109/ISGT.2013.6497873.

Modesto, W. (2021). Legacy smart grid to iot protocol integration approach (sg2iot). Available at: `https://gitlab.com/wanmdt/sg2iot`.

Modesto, W., Neto, A. V., Rosário, D., and Cerqueira, E. (2021). SG2IoT - An Architecture for Integration of Intelligent Electrical Devices with Legacy Approach in IoT-Based Smart Grid Systems. In *13th Brazilian Symposium on Ubiquitous and Pervasive Computing (SBCUP)*, pages 31–40, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbcup.2021.16001.

Mota, R., Riker, A., and Rosário, D. (2019). Adjusting group communication in dense internet of things networks with heterogeneous energy sources. In *11th Brazilian Symposium on Ubiquitous and Pervasive Computing (SBCUP)*, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbcup.2019.6594.

Negash, B., Rahmani, A. M., Liljeberg, P., and Jantsch, A. (2018). *Fog Computing Fundamentals in the Internet-of-Things*, pages 3–13. Springer International Publishing. DOI: 10.1007/978-3-319-57639-8_1.

Njova, D., Ogudo, K., and Umenne, P. (2020). Packet analysis of dnp3 protocol over tcp/ip at an electrical substation grid modelled in opnet. In *IEEE PES/IAS PowerAfrica*, pages 1–5. IEEE. DOI: 10.1109/PowerAfrica49420.2020.9219968.

Nugur, A., Pipattanasomporn, M., Kuzlu, M., and Rahman, S. (2019). Design and development of an iot gateway for smart building applications. *IEEE Internet of Things Journal*, 6(5):9020–9029. DOI: 10.1109/JIOT.2018.2885652.

opendnp3 (2020). dnp3/opendnp3. Available at: `https://github.com/dnp3/opendnp3`.

Pfeiff, G., Araújo, F., Oliveira, H., Rosário, D., and Cerqueira, E. (2020). Modelo de detecção de fraudes elétricas baseado em aprendizado de máquina. In *Brazilian Symposium on Ubiquitous and Pervasive Computing (SBCUP)*, pages 51–60. SBC. DOI: 10.5753/sbcup.2020.11211.

Ramalho, F. and Neto, A. (2016). Virtualization at the network edge: A performance comparison. In *IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. DOI: 10.1109/WoWMoM.2016.7523584.

Rouhani, R., Sadeghkhani, I., and Guerrero, J. M. (2020). Directional element for faulty feeder identification of high-resistance fault in high-surety power supply systems. *IET Generation, Transmission & Distribution*. DOI: 10.1049/gtd2.12006.

Saleem, Y., Crespi, N., Rehmani, M. H., and Copeland, R. (2019). Internet of things-aided smart grid: technologies, architectures, applications, prototypes, and future research directions. *IEEE Access*, 7:62962–63003. DOI: 10.1109/ACCESS.2019.2913984.

Saxena, N., Roy, A., and Kim, H. (2017). Efficient 5g small cell planning with embms for optimal demand response in smart grids. *IEEE Transactions on Industrial Informatics*, 13(3):1471–1481. DOI: 10.1109/TII.2017.26811051.

Shin, I.-J., Song, B.-K., and Eom, D.-S. (2017). International electronical committee (iec) 61850 mapping with constrained application protocol (coap) in smart grids based european telecommunications standard institute machine-to-machine (m2m) environment. *Energies*, 10(3):393. DOI: 10.3390/en10030393.

Silva, H., Neto, A., Cerqueira, E., Dantas, F., Barros, H., and Almeida, E. (2013). Advanced communication system for rich and green smart grid networking. In *IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*, pages 1–4. DOI: 10.1109/ISGT-LA.2013.6554413.

Tightiz, L. and Yang, H. (2020). A comprehensive review on iot protocols' features in smart grid communication. *Energies*, 13(11):2762. DOI: 10.3390/en13112762.