On Private Server Implementations and Data Visualization for LoRaWAN

Sheikh Tareq Ahmed
Prairie View A&M University
Prairie View, Texas, USA
sahmed13@pvamu.edu

Annamalai Annamali Jr
Prairieview A&M University
Prairie View, Texas, USA
annamalai@pvamu.edu

Abstract—**The paper titled "On Private Server Implementations and Data Visualization for LoRaWAN" explores the implementation of private LoRaWAN servers for secure and scalable data transmission in Internet of Things (IoT) applications. The authors present a detailed analysis of the existing server implementations and propose a novel architecture for private server deployment. The paper also addresses the challenges associated with data visualization in LoRaWAN networks and presents a comprehensive solution for efficient data representation using advanced visualization techniques. The proposed implementation is evaluated through experiments and simulations, and the results demonstrate the feasibility and effectiveness of the proposed architecture for secure and scalable IoT applications. The paper concludes with a discussion on the future directions for research in this area, highlighting the need for further investigation on the scalability and security of LoRaWAN networks.**

Keywords—TTN, Grafana, IOT, LoRaWAN

I. Introduction

LoRaWAN is getting popular day by day in the field of internet of things (IOT) due to its low powered and long-range nature. The LoRaWAN technology has been around for a while as one of the LPWAN technologies, which at the very least include end nodes, a gateway, a network server and an application server. LoRaWAN may be used for a wide range of purposes, including tracking, agriculture, precision farming, senior care, home security, and water and air pollution monitoring. A lot of implementations have been done on by using LoRaWAN gateway Public Server.[1]. But public servers are not opensource and suffers from several security issues. An obvious choice of alternative is opensource private servers and many more data visualization tools that can be made and set up to send and receive sensor data from Lora WAN devices [2]. But setting up theses servers and visualization tools is often challenging due to their different features they offer over each other.
In this paper, we have represented setting up a private server of thing stack V3 and a IOT data visualization tools Grafana and their pros and cons over each other. These empirical studies will help the community to make a choice in selecting the deployment of servers on their projects. Our goal is to investigate Efficacy of different private server implications (in terms of ease of set up, security, features etc.) and data visualization for the low powered long-range devices.

II. PRIOR WORK

LoRaWAN technology was developed and first deployed in 2012.Many network service providers, such as Comcast in the US, KPN in Europe, and SK Telecom in Asia, offer infrastructure for using the technology. There is also a non-commercial, public LoRaWAN service provider called The Things Network. Unlike Sigfox, another Low-Power Wide-Area Network (LPWAN) technology, anyone can build and deploy their own LoRaWAN gateways and servers, which is a significant advantage of using LoRaWAN. [3]

III. OVERVIEW

The Long-Range Wide Area Networks a Low Power, LPWAN protocol designed to link battery-operated IoT devices wirelessly to the Web where key requirements of the Internet of Things devices are such as bi-directional communication, secure communication, availability, and localized services. In the real-life scenario, one or more end devices need to be reachable from the LoRaWAN gateway to transmit the data to the LoRaWAN network server. The main difference from the star topology of LoRaWAN is in mesh networking end devices also can transmit data among themselves. [4]
In Order to successfully transmit data from end devices to the LoRaWAN network server, the end device should be directly connected to the LoRaWAN gateway, or it needs to transmit data to an end device that is reachable from the source and from the LoRaWAN gateway. LoRaWAN server is classified into two kinds: Public and Private. The public server is usually maintained by enterprises and most of the time imposed with Fair User Policy (FUP) restrictions. Moreover, the data is shared here thus resulting in further security issues. For example, the FUP that limits the uplink airtime to thirty seconds per day per end device and the down-link packets to ten packets per day per end-device effects on the Things Network's public community network.

*a) **LoRaWAN***

LoRaWAN is deployed in a star-topology where gateways send data to and from end-devices and key network servers. The gateway enable connection to the servers over a IP and acts as a bridge, simply passing packets between RF link and IP networks. This type of Wireless networs uses the loRa abilities using LoRa Phy layer to initiate a connection between the end device and gateways.

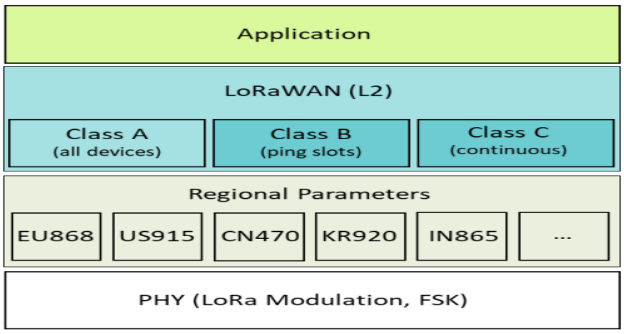

Fig.1. LoRaWAN Architecture [4]

Bi-directional communication is supported in all modes, and multi-address clusters are supported for long-range usage in a well-organized manner during tasks such as OTA firmware updates and bulk packet transfers as well. While the LoRa-WAN design defines the practical employment of LoRa networks, it does not define any viable model such as public, shared, private, or enterprise. It offers the production industry the options to renovate and decide how it is used.[4]

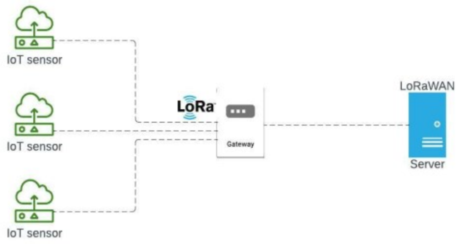

Fig.2.Conventional LoRa-WAN network

i. DEVICES

LoRa-WAN provides three kinds of devices to meet different application criteria. Some classes provide high bandwidth while compromising on device battery life and vice versa.[5]

- A is functional type with the lowermost performance. Up-link packet transfer is possible. Edge devices can remain in low power mode if your running application decides. Network does not require devices to wake-up on periodic interval to keep the packet transmission possible. Following are the specifications of class A device.[5]
- Devices in class B connect to the Gateway at programmed times. While this provides possibility to transmit the downlink in a deterministic manner, it also increases the power consumption of edge-devices. [5]
- In this class down-link is reduces by keeping the device in receiving mode when it is not receiving data. Class C allows the application servers to allow down-link transmissions at any time based on the always open-edge device receiving connection. This reduces downlink latency. However, this is a compromise for Edge device battery life. Therefore, if continuous power is available, Class C should be used.[5]

ii. LoRaWAN REGIONAL PARAMTERS

Lora radio operates in the ISM band, which means it does not require any licenses to operate LoRa radio devices. LoRa radio is like Wi-Fi, while Wi-Fi operates at 2.4 GHz and 5 GHz LoRa operates at different frequencies depending on the region. The restrictions are applied by the Local Government bodies. That's why LoRaWAN has defined regional perimeters to enable LoRaWAN communication worldwide. These specifications are called LoRaWAN regional perimeters.[6]

iii. SPREAD FACTOR

The spread factor is an integer value that ranges from 7 to 12, with 7 corresponding to the highest data rate and 12 corresponding to the lowest data rate. A higher spread factor results in a lower data rate but a longer range, while a lower spread factor results in a higher data rate but a shorter range. The spread factor is often used to optimize the performance of a LoRaWAN communication system for a specific application or environment. [7]

For example, if you need to transmit small amounts of data over long distances with low power consumption, you might choose a higher spread factor to achieve a longer range. On the other hand, if you need to transmit large amounts of data at a high rate, you might choose a lower spread factor to achieve a higher data rate.

In addition to the spread factor, LoRa also uses other parameters, such as the frequency band and the modulation scheme, to optimize the performance of the communication system.

iv. DATARATE

You can set variable data rates between edge devices and gateways. DR selection allows an active compromise between communication range and message duration. With range-spectrum technology, communications with different data rates do not interfere with each other, creating a series of virtual "code" channels that increase gateway capacity. To improve edge device battery life and overall network functionality. LoRaWAN network servers can be configured to dynamically change the data rate and transmit power of each end device. LoRaWAN rates range from 0.3 kbps to 50 kbps. [8]

v. SECURITY

In LoRaWAN (Long Range Wide Area Network), a network session key (NSK) is a cryptographic key that is used to establish a secure connection between a LoRaWAN device and

a) LoRaWAN gateway

It is typically used in combination with a network authentication key (NAK) to provide authentication and encryption for the communication between the device and the gateway. [9]

In LoRaWAN, the NSK is a unique, randomly generated key that is shared between the device and the gateway during the authentication process. It is used to encrypt the communication between the device and the gateway, ensuring that only authorized devices can access the network and that the communication cannot be intercepted or tampered with by an unauthorized party.

b) Private and Public LoRaWAN network

A public LoRa network server is a network server that is accessible to the public and is typically used to connect and communicate with end devices over long distances. Public LoRa network servers are often used in Internet of Things (IoT) applications, such as smart cities, agriculture, and asset tracking.

A private LoRa network server, on the other hand, is a network server that is only accessible to a specific group or organization. Private LoRa network servers are often used for applications that require a higher level of security and control, such as industrial automation and critical infrastructure.

ii) Grafana

Grafana is an open-source data visualization and monitoring platform that is often used in the context of the Internet of Things (IoT). It allows users to create interactive dashboards and charts to visualize and monitor data from various sources, including IoT devices and systems.

One of the key features of Grafana is its support for a wide range of data sources, including time series databases, metrics platforms, and IoT platforms. This makes it easy to integrate Grafana with IoT systems and devices, and to visualize and monitor the data they generate in real-time.

In addition to data visualization and monitoring, Grafana also provides alerting and notification capabilities, allowing users to set up alerts based on specific conditions or thresholds and to receive notifications when these conditions are met. This can be useful for detecting and responding to anomalies or issues in IoT systems and devices.

Grafana is a powerful tool for visualizing and monitoring data from IoT systems and devices and can help users to gain insights into the performance and behavior of these systems and make informed decisions based on the data.[10]

Grafana availability options:

The Grafana is available in different versions which are.

- Grafana cloud.
- Grafana OSS.

a) Grafana cloud

Grafana offers cloud service. It is a subscription-based service where you will be charged according to your usage. On a free cloud account, there are limitations for active members. [11]

b) Grafana OSS:

Since Grafana is open source, you can install it on your own servers or PC. It provides features similar to Grafana cloud.[12]

A) Metrics logs, and traces exploration:

Grafana provide option to explore metrics from database. It helps to develop dashboards and troubleshooting your queries from databases. When you make query from database it provides a visualization of data as line, or bars. You can also have a look into Json data as it is fetched from database. [11]

iii) Thingstack

The ThingStack is open-source LoRaWAN network server. ThingStack is an open-source software platform that enables developers and organizations to build and deploy Internet of Things (IoT) applications using the LoRaWAN protocol. It consists of two main components: the LoRaWAN Network Server and the LoRaWAN Application Server.

The LoRaWAN Application Server is responsible for managing the application-level data and functionality. It provides APIs and tools for developers to build and deploy IoT applications. It can be used to store and process data from devices, integrate with other systems and services, and provide user interfaces for interacting with the data and devices.

ThingStack is designed to be flexible and scalable, so it can be used in a variety of applications and environments. It can be deployed on-premises or in the cloud and can be customized and extended using the available APIs and source code.

a) ThingStack Open-source and enterprise distributions:

You can run ThingStack open-source and enterprise distributions on your own hardware.

The Thing Network provide cloud options as paid service, this way you don't have to install anything on your cloud or local hardware setup. The things-Industries recommends using things-stack cloud or AWS launcher for high availability and throughput.

b) Thing Stack community edition:

Other than cloud and enterprise options you can run Thing Stack server on your own hardware using Thing Stack community edition. Thing Stack community edition is free of cost open source LoRaWAN server. That can run-on low-cost hardware like raspberry Pi.

Thing Stack community edition managed by things network can also be used for deployment. It multi cluster server deployment. You can connect your gateway to local nearby Thing Network cluster depending on location thus reducing overall network latency. These cluster are located in following location:

- Australia (Sydney)
- Europe (Ireland)
- North America (California)

Since Thing Stack is resource intense, you need pretty good hardware specification of server if you are going to deploy it on private server. The TTN describes specification as follow:

- Quad core CPU
- 16GB ram
- DNS server to point communication to your server.
- A license of ThingStack if you're using enterprise edition.

c) deploying on Raspberry Pi

You can deploy ThingStack on raspberry Pi using docker compose. The ThingStack provide git-hub repository to install ThingStack on raspberry Pi. This deployment can run on AMD64 based Pc, ARMv8 like raspberry pi3/4, and ARMv7 like raspberry Pi.

Fig.3. Things Network Cluster locations [14]

To install it on raspberry Pi steps are as follow:

- An operating system installed on host machine.
- Docker
- An image installation software, Balena Etcher.

You can use docker compose .yml file to configure and deploy ThingStack server. It is available on git-hub repository.

The other way to deploy ThingStack server is using Balena Cloud.

IV. Methodology and Experimental Setup

In this experimental setup we are going to use a data visualization tool, Grafana time-series database, InfluxDBand Node-Red. For LoRaWAN server implementation we are going to use ChirpStack.

On hardware side, we have Raspberry Pi-3, to run all our installation. For LoRaWAN gateway we used Dragino LPS8.

For our end-devices microcontroller we used ESP-32 microcontroller-based development board connected to our

LoRa radios. Hope-RF RF-95W is used for connecting our sensor nodes to our LoRaWAN gateway.

Raspberry Pi is a small, affordable, single-board computer developed by the Raspberry Pi Foundation. The Raspberry Pi 3 Model B is the third generation of the Raspberry Pi and was released in February 2016. It has a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, 1 GB of RAM, and integrated 802.11n wireless LAN and Bluetooth 4.1. [12]

The Raspberry Pi 3 is a versatile device that can be used for a wide range of applications.

The LPS8 gateway is a compact, low-power device that is designed for use in outdoor or industrial environments. It can be powered by an external power supply or by a solar panel, making it suitable for remote or off-grid installations.

The LPS8 gateway supports both Class A and Class C devices and can connect to the Internet using Ethernet or Wi-Fi. It also has a built-in GPS module, which allows it to provide location information to connected devices. The LPS8 gateway is designed to be easy to install and configure, and it comes with a web-based configuration interface that allows users to set up and manage their LoRaWAN network quickly and easily. [11]

ESP32 is a series of low-cost, low-power system-on-a-chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. These microcontrollers are designed for a wide range of applications, including home automation, Internet of Things (IoT) devices, and wearable electronics. [15]

HopeRF95w is a wireless transceiver module made by Hope-RF. It is a low-cost, low-power device that operates in the range of frequencies adaptable to specific region. And is designed for use in IoT and M2M (machine-to-machine) applications.

The HopeRF95w module is based on the Semtech SX1276 chip, which supports the LoRaWAN communication protocol. LoRaWAN is a long-range, low-power protocol that is well suited for IoT applications that need to transmit data over long distances with low power consumption.

The HopeRF95w module has a range of up to 6 km in an open field and is capable of transmitting data at speeds of up to 300 kbps. It has a built-in antenna and can be easily integrated into a wide range of IoT and M2M projects. The module is supported by several development platforms, including Arduino. [8]

i) Developing a dashboard in grafana:

A Grafana dashboard is a customizable and interactive web-based interface that displays data from one or more data sources. Dashboards in Grafana are made up of rows and panels, and each panel can display a different type of visualization, such as a graph, a table, or a single value.[18]

Grafana dashboards are used to monitor and analyze various types of data, such as server metrics, application performance, and business metrics. They can be customized to show data in different ways, such as as real-time graphs or as summary tables, and they can be shared with other users or embedded in other applications.

To create a dashboard in Grafana, you can either start from scratch or use one of the pre-built dashboard templates that are available. You can then add panels to the dashboard, configure them to display data from a data source, and customize the display and layout of the dashboard.

Node-red was used as a middle-ware interface to collect data from edge devices using the MQTT protocol. Latter this data was sent to InfluxDB running locally on Raspberry Pi-4.

InfluxDB query language, InfluxQL was used to send data to InfluxDB. This data collection and visualization system make up of a total of four tools, MQTT broker, Node-red, InfluxDB, Grafana

Start Grafana by running Grafana-server application, preferably from the command line. Download NSSM if need to run Grafana as a Windows service. This tool is used to make Grafana as a Windows-service.[17]

To develop a Grafana dashboard, you can follow these steps:

I. Install Grafana: The first step is to install Grafana on your system. You can download the latest version of Grafana from the official website and follow the instructions to install it.

II. Add a data source: Once Grafana is installed, you need to add a data source that you want to use to visualize your data. Grafana supports a wide range of data sources, including time series databases, metrics platforms, and IoT platforms. You can add a data source by going to the "Configuration" menu and selecting "Data Sources".

III. Create a new dashboard: To create a new dashboard, click on the "+" icon in the left menu and select "Create dashboard". This will open a new, empty dashboard where you can add panels and widgets to visualize your data.

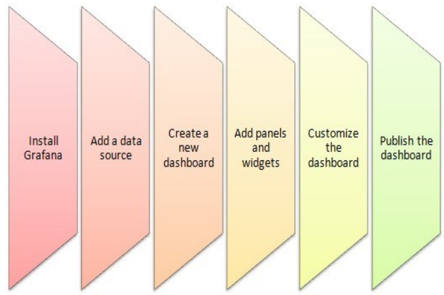

Fig. 4. Grafana dashboard Installation.

IV. Add panels and widgets: To visualize your data, you need to add panels and widgets to your dashboard. You can add different types of panels, such as graphs, tables, and maps, to display your data in various formats. To add a panel, click on the "Add panel" button and select the type of panel you want to add. Then, use the panel editor to customize the appearance and behavior of the panel.

V. Customize the dashboard: Once you have added the panels and widgets to your dashboard, you can customize the appearance and behavior of the dashboard by using the dashboard settings. You can change the layout, color scheme, and other options to suit your needs.

VI. Publish the dashboard: When you are satisfied with your dashboard, you can publish it to make it available to others. You can do this by clicking on the "Save" button and selecting the "Save and publish" option.

Since Grafana is just a data visualization tool, you need additional tools to collect and store your IoT device data[19]. In this example, we used node red as the data integrator and InfluxDB as our data source. Furthermore, we also must use a mosquito MQTT broker to collect data using the MQTT protocol. We used Raspberry Pi-4 to install the required tools.

ii) ThingStack setup

To setup the things stack as a private server in the localhost we need to have a server running Docker and Docker Compose with the recommended 4 virtual CPUs and 16GB of RAM with pogresql and radis database installed

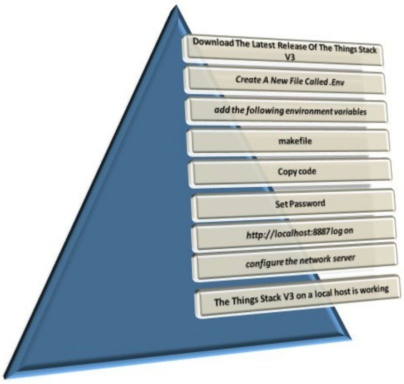

Fig. 5. Step by Step TTN setup on local host

.Device sessions, metrics, and cache are kept in a Redis database by the things stack and it uses PostgreSQL as persistent data-storage.

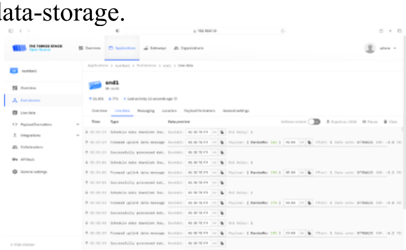

Fig. 6. Getting random number from end devices to the private server with a static IP of 192.168.1.9.

V. Results

In this our experimental setup, we were able to successfully send data from our LoRa based sensor nodes to LoRaWAN gateway. We were able to receive and store data in our time-series database InfluxDB and latter visualize it in Grafana.

For Grafana visualization we used temperature, humidity, barometric pressure sensor and air-quality sensors.

Fig. 7. InflxuDB query and data received by Grafana.

In fig5, we can see InfluxDB query and result.

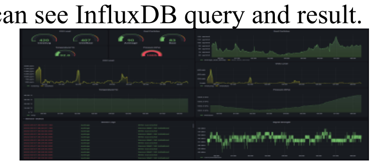

Fig. 8. Grafana weather sensor dashboard.

Here in Fig.8, we have developed a complete dashboard to visualize data from our sensors.

Fig.9.Second dashboard developed with Grafana.

For ThingStack server we used a GPS receiver connected to our microcontroller and we were able to successfully receive location data on our LoRaWAN server.

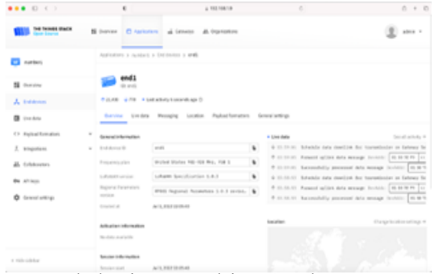

Fig. 10. Over end-device on ThingStack LoRaWAN server.

In Fig.10, We can see our device as configured in ThingStack server.

We can see we have a working setup for LoRaWAN network. We used ThingStack community edition to setup a LoRaWAN server and registered our LoRa gateway and our end device.

VI. Evaluation

After performing the above two experiments we have working set-up to get data from our node-devices and display that data in our LoRaWAN server.

On the other hand, we have created nice looking graphical interface to display data from our weather sensors.

It is possible to establish a working private LoRaWAN network server and application server tailored to our needs. It provides possibility to scale up LoRaWAN networks without compromising on security.

As described earlier, tools used in this project are totally free of cost and open source thus reducing cost of developing IoT applications based on LoRaWAN servers and further provide options for customization of such tools for specific needs.

VII. Conclusion

In above experimental setup we were able to set-up our ThingStack LoRaWAN server. The other approach we took was using Grafana and InfluxDB, together with Node-Red to receive and decode LoRaWAN packets and latter store them in our timer-series database, InfluxDB.

Even though the ThingStack can be used as private LoRaWAN server as we proved in this paper, it does not provide any data analysis and visualization tool.

Here Grafana can be used as data-analysis and data visualization. It is quite possible to use MQTT protocol from LoRaWAN server to send data directly to database thus eliminating the need of middleware tool such as node-red.

This approach can be adapted for highly scalable monitoring solutions such as agriculture, environment, and industrial applications. Several observations have been made during our experiments:

Things Stack V3 & Grafana			
Things Stack V3		***Grafana***	
Pros:	Cons:	Pros:	Cons:
Designed specifically for LoRaWAN networks and provides functionalities for managing and operating IoT devices.	May have a steeper learning curve for those who are not familiar with LoRaWAN networks or IoT device management	Supports a wide range of data sources, including databases, time-series data, and cloud-based services.	May require additional setup and configuration to connect to data sources.
Open-source and can be customized to meet specific needs.	Requires a dedicated server or cloud infrastructure to run.	Provides flexible data visualization options, including charts, graphs, and dashboards.	May not be as feature-rich as other data visualization tools.
Supports various LoRaWAN versions and can be integrated with other LoRaWAN devices and gateways.	May not be suitable for applications outside of LoRaWAN networks.	Offers a plugin architecture that can be used to extend its functionalities.	Some plugins may require a paid subscription.
Provides features for device activation, downlink messaging, and network management.	-----	Supports multiple users and roles for accessing and sharing data	Can consume significant system resources when running large queries or visualizations.
Offers a microservices architecture, which allows for flexibility and scalability.	-----	Can be accessed through a web browser, making it accessible from anywhere.	-----

Overall, both Things Stack V3 and Grafana have their strengths and weaknesses, and choosing between them will depend on the specific needs and use cases. If you are managing an IoT network that uses LoRaWAN devices, Things Stack V3 may be the better choice. If you need to visualize data from various sources, Grafana may be the better option.

REFERENCES

[1] Oniga, Bogdan &Dadarlat, V. & De Poorter, Eli & Munteanu, Adrian. (2017). Analysis, design and implementation of secure LoRaWAN sensor networks. 421-428. 10.1109/ICCP.2017.8117042

[2] Lee H.C., Ke K.H. Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation. IEEE Trans. Instrum. Meas. 2018;67:1–11. doi: 10.1109/TIM.2018.2814082.

[3] Lee, John & Souryal, Youssef & Kim, Dongsoo & Kang, Kyubyung & Koo, Dan. (2019). Building a Private LoRaWAN Platform.

[4] LoRa Alliance. https://lora-alliance.org/about-lorawan/

[5] LoRaDeveloper.https://loradevelopers.semtech.com/documentation/tech-papers-and-guides/lorawan-class-a-devices/

[6] RP2-1.0.3 LoRaWAN® Regional Parameters, https://lora-alliance.org/resource_hub/rp2-1-0-3-lorawan-regional-parameters/

[7] LoRa Technical overview, Radio modulation And LoRa. https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[8] Documentation.https://loradevelopers.semtech.com/documentation/tech-papers-and-guides/implementing-adaptive-data-rate-adr/implementing-adaptive-data-rate/

[9] G. Ferré and A. Giremus, "LoRa Physical Layer Principle and Performance Analysis," 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Bordeaux, France, 2018, pp. 65-68, doi: 10.1109/ICECS.2018.8617880.

[10] Raspberry-pi3 https://www.raspberrypi.com/products/raspberry-pi-3-model-b/

[11] LPS8 Gateway Datasheet_LPS8_LoRaWAN Pico Station.pdf.

[12] ThingStackinstallation.https://www.thethingsindustries.com/docs/getting- started/installation/

[13] ThingStack GitHub repository. https://github.com/xoseperez/the-things-stack-docker

[14] ThingStack.https://www.thethingsindustries.com/docs/getting-started/what-is-tts/#ttn-the-things-network

[15] influxdb.https//www.influxdata.com/

[16] Grafana OSS https://grafana.com/oss/grafana/

[17] Grafana.https://grafana.com/grafana/

[18] M. C. Al Fajar and O. N. Samijayani, "Realtime Greenhouse Environment Monitoring Based on LoRaWAnProtocol using Grafana," 2021 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 2021, pp. 1-5, doi: 10.1109/ISESD53023.2021.9501628.