University of Nevada, Reno

Cyberinfrastructure for IoT-Based Environmental Monitoring: Advancements with LoRaWAN

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in Computer Science and Engineering

by

Zachary M. Estreito

Dr. Sergiu M. Dascalu, Co-Advisor
Dr. Frederick C. Harris, Jr., Co-Advisor

December, 2024

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Zachary M. Estreito

Entitled

Cyberinfrastructure for IoT-Based Environmental Monitoring: Advancements with LoRaWAN

be accepted in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Dr. Sergiu M. Dascalu, Co-Advisor

Dr. Frederick C. Harris, Jr., Co-Advisor

Dr. Scotty Strachan, Graduate School Representative

Dr. Markus Kemmelmeier, Dean, Graduate School

December, 2024

Abstract

Environmental research increasingly relies on advanced technological systems to handle data collection, processing, and application at scale. However, issues like device interoperability, system scalability, and accessibility often limit adoption. This thesis presents a LoRaWAN-based environmental monitoring system developed as part of the National Science Foundation's Cyberinfrastructure for Sustained Scientific Innovation (CSSI) initiative. LoRaWAN's energy-efficient and long-range communication capabilities make it ideal for applications such as environmental monitoring, precision agriculture, and industrial operations.

The system integrates cutting-edge hardware and powerful open-source software frameworks to ensure seamless data collection and management. Comprehensive assessments of the system and its underlying technology demonstrate its reliability, energy efficiency, and scalability in varied operational scenarios. By addressing key technical challenges, this work provides a foundation for highly scalable, user-friendly environmental monitoring systems that meet the diverse needs of researchers, industries, and the general public.

Dedication

I dedicate this thesis to my incredibly supportive family and friends. I thank my parents and brothers for being a constant source of encouragement even from hundreds of miles away, and I thank Vinh Le and all of the other friends I made during my studies for helping me realize my potential.

Acknowledgments

I would like to thank my co-advisors Dr. Sergiu M. Dascalu and Dr. Frederick C. Harris, Jr. for their shared wisdom and guidance throughout my graduate studies. I extend my gratitude to Dr. Scotty Strachan, member of this thesis' examination committee, for his expert technical advisement on the development of the environmental monitoring system presented here. I am grateful to Co-Principal Investigator Dr. William Savran for his software engineering advice. My thanks go also to my graduate student peers Chase Carthen, Kaden Nesch, and Nikhil Sharma for their contributions to the system's architecture and user interfaces. I would also like to thank PhD student Jehren Boehm for his contributions to the project in sensor and gateway deployments as well as web interface feedback.

This material is based in part upon work supported by the National Science Foundation under grant number OAC-2209806. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, IoT technology has significantly improved the way environmental data is collected, processed, and applied. Among the emerging technologies in this domain, LoRaWAN (Long Range Wide Area Network) is gaining traction due to its ability to transmit data across long distances with minimal power consumption. This makes it an attractive option for applications such as environmental monitoring, precision agriculture, and industrial operations. Nevertheless, deploying LoRaWAN-based systems presents challenges, including ensuring device interoperability, managing large-scale data streams, and making the technology accessible to users with varying levels of expertise.

This thesis focuses on the design and implementation of a LoRaWAN-based environmental monitoring system developed as part of the NSF-funded Cyberinfrastructure for Sustained Scientific Innovation (CSSI) Elements: Innovating for Edge-to-Edge Climate Services research project [16]. For simplicity, this project will be referred to as the "CSSI project" throughout this thesis. The platform integrates cutting-edge technologies, including The Things Stack [34] for network management, Apache NiFi [15] for data processing, and TimescaleDB for time-series data storage [25]. Its aim is to support a broad range of stakeholders, from researchers and citizen scientists to industry professionals, by delivering a robust, scalable, and user-friendly solution for real-time monitoring and data analysis.

The growing threats of climate change, resource depletion, and environmental degradation have amplified the need for reliable and cost-effective monitoring solutions. Traditional monitoring technologies often fall short due to their high costs, limited scalability, and significant energy demands. LoRaWAN technology, with its efficiency and affordability, provides a promising alternative but remains underutilized in broader contexts due to a lack of accessible, scalable implementations.

This thesis seeks to address this gap by creating an accessible platform tailored to a wide range of environmental monitoring needs. The project aligns with the NSF's mission to advance cyberinfrastructure by developing tools that are technically capable, yet approachable to non-expert users. The overarching goal is to foster innovation and inclusivity in scientific research, encouraging diverse communities to participate in the collection and analysis of environmental data.

The primary contribution of this work lies in the development of a LoRaWAN-based platform that balances technical sophistication with ease of use. The system demonstrates how open source tools can be combined to create a modular architecture that adapts to varying scales and applications. By addressing usability barriers, it enables a broader participation in scientific monitoring efforts, from academic researchers to grassroots citizen science initiatives. This work also highlights the scalability of the platform, showcasing its utility across academic, industrial, and public sector use cases. Through real-world testing, the system validates LoRaWAN's potential to provide actionable insights into environmental conditions, supporting both immediate applications and long-term scientific studies.

This thesis is organized to systematically address the conceptual, technical, and practical dimensions of the project. It begins with a review of IoT and LoRaWAN fundamentals in Chapter 2, laying the groundwork for understanding the system's architecture and capabilities. Chapter 3 surveys related research, identifying existing challenges and gaps that informed the platform's design. Chapter 4 delves into the technical implementation of the system, detailing the integration of hardware, software, and network components. In Chapter 5, various use cases are explored, demonstrating the versatility of the system and the practical relevance of the system in various fields. Chapter 6 evaluates the results of usability testing, providing insight into how the platform meets user needs and where improvements are possible. Finally, Chapter 7 summarizes the contributions of this work, discusses its limitations, and outlines potential avenues for future research.

By combining innovative technology with an emphasis on inclusivity and sustainability, this thesis demonstrates the transformative potential of LoRaWAN within the broader context of IoT-enabled monitoring. The platform not only addresses current environmental monitoring challenges but also establishes a foundation for future exploration and application, advancing the mission of CSSI to sustain scientific innovation through adaptable and impactful infrastructure.

Chapter 2

Background

2.1 LoRa

LoRa (Long Range) is a wireless radio frequency transmission protocol designed for long-distance, low-power communication [35]. LoRa was originally developed by the French company Cycleo and was patented by its co-inventor Olivier Bernard Andre Seller in 2014 [41]. Cycleo was later purchased by Semtech [13]. The key features of the LoRa protocol are as follows:

1. **Long Range**: The LoRa protocol is effective at ranges in excess of 10 kilometers under perfect conditions. In practice, this would entail a rural area with an unobstructed line of sight between the two devices. Due to this, LoRa has seen increasing use in agriculture, such as smart irrigation systems and monitoring of environmental conditions on farms [22, 29]. In urban environments, the effective range is generally reduced to a few kilometers. This capability makes LoRa suitable for applications requiring wide-area coverage [30, 48].

2. **Low Power Consumption**: The LoRa protocol is designed to operate with minimal power consumption, allowing LoRa devices to have years of battery life. This makes the LoRa protocol ideal for remote sensors and IoT devices where frequent battery replacement is impractical [4].

3. **Low Data Rate**: LoRa supports data rates ranging from 0.3 kbps to 50 kbps. Although this data rate is quite low compared to other wireless communication

protocols, it is sufficient for many IoT applications, especially those involving remote sensors.

4. **Robustness and Reliability**: LoRa's spread-spectrum modulation technique ensures reliable communication even in environments with substantial interference. Its ability to overcome obstacles such as buildings and trees while maintaining communication integrity is highly beneficial for IoT deployments in areas with significant line-of-sight obstructions. [40, 45].

5. **Unlicensed Spectrum**: LoRa typically operates in the unlicensed ISM (Industrial, Scientific, and Medical) bands, which means that it can be deployed without the need for any regulatory board approval or licensing. This reduces the regulatory overhead of IoT projects that utilize LoRa devices. Specific frequency bands vary by region (e.g., 915 MHz in North America, 923 MHz in Asia and South America, and 868 MHz in Europe)[20].

LoRa serves as just the physical layer of a communication system, providing the underlying mechanism for data transmission over long distances. LoRa on its own does not define any higher-layer protocols, such as those needed for network management, data routing, or application interfaces. Instead, those are provided by the larger LoRaWAN protocol.

2.2 LoRaWAN

LoRaWAN (Long Range Wide Area Network) is a communication protocol and system architecture built on top of the LoRa physical layer. LoRaWAN is developed and maintained by the LoRa Alliance. LoRaWAN defines the network structure, device communication protocols, and overall system architecture to enable scalable and secure long-range communication for IoT applications. The main features of LoRaWAN are as follows:

1. **Network Architecture**: LoRaWAN uses a hierarchical star network topology,

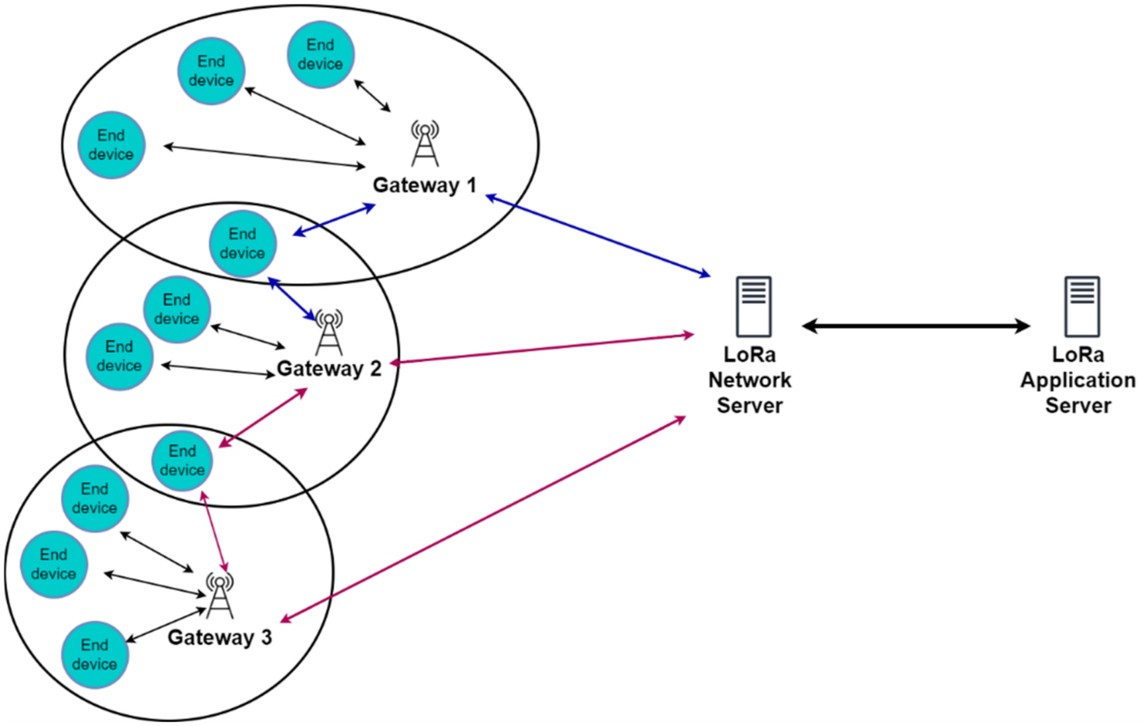


Figure 2.1: Diagram of LoRaWAN's hierarchical star topology [3].

which can be seen in Fig. 2.1. End devices communicate with gateways using the LoRa physical layer. Gateways act as bridges, forwarding data between end devices and a central network server. Communication between LoRaWAN gateways and the central network server uses standard internet protocol (IP) connections. The central network server manages the network. This includes device authentication, data routing, and security [9, 14].

2. **Classes of Devices**:

   (a) **Class A**: The base LoRaWAN class with the most minimal requirements. This is the most energy-efficient end device class, suitable for applications with sparse communication. Class A LoRaWAN devices have two short receive windows after each transmission. They are only able to receive packets from the network server during these receive windows. This can be visualized in Fig. 2.2

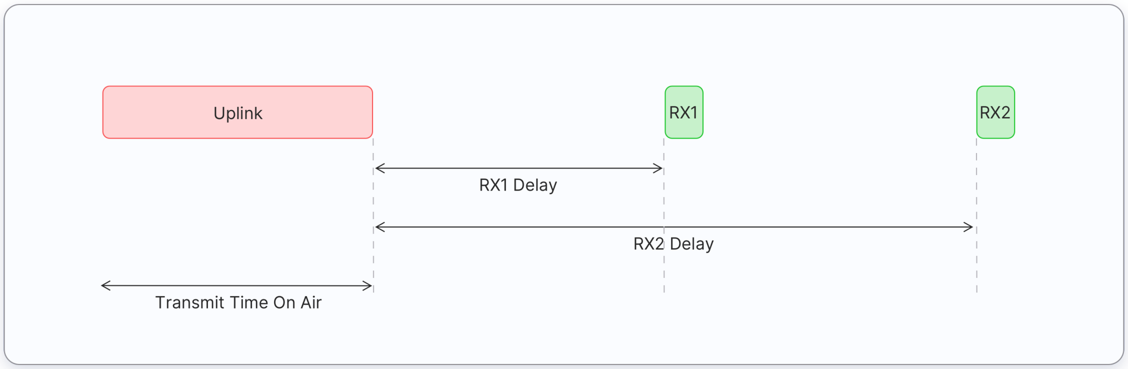


Figure 2.2: Uplink and downlink communication timing in LoRaWAN Class A devices [33].

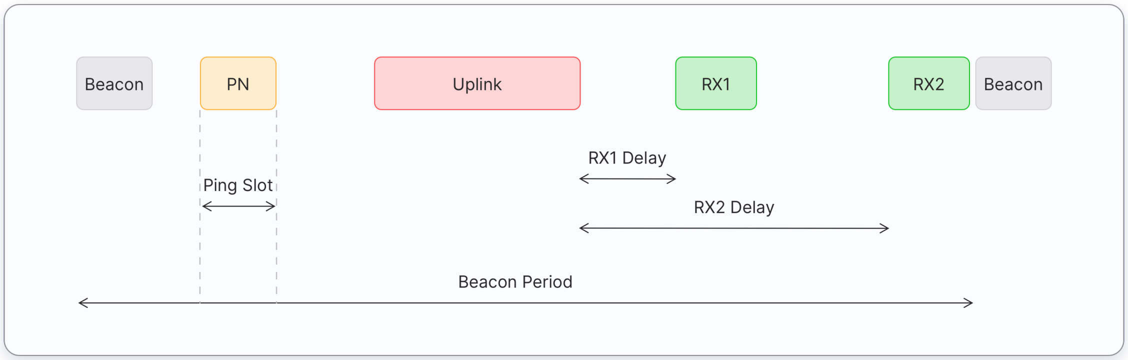


Figure 2.3: Uplink and downlink communication timing in LoRaWAN Class B devices [33].

(b) **Class B**: This expands upon Class A by adding scheduled receive windows, called *ping slots*, during which devices listen for downlink packets from the central network server, as shown in Fig. 2.3. This is suitable for interactive applications that require bidirectional communication with end devices.

(c) **Class C**: This further expands on the bidirectional communication capabilities of Class B by having continuous receive windows. These receive windows only close when the end device is transmitting, which allows for low-latency bi-directional communication. This functionality comes at the expense of higher power consumption. The timings can be seen in Fig. 2.4

3. **Security**: LoRaWAN includes powerful security features to ensure data in-

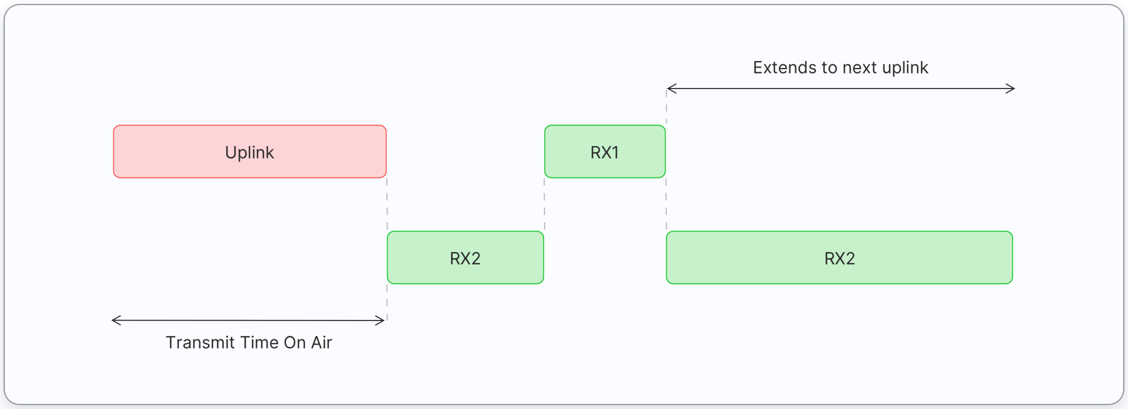


Figure 2.4: Uplink and downlink communication timing in LoRaWAN Class C devices [33].

tegrity and privacy. The network and application layers are both encrypted with session keys. This provides end-to-end security between LoRa devices and application servers.

4. **Scalability**: LoRaWAN is designed to support millions of devices within a single network. The highly scalable architecture allows for dynamic network resource allocation, adaptive data rates (ADR), and efficient spectrum utilization, which is essential for large-scale IoT deployments [49]. LoRaWAN systems can also be optimized for performance and efficiency using various AI techniques, ensuring that network parameters are dynamically adjusted to improve communication reliability and coverage [24].

5. **Interoperability**: The LoRaWAN protocol defines standard compliance and certification processes to ensure interoperability between devices and networks. This allows devices from different manufacturers to work seamlessly with each other on the same network.

Building upon the LoRa physical layer, LoRaWAN provides an encompassing networking framework that meets the needs of large-scale IoT deployments by ensuring reliable, secure, and efficient communication between end devices and network servers. Currently, one of the most common use cases for LoRaWAN is environmental and agricultural monitoring. In addition, LoRaWAN has been widely adopted in smart city infrastructure for applications such as traffic management and smart lighting [31].

2.3 The Things Stack Open Source Edition

The Things Stack (TTS) [34] is an advanced LoRaWAN network server developed by The Things Industries. It provides an exhaustive solution for managing LoRaWAN networks. The Things Stack is available in multiple versions tailored for different needs, including the Open Source Edition, Enterprise Edition, and Cloud-hosted versions. This section focuses specifically on the Open Source Edition, which is open-source and freely available.

The Things Stack Open Source Edition leverages LoRaWAN technologies to enable seamless communication between devices, applications, gateways, and organizations. Below, the key components of The Things Stack Open Source Edition are outlined.

2.3.1 End Devices

End devices are the sensors or actuators that collect or use data in a LoRaWAN network. The Things Stack supports a broad spectrum of end devices, making it versatile for many applications. Technical aspects of end devices within TTS include:

1. **Device Registration**: TTS provides mechanisms for easy device registration and management through a web interface, typically involving the configuration of device-specific parameters such as DevEUI, JoinEUI, and AppKey.

2. **Class Support**: There is complete support in TTS for all LoRaWAN device classes (A, B, and C), which cater to different application needs.

3. **Over-the-Air Activation (OTAA)**: The Things Stack supports OTAA, a secure method for device activation where devices dynamically join the network

using a join request and a join accept message, ensuring that network session keys are unique and secure.

4. **Payload Formatters**: The platform includes payload formatters that allow users to encode and decode binary payloads. These formatters can be written in JavaScript, allowing customization of data processing to match application-specific requirements.

5. **Device Repository**: The device repository is an extensive database of LoRaWAN devices with predefined settings and configurations. This feature streamlines the process of adding new devices to the network by providing templates that ensure correct parameter settings, reducing setup time and potential errors. It also includes payload formatters written by the device manufacturers, alleviating the need to write payload formatters for every unique end device type.

2.3.2 Gateways

Gateways serve as the bridge between end devices and the network server in a LoRaWAN network. They receive RF signals from end devices using the LoRa modulation technique and forward the data to the network server via IP connections. Gateway-specific features within The Things Stack include:

1. **Diverse Gateway Support**: TTS supports a wide range of gateway models, including commercial and DIY options, allowing flexibility in hardware choices.

2. **Easy Configuration**: The Things Stack facilitates the straightforward setup and configuration of gateways, often only involving the configuration of parameters such as Gateway EUI, frequency plan, and network server address. The built-in support for popular gateway models simplifies this process.

3. **Scalability**: Multiple gateways can be deployed to enhance network coverage and redundancy. This is what makes LoRaWAN a hierarchal star network

topology. This setup ensures robust and reliable data transmission across large deployment areas, optimizing network performance and reliability.

2.3.3 Applications

Applications within The Things Stack are responsible for grouping end devices and processing their data. The application page allows for seamless integration with various third-party applications, providing flexibility for custom solutions. The important aspects of applications within The Things Stack include:

1. **Data Routing**: Efficient data routing mechanisms ensure that data from end devices is reliably delivered to the appropriate applications. This involves the use of MQTT, HTTP, or gRPC for data transmission.

2. **Integrations**: The Things Stack includes built-in integrations with major cloud platforms and services such as AWS IoT, Azure IoT Hub, and Google Cloud IoT Core. These integrations facilitate advanced data processing, analytics, and storage capabilities.

3. **API Access**: The Things Stack provides a powerful API that allows developers to create custom applications and integrations. The APIs allow for operations such as device management, data retrieval, and network configuration. The API supports both the REST and gRPC protocols.

2.3.4 Organizations

Organizations in The Things Stack represent logical groupings of users, devices, and applications. They provide a structured approach to managing and sharing resources within a network. The technical features of organizations within The Things Stack include:

1. **Multi-Tenancy**: TTS supports multi-tenancy, allowing multiple organizations to coexist within a single instance of The Things Stack. This enables efficient resource management and the logical separation of different entities.

2. **Role-Based Access Control (RBAC)**: Granular access control mechanisms ensure that users have the appropriate level of access to resources within TTS. RBAC allows for the definition of roles and permissions, enhancing security and operational efficiency.

3. **Collaboration**: The Things Stack provides tools for collaboration within and among organizations. This includes features for sharing gateways, devices, and applications.

2.4 MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for efficient communication in constrained environments, such as IoT devices and low-bandwidth networks [4]. MQTT follows a publish-subscribe messaging model, which decouples the communication between devices, enabling scalable and flexible network architectures. Developed by IBM in the late 1990s, MQTT has since become a widely adopted standard for IoT communication. The MQTT standard is overseen by the OASIS MQTT Technical Committee. The key components and features of MQTT are as follows:

1. **Clients**: MQTT clients can be devices, applications, or services that publish or subscribe to topics. Clients communicate with the MQTT broker using TCP/IP connections [5].

2. **Broker**: The MQTT broker is the central hub that receives messages from publishers and routes them to the appropriate subscribers. It manages client connections, subscriptions, and message delivery to ensure efficient and reliable communication.

3. **Lightweight and Efficient**: MQTT is designed to be lightweight, requiring minimal bandwidth and processing power. This makes it suitable for resource-constrained devices and networks with limited bandwidth.

4. **Publish-Subscribe Model**: In MQTT, clients can act as publishers, subscribers, or both. Publishers send messages to topics, and subscribers receive messages from topics they are subscribed to. This model decouples message producers and consumers, enhancing scalability and flexibility.

5. **Quality of Service (QoS) Levels**: MQTT supports three QoS levels to ensure message delivery reliability:

    (a) **QoS 0 (At most once)**: Messages are delivered at most once, without acknowledgment. This is suitable for applications where occasional message loss is acceptable.

    (b) **QoS 1 (At least once)**: Messages are delivered at least once, requiring acknowledgment from the receiver. This ensures that messages are not lost, but may result in duplicate messages.

    (c) **QoS 2 (Exactly once)**: Messages are delivered exactly once. This involves a two-step acknowledgment process to ensure that there are no duplicates. This is the highest level of delivery assurance.

6. **Retained Messages**: MQTT allows publishers to send retained messages, which are stored by the broker and delivered to new subscribers immediately upon subscription. This feature is useful for sending the latest status updates or configuration information.

7. **Last Will and Testament (LWT)**: Clients can specify a Last Will and Testament message, which the broker sends if the client disconnects unexpectedly. This feature helps to detect and handle client failures.

8. **Encryption**: MQTT supports Transport Layer Security (TLS) to encrypt data transmitted between clients and the broker, protecting against eavesdropping and tampering.

9. **Authentication**: MQTT clients can authenticate with the broker by using a username and password. In addition, mutual TLS authentication can be employed for enhanced security.
10. **Access Control**: MQTT brokers can implement access control mechanisms to restrict clients' access to specific topics, ensuring that only authorized clients can publish or subscribe to sensitive data.

In summary, MQTT is a powerful and flexible protocol tailored for efficient communication in IoT environments. Its lightweight design combined with the publish-subscribe model, QoS levels, and robust security features makes it an ideal choice for a wide range of applications - especially IoT applications.

2.5 Apache NiFi

Apache NiFi [15] is an open-source data integration and distribution framework designed to automate the flow of data between systems. Originally developed by the United States National Security Agency (NSA) and later released as part of the Apache Software Foundation, NiFi provides a user-friendly interface and versatile capabilities for building data pipelines. It supports data ingestion, routing, transformation, and system-to-system transfer, making it suitable for managing complex data flows in real-time [8]. Apache NiFi is often used to automate the flow of data between IoT devices and processing systems due to its ability to handle large, real-time data streams [11, 37]. Key components and features of Apache NiFi include:

1. **Flow-Based Programming**: NiFi uses a flow-based programming model in which users design data flows through a graphical interface. This approach simplifies the visualization and management of data movement, allowing the creation of complex workflows by connecting various processors.
2. **Processors**: NiFi offers a wide range of processors that handle different aspects of data processing, such as ingestion, transformation, and routing. These

processors can be configured and linked together to form custom data flows that meet specific application needs.

3. **Provenance Tracking**: NiFi tracks the origin, movement, and transformation of data within the system, aiding in debugging, auditing, and ensuring data integrity.
4. **Scalability**: NiFi scales horizontally by adding more nodes to the system, accommodating increasing data loads. This makes it suitable for both small-scale deployments and large enterprise-level data integration tasks.
5. **Security**: NiFi includes security features such as role-based access control (RBAC), data encryption, and secure communication via TLS, ensuring that data remains protected throughout the pipeline.
6. **Customizability**: NiFi allows users to develop custom processors and integrate them into existing data flows, offering flexibility for a wide range of use cases and industries.

NiFi is a valuable tool for building, managing, and monitoring data flows in real-time. Its user-friendly interface, extensive processing capabilities, and security features make it a reliable choice for automating data integration and ensuring consistent data movement between systems.

2.6 TimescaleDB

TimescaleDB [25] is an open-source time-series database built on PostgreSQL [46], designed to efficiently handle high-volume time series data. It extends PostgreSQL by providing additional functionality and optimizations for time-series data, making it suitable for applications that involve tracking and analysis of data points over time, such as collecting climate data from LoRaWAN sensors [4]. Key features and components of TimescaleDB include:

1. **Time-Series Data Storage**: TimescaleDB introduces hypertables, optimized for storing time-series data. Hypertables partition data by time and space, enabling efficient storage and retrieval of large datasets.

2. **Scalability**: Built on PostgreSQL, TimescaleDB supports horizontal and vertical scaling, with automatic data partitioning to manage large volumes of time-series data without compromising performance [51]. Compared to other time-series focused databases, TimescaleDB benchmarks excellently [27].

3. **Advanced Querying**: TimescaleDB supports complex time-series queries, including time-based aggregations, downsampling, and time-bucket functions, allowing users to analyze time-series data directly within the database [19].

4. **Integration with PostgreSQL Ecosystem**: As an extension of PostgreSQL, TimescaleDB integrates seamlessly with existing PostgreSQL tools, extensions, and SQL queries, leveraging the full capabilities of PostgreSQL for time-series data.

5. **Data Retention and Compression**: TimescaleDB includes support for data retention policies and compression. Users can define policies to automatically delete old data and apply compression techniques to reduce storage requirements.

6. **Real-Time Analytics**: TimescaleDB is optimized for real-time analytics, enabling the ingestion and analysis of time-series data as it arrives, making it suitable for applications requiring immediate insights, such as real-time monitoring of climate data from sensors [18, 32].

TimescaleDB combines PostgreSQL's reliability with specialized time-series capabilities, making it an effective tool for managing and analyzing climate data collected from LoRaWAN sensors.

Chapter 3

Related Work

LoRaWAN technology in the context of Internet of Things (IoT) has gained significant attention in recent years due to its low power consumption and long-range communication capabilities. These attributes make it highly suitable for a variety of domains, including environmental monitoring, smart agriculture, and industrial applications. This section summarizes recent research on LoRaWAN in these fields, examining the advantages and challenges of its implementation while exploring key contributions that highlight its relevance in sustainability, resource management, and industrial efficiency.

3.1 LoRaWAN in Smart Agriculture

Smart agriculture is a highly relevant application for IoT-based monitoring systems, driven by the need for greater efficiency in food production, water management, and environmental sustainability. LoRaWAN technology is a great candidate for the enablement of precision agriculture. The use of LoRaWAN technology in smart agriculture offers scalable, low-cost solutions to monitor and manage agricultural environments [29].

One such example is Citoni et al. [12], who provided a comprehensive review of smart farming enabled by IoT and LoRaWAN. The authors address the global challenge of feeding an ever-growing population in the face of limited resources and climate change. Their study emphasizes LoRaWAN's low-power and long-range capabilities, which allow for the deployment of sensor networks that monitor essential parameters. These parameters include air temperature, humidity, light intensity, and soil moisture. The ability to remotely collect and analyze data makes it possible for farmers to optimize resource use and improve crop yields while minimizing water waste and energy consumption. The ability of LoRaWAN to scale with the demands of large-scale farming operations makes it an ideal technology for the future of agriculture, especially in regions facing resource constraints due to climate change and urbanization.

Valente et al. [47] presented a similar application in their development of a LoRaWAN-based system for monitoring vine water status in vineyards. Their research highlights how real-time data collection on soil moisture and other environmental conditions can improve irrigation decisions, leading to more efficient water use in agricultural practices. This is particularly important in areas facing water scarcity, where precise irrigation can mitigate the negative effects of droughts. Their work exemplifies how LoRaWAN technology can be utilized to address difficult environmental challenges while ensuring agricultural productivity. Water efficiency is becoming increasingly important as water shortages and climate variability affect many regions around the world, making precise control over irrigation a necessity rather than a cost-saving measure.

Similarly, Fraga-Lamas et al. [17] proposed a LoRaWAN-based smart irrigation system that uses a decision-support system to optimize water usage in urban environments. In this context, a decision-support system refers to a system that would make recommendations or automate decisions based on LoRaWAN sensor readings. Their work addresses the unique challenges of deploying IoT networks in cities, where physical obstacles and long distances often hinder wireless communications. Using LoRaWAN, they were able to implement a system that reduces energy consumption while improving irrigation efficiency. This shows how LoRaWAN's long-range capabilities make it an ideal solution for both rural and urban agriculture. As cities expand, the need to integrate smart irrigation systems into urban planning to promote sustainability increases, particularly in public green spaces and urban farms.

Precision agriculture in controlled environments, such as greenhouses, has also benefited from LoRaWAN deployment. Singh et al. [43] explored its use in greenhouses, where the technology is used to monitor temperature, humidity, and CO_2 levels to optimize growing conditions of plants. In this study, the authors highlight how LoRaWAN's low-power communication helps to maintain optimal conditions for plant growth, even in environments where wireless signals may be disrupted by the greenhouse structure. These findings are significant for greenhouse agriculture, where maintaining precise control over environmental conditions is critical to maximize crop yield and quality. Another study by Singh et al. [44] demonstrated a proof-of-concept utilizing LoRaWAN signal metadata, specifically received signal strength indicator (RSSI), to infer plant growth in a greenhouse without requiring additional sensing hardware. Their study leveraged a multilayer perceptron (MLP) neural network trained on a dataset of RSSI values and corresponding manual plant height measurements. This innovative approach revealed a correlation between RSSI fluctuations and plant growth, where increasing plant height caused measurable changes in RSSI due to signal attenuation and physical obstruction. By repurposing the existing LoRaWAN infrastructure, the system offers a cost-effective solution for monitoring plant growth while minimizing complexity and additional investments. With a reported root mean square error of 10% in plant height predictions, the methodology underscores the potential of LoRaWAN to improve precision agriculture, particularly in greenhouse settings where optimizing resource use is critical.

The integration of IoT with smart farming is not limited only to crops and irrigation systems. Animal welfare and farm equipment management are other potential applications in which LoRaWAN can play a role. By attaching sensors to animals or equipment, farmers can remotely monitor livestock health or equipment status, further enhancing the farm's overall efficiency and reducing labor requirements.

3.2 Environmental Monitoring Systems

LoRaWAN's applications extend beyond agriculture to environmental monitoring, particularly in scenarios where continuous data collection over long distances is required. A notable example is Jabbar et al. [23], who developed a LoRaWAN-based air quality monitoring system to track pollutants such as NO_2, SO_2, and CO_2. Their system relies on solar-powered sensors that are connected via LoRaWAN to provide real-time data on air quality in urban environments. This allows for more energy-efficient monitoring of pollution levels, enabling authorities to take timely action in response to deteriorating air quality. The solar power functionality also allows the sensors to potentially function for longer periods without battery maintenance. Long-distance communication with minimal energy consumption makes LoRaWAN a promising technology in the effort to monitor and mitigate environmental hazards [37]. In urban settings, air quality monitoring is becoming increasingly critical as cities grow and industrial activities expand. LoRaWAN provides a cost-effective and scalable means of deploying air quality sensors in large metropolitan areas, contributing to more sustainable urban development.

Boonyopakorn et al. [6] focused on environmental monitoring for smart agriculture in Thailand, where they deployed a LoRaWAN climate sensor network to measure environmental factors in rural farming areas. Their research highlights the importance of real-time data collection in helping farmers make informed decisions regarding soil preparation, crop selection, and irrigation. The integration of IoT with traditional farming practices has the potential to significantly improve agricultural productivity while reducing environmental degradation. This study also highlights the importance of using technology to support small-scale farmers in developing regions, where access to real-time environmental data can be transformative. By deploying inexpensive LoRaWAN-based solutions, small-scale farmers can gain access to the same technological advantages as larger, more resource-rich farms.

A different approach to environmental monitoring was demonstrated by Ramson et al. [39], who developed a LoRaWAN-based system for continuous soil health monitoring. Their system measures essential soil properties, including moisture, temperature, and electrical conductivity, which are crucial for maintaining soil fertility and improving crop yields. The study emphasizes the importance of long-range, low-power communication in remote areas, where traditional soil health monitoring infrastructure is often unavailable. LoRaWAN's ability to transmit data over several kilometers without the need for additional nodes highlights its suitability for large-scale environmental monitoring projects. These types of systems are vital for precision farming in regions where poor soil management practices have historically led to degradation, reducing productivity over time.

3.3 LoRaWAN in Extreme Environments

The resilience of LoRaWAN in extreme environments has been the subject of several studies, further demonstrating its versatility. Bezerra et al. [45] conducted a case study in northern Sweden, where they evaluated the performance of LoRaWAN in subarctic conditions. Their research found that cold weather improved the signal-to-noise ratio (SNR), while warmer conditions decreased it. This ability to function in extreme temperatures highlights LoRaWAN's adaptability, making it a reliable choice for deployments in both harsh climates and temperate regions. This flexibility is critical for remote sensing applications in areas where extreme weather can affect traditional communication networks. The study shows how LoRaWAN can be adjusted to maintain reliable performance even in very challenging environments, further validating its use in global monitoring networks.

In a similar vein, Wang et al. [50] developed a LoRaWAN-based system that uses solar and RF energy to power environmental monitoring sensors in remote areas. This system is designed to function without regular maintenance, using energy harvesting techniques to ensure long-term sustainability. The study highlights LoRaWAN's potential for monitoring difficult-to-reach areas where access to the electrical grid is limited or nonexistent, further cementing its role in global environmental monitoring efforts. Remote areas, such as those in developing countries or sparsely populated regions, stand to benefit from this technology, which reduces the need for costly and labor-intensive infrastructure development. The ability to monitor environmental factors such as air quality, soil conditions, or weather patterns in real time could provide essential data for both local authorities and global climate models.

3.4 Data Center and Industrial Applications

In addition to its applications in agriculture and environmental monitoring, LoRaWAN has been deployed in industrial settings, where energy efficiency is a key concern. Polonelli et al. [36] illustrated the utility of LoRaWAN in data center management by using it to monitor temperature levels within high performance computing environments. Their study revealed that fine-grained temperature monitoring via LoRaWAN can lead to significant energy savings by optimizing cooling processes. Given that data centers consume high amounts of energy both to power computing hardware and to cool it, the ability to monitor and adjust cooling levels in real-time represents a substantial opportunity for reducing operational costs and improving sustainability.

This industrial application showcases LoRaWAN's flexibility and scalability in nonagricultural settings, where the technology can be adapted to address energy and resource management challenges. By using LoRaWAN to monitor environmental variables in real time, industries can reduce their carbon footprint and improve operational efficiency. For instance, factories and warehouses with large areas to monitor can deploy low-power LoRaWAN sensors for real-time monitoring of environmental conditions, contributing to better energy management and lower operational costs.

3.5 Challenges and Future Directions

Although LoRaWAN offers numerous benefits, several challenges remain that limit its full potential, particularly in large-scale applications. Citoni et al. [12] identified key bottlenecks in LoRaWAN deployment for smart farming, including limitations in data transmission rates and network scalability. Although LoRaWAN excels at transmitting small data packets over long distances, it struggles with high-bandwidth applications, such as those requiring the transmission of large datasets or multimedia files. To overcome this, hybrid systems that combine LoRaWAN with other wireless technologies may be necessary to achieve greater network capacity and higher data throughput.

Temperature variations also present a challenge for LoRaWAN networks. Bezerra et al. [45] discussed the deleterious effects of environmental conditions on network performance, particularly elevated temperatures, highlighting the need for careful network planning to ensure consistent performance in varying climates. The potential integration of artificial intelligence (AI) with LoRaWAN networks, as proposed by Singh et al. [44], could help address these challenges. AI-driven systems could optimize data transmission, manage network resources more effectively, and provide predictive maintenance capabilities, further enhancing the reliability and scalability of LoRaWAN deployments [37]. Artificial intelligence is increasingly being incorporated into IoT systems to predict network failures, optimize routing protocols, and improve overall system performance.

In the coming years, the role of LoRaWAN in the Internet of Things (IoT) is expected to expand further as the demand for scalable, energy-efficient communication networks grows. Research into hybrid systems, improved power management techniques, and AI-based optimization will be important to overcome the current limitations of LoRaWAN technology.

Chapter 4

System Design and Implementation

4.1 System Architecture Overview

The system architecture for the LoRaWAN-based climate data collection project is designed to efficiently gather environmental data from distributed sensors, transmit it through LoRaWAN gateways, process and store the data, and provide tools for data visualization and export. The integration of all software packages outlined in this section creates a scalable system capable of handling real-time data from numerous end devices. A high-level diagram of the network architecture of this system can be seen in Fig. 4.1. The key components and their interactions are described below.

4.1.1 Data Collection and Transmission

Environmental data is measured by LoRaWAN sensors (end devices) deployed in the field. These devices collect various climate parameters such as air temperature, air humidity, and soil moisture. The data collected is transmitted wirelessly to the nearest LoRaWAN gateway using the LoRa radio frequency networking protocol, leveraging its long-range and low-power capabilities.

4.1.2 Gateway Communication

The LoRaWAN gateways receive the data from the end devices and forward it to the network server. Depending on the gateway model and the selected communication protocol, data transmission from the gateways to the network server can occur via

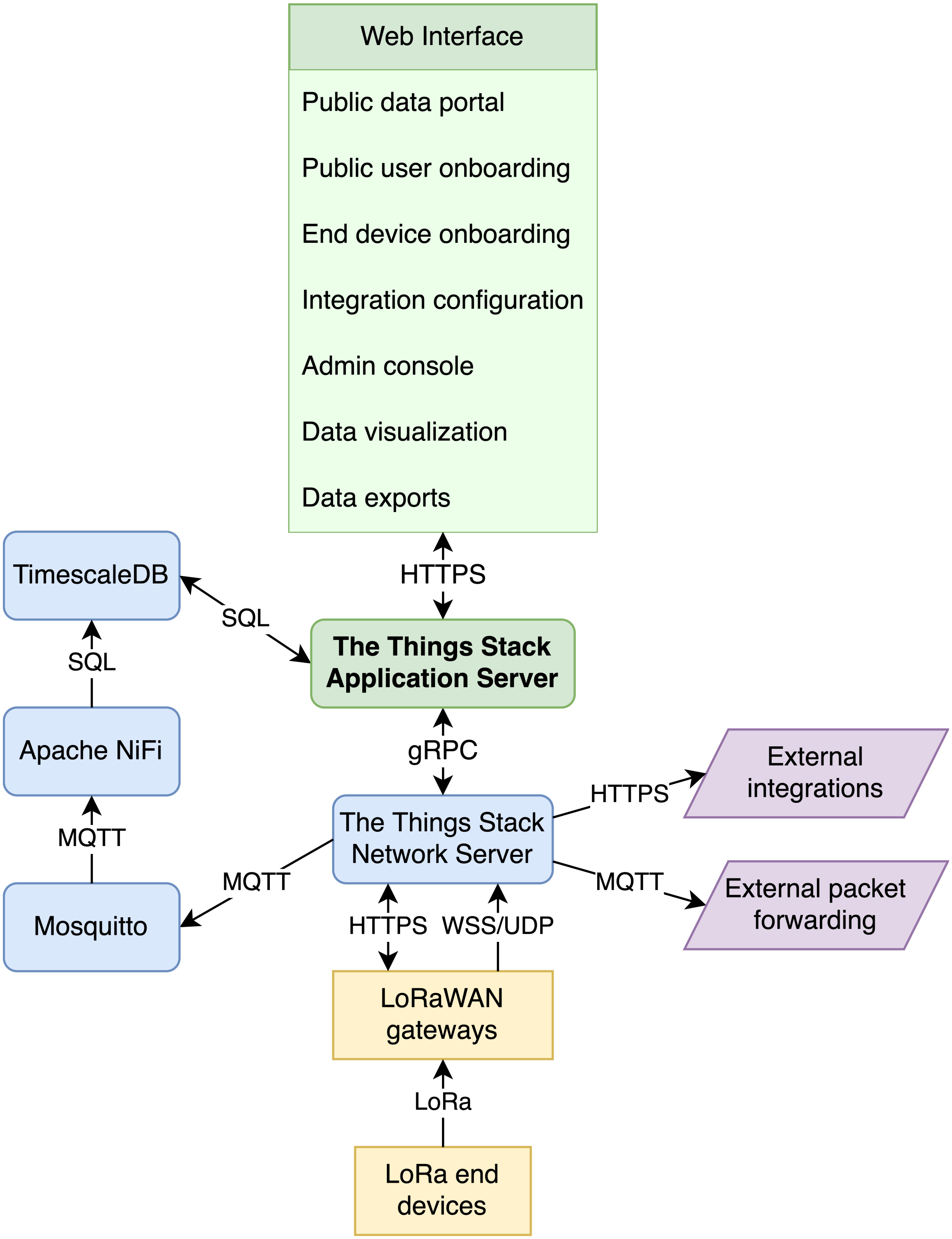


Figure 4.1: A high level diagram of the network architecture of the CSSI LoRaWAN-based climate data collection project, including all inter-process communication protocols.

UDP, TCP, WebSockets, or MQTT. Other communication protocols are supported by some gateways, but these are the most common. For the current prototype implementation, UDP is used due to its simplicity and lower overhead.

4.1.3 Network Server and Data Decoding

The Things Stack [34] (TTS) serves as the LoRaWAN network server. It receives data from the gateways and handles device management, data routing, and security. TTS decodes received packets using JavaScript-based packet decoders, also called payload formatters, which translate the encoded data (usually in hexadecimal or base64 format) into human-readable information. The Things Stack provides a repository of decoders for common devices, simplifying this process.

4.1.4 Data Forwarding via MQTT

After decoding, The Things Stack is configured to forward the data to a Mosquitto [26] MQTT broker. Although TTS includes built-in MQTT broker functionality, it creates separate MQTT brokers for each application, which would not be scalable as the project grows larger. By utilizing Mosquitto as a centralized MQTT broker, all applications can publish to the same broker, allowing downstream systems like Apache NiFi [15] to have a single MQTT subscription point instead of forcing them to subscribe to a numerous and ever-increasing amount of MQTT brokers. A diagram of this configuration can be seen in Fig. 4.2.

4.1.5 Data Processing with Apache NiFi

Apache NiFi subscribes to the Mosquitto MQTT broker using a subscription node to retrieve the data. Within NiFi, the data is processed to extract payloads, metadata, and other relevant fields. The data is then formatted into SQL statements for insertion into the database. Apache NiFi then pushes the SQL statements to the connected TimescaleDB [25] database.

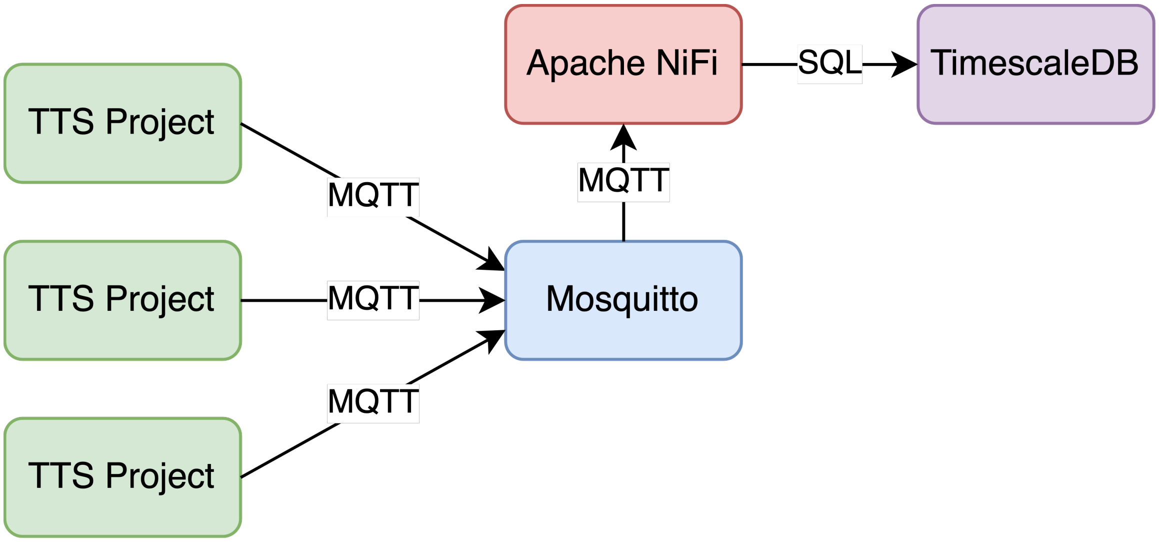


Figure 4.2: A flowchart demonstrating how the Mosquitto MQTT broker addresses the scalability concerns of having multiple TTS MQTT publishers.

4.1.6 Data Storage with TimescaleDB

The processed data is stored in TimescaleDB, a time-series extension of PostgreSQL [46]. TimescaleDB is optimized for handling large volumes of time-series data, providing efficient storage, indexing, and querying capabilities essential for environmental data analysis. This allows the front-end to efficiently query data over long periods of time.

4.1.7 Data Visualization and Export

Data visualization and export functionalities are integrated into The Things Stack by forking its open-source Open Source Edition and adding custom features. New tabs within the application pages allow users to visualize the collected data in real-time and export it for further analysis. Data can be visualized on a per-device basis, or devices can be grouped for multi-device visualization. Data can be exported to CSV files.

4.1.8 System Integration

The integration of these components creates a seamless flow from data collection to visualization. The system is designed to be scalable, secure, and flexible, accommodating the addition of new devices, gateways, and applications with minimal configuration changes. Because this system will be used by the general public, simplicity and intuitiveness are important design factors.

4.2 Implementation Details

4.2.1 LoRaWAN End Devices

The environmental sensors used in this project are LoRaWAN-compliant end devices capable of measuring various climate parameters. These devices are configured to transmit data using the LoRaWAN protocol, ensuring compatibility with the network infrastructure. The first key configuration step is end device activation via the Over-the-Air Activation (OTAA) protocol, which provides enhanced security by dynamically assigning session keys during the join process. End devices are configured to operate on the US915 Frequency Sub-Band 1 (FSB1) to match gateway configurations and to avoid interference with The Things Network, a crowd-sourced LoRaWAN project that operates on US915 FSB2. Lastly, device identifiers are entered, including the unique DevEUI, JoinEUI (AppEUI), and AppKey, which are used for device registration and authentication within The Things Stack.

4.2.2 LoRaWAN Gateways

LoRaWAN gateways serve as the bridge between the end devices and the network server. For this project, gateways are configured to forward data to The Things Stack using the UDP Packet Forwarder protocol due to its simplicity and compatibility with the hardware used.

4.2.2.1 Gateway Registration in The Things Stack

Gateways are registered in The Things Stack with a process similar to end device registration. The first piece of information required for gateway registration is the gateway ID, which is a unique identifier used internally by The Things Stack. It is important to note that this identifier is only unique to its TTS instance, so a standardized naming convention can be created. The gateway EUI is the next identifier to be entered. Similarly to DevEUIs of end devices, gateway EUIs are globally unique, and no two gateways in the world have the same EUI. The frequency plan must then be set to US915 FSB1 to match the end devices and other gateways. Lastly, authentication can be configured to the user's desired level of security. It is important to note that while gateway authentication in The Things Stack can be configured for TLS-based communication such as HTTPS gateway management and WSS packet forwarding, but not for UDP packet forwarding. Gateway authentication is not required, but it can prevent potential bad actors from intercepting unencrypted packets or spoofing gateways for man-in-the-middle attacks.

4.2.2.2 Gateway Configuration

After registration, the gateways are configured to point to the TTS network server. This involves setting the server address, ports, and protocol in the gateway's management interface. Gateway management interfaces are generally accessed via a local area network connection, but some gateways support remote management configurations. For UDP communication, the gateway is configured to forward packets to the TTS UDP listener port.

4.2.3 The Things Stack

The Things Stack Open Source edition is installed and configured to manage the LoRaWAN network, handling device registration, data decoding, and data forwarding.

4.2.3.1 Installation

The Things Stack is deployed using Docker and Docker Compose for ease of installation and management. The required files include:

- **docker-compose.yml**: Specifies the Docker images and services, including TTS, TimescaleDB, and Redis.
- **ttn-lw-stack-docker.yml**: Contains network configuration parameters such as URIs, ports, and encryption settings.

The prerequisites for installation include a server with at least 4 virtual CPUs, Docker installed, and DNS records pointing to the server's IP address. The configuration files are customized for the project and are currently stored in a private repository.

4.2.3.2 Application and End Device Management

Applications in The Things Stack group end devices and manage their data. To create an application, a series of parameters must be given. The first field is the application ID. Similarly to device IDs and gateway IDs, the application ID is an internal unique identifier used by The Things Stack to identify different applications. There is also the option of including a human-readable name field and a description field for further clarity. Although these fields are not required, they help tremendously with scaling projects such as this one. After creating the application, end devices are registered in applications by providing the necessary identifiers and configuration. End devices can have most of their details automatically filled out by the device repository, assuming that the specific model being registered is part of The Things Stack's device repository. Using this functionality, the only fields that need to be specified are the frequency plan, which is US915 FSB1 for this project, the DevEUI, JoinEUI (AppEUI), and AppKey. LoRaWAN version may also need to be specified, but this is often included in the device repository profile for end devices. The Things Stack also includes functionality to use QR codes for device registration, which automatically fills out the DevEUI, JoinEUI (AppEUI), and AppKey fields. This only works with devices that have a QR code printed on them.

4.2.3.3 Payload Formatting

Payload formatters in TTS decode binary payloads from end devices into usable data. They can be assigned at the application or device level and are typically written in JavaScript. The Device Repository provides payload formatters for common devices, reducing the need for custom development. If a device is not in the repository, custom formatters can be written based on the device's data encoding scheme. The possibility of members of the general public registering devices without a payload formatter available in the device repository remains a possibility, and it will have to be addressed by the project's maintainers.

4.2.3.4 Data Forwarding to Mosquitto

The Things Stack is configured to forward decoded data to the Mosquitto MQTT broker. This involves setting up an MQTT integration in The Things Stack that points to the Mosquitto broker's address and port. By centralizing data forwarding through Mosquitto, the system avoids the scalability issues associated with TTS's per-application MQTT servers.

4.2.4 Mosquitto MQTT Broker

Mosquitto serves as the centralized MQTT broker, facilitating the efficient and scalable forwarding of data from TTS to Apache NiFi.

4.2.4.1 Setup and Configuration

Mosquitto is installed on the server as a Docker container and configured to accept connections from The Things Stack and Apache NiFi. The key configuration steps include listener ports and security settings. Mosquitto has been configured to operate on port 30000, so both The Things Stack and Apache NiFi are pointed at that port. TLS encryption can be set up, but it was not implemented because this Mosquitto server operates on the same host as The Things Stack and Apache NiFi, so all communication is done over localhost without any external facing communication. This makes TLS irrelevant, as only project administrators have direct access to the server, and by extension, the Mosquitto container. By having TTS publish to Mosquitto and Apache NiFi subscribe to it, the system ensures that all application data flows through a single point, simplifying management and scaling.

4.2.5 Apache NiFi

Apache NiFi is used to process the data received from Mosquitto and prepare it for insertion into TimescaleDB.

4.2.5.1 NiFi Flow Configuration

Apache NiFi is configured with a data flow that includes four nodes. These nodes can be visualized in Fig. 4.3. The first node is the MQTT subscription node. The node type for this is ConsumeMQTT. This node subscribes to the Mosquitto MQTT broker to receive the relayed packets from The Things Stack. The next two nodes are data extraction and formatting nodes of the type JoltTransformJSON and EvaluateJsonPath. The JoltTransformJSON node extracts useful fields from the packet, including DevEUI, payload, metadata, and timestamp. It also adjusts the timestamp format by reducing it from nanosecond precision to millisecond precision and removing the trailing timezone. Apache NiFi is written in Java, and the basic Java timestamp format has millisecond precision. Entering a timestamp with nanosecond precision in Apache NiFi does not result in an error, but it will result in the timestamp being misinterpreted, leading to mistakes in data entry. Because the LoRaWAN standard defines the timezone as UTC, the trailing timezone definition is redundant and has the potential to cause issues when stored in a SQL database due to automatic timezone conversion behaviors found in some databases. The EvaluateJsonPath node takes these extracted and cleaned JSON fields and converts them to flowfile attributes. This allows the individual fields *dev_ eui*, *metadata*, *payload*, and *time* to be easily passed into a SQL statement in the final node. The final node is of type PutSQL, and it defines a SQL insert statement, using flowfile attributes as variables for each field. Using the JsonTreeReader and PostgreSQLDBCPConnectionPool controller services, Apache NiFi is able to push the SQL statement to a TimescaleDB server for time-series data storage.

4.2.6 TimescaleDB

TimescaleDB is configured to store the time-series data efficiently and provide automatically updating indexed views for data visualization on the web server.

4.2.6.1 Hypertables and Indexing

Hypertables are used to partition the data by time intervals, improving query performance. Indexes are created on commonly queried fields, such as timestamps and device identifiers, to optimize data retrieval. These indexes serve to improve loading times for data visualization in the web interface.

4.2.7 Data Visualization and Export

The The Things Stack Open Source Edition GitHub repository was forked to provide data visualization and data exportation functionality. New tabs have been added to the menu bar within the application pages to access the data visualization and data exportation pages.

4.2.7.1 Integration with TimescaleDB

The visualization and exportation tools interface with TimescaleDB to retrieve the necessary data. Efficient querying is ensured through the use of appropriate SQL queries and leveraging TimescaleDB's time-series optimizations.

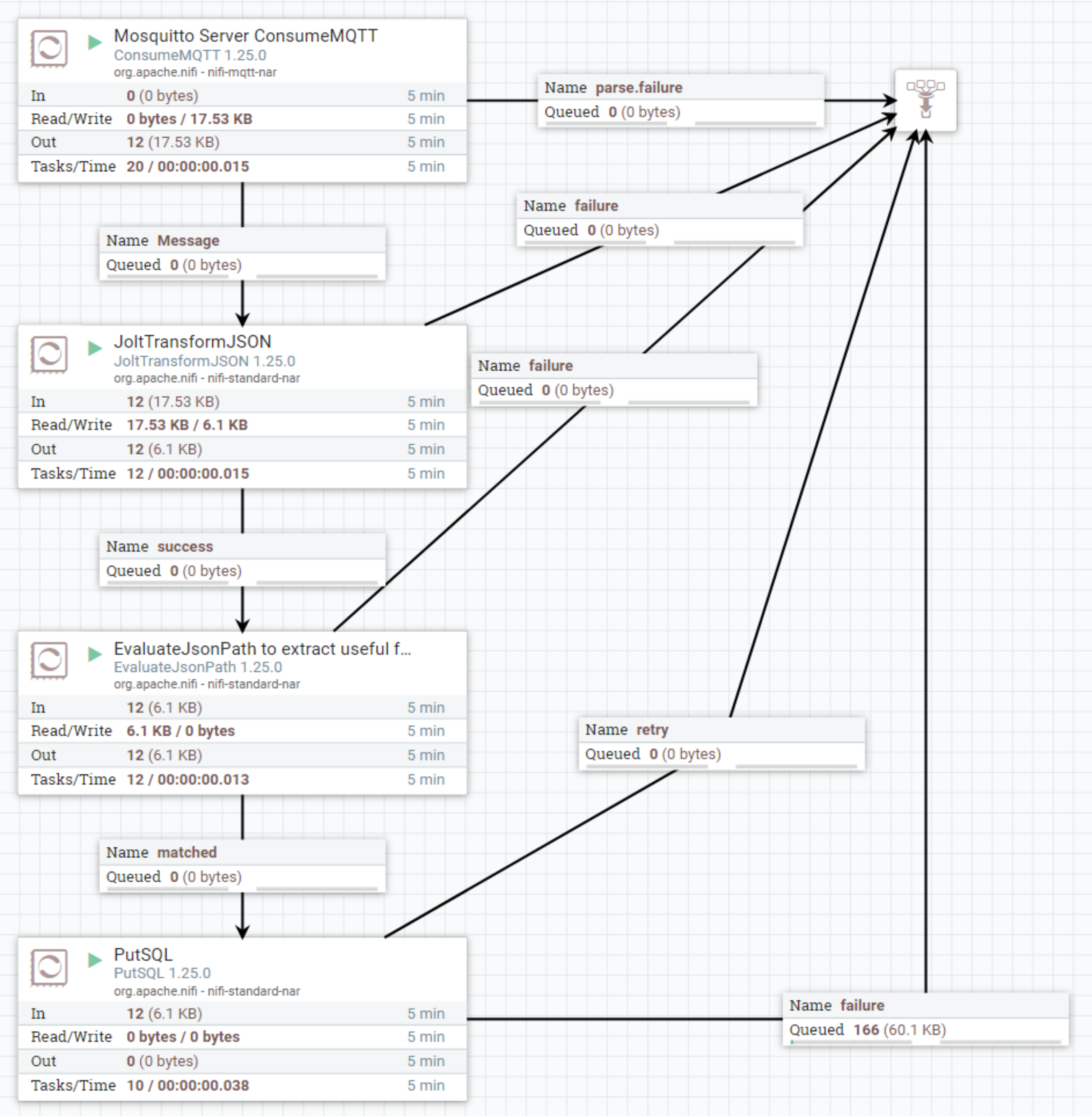


Figure 4.3: A screenshot of the current Apache NiFi node configuration.

4.3 System Integration and Workflow

The integration of the aforementioned components establishes a cohesive workflow for the climate data collection system. The end-to-end process is as follows:

1. **Data Measurement and Transmission**: LoRaWAN sensors measure environmental parameters and transmit the data to the nearest LoRaWAN gateway.
2. **Data Reception at Gateways**: The gateways receive the data and forward it to The Things Stack via WSS or UDP communication.
3. **Data Handling in The Things Stack**:
    - *Packet Reception*: The Things Stack receives the packets from the gateways.
    - *Packet Decoding*: Payload formatters decode the packets into usable data.
    - *Data Forwarding*: Decoded data is forwarded to the Mosquitto MQTT broker.
4. **Data Processing with Apache NiFi**:
    - *Data Retrieval*: NiFi subscribes to the Mosquitto MQTT broker and retrieves the data.
    - *Data Transformation*: NiFi processes the data, handling payload extraction, timestamp conversions, and formatting.
    - *Data Insertion*: NiFi inserts the data into TimescaleDB using SQL statements and the PostgreSQL controller.
5. **Data Storage in TimescaleDB**: Data is stored in hypertables, indexed, and made available for efficient querying.
6. **Data Visualization and Export**:

- *Visualization*: Users access the new tabs in The Things Stack to curate and view time-series data visualizations.
- *Export*: Users can export data to CSV files based on specific criteria for further analysis.

4.4 Scalability and Future Enhancements

The system is designed with scalability in mind, allowing the addition of new devices, gateways, and applications without significant changes to the infrastructure. Key considerations include:

- **Centralized MQTT Broker**: Using Mosquitto as a centralized broker simplifies the integration of new applications.
- **Flexible Data Processing**: Apache NiFi's modular design allows for easy modification of data flows to accommodate new data types or processing requirements.
- **Database Scalability**: TimescaleDB's time-series indexing enhancements and hypertable architecture support horizontal scaling, ensuring that the system can handle increased data volumes.
- **Open-Source Customization**: Forking The Things Stack enables the development of additional features as needed.

Chapter 5

Scenarios of Use

The LoRaWAN-based climate sensor network developed for the CSSI project is an IoT system initially designed for environmental monitoring around Lake Tahoe. However, its adaptability extends far beyond its current academic applications. The architecture of this system makes full use of the long-range and low-power capabilities of LoRaWAN technology, opening up many possibilities for use by stakeholders from different backgrounds. These stakeholders range from academic researchers to citizen scientists, local communities, and even industries. This chapter explores the various real-world scenarios in which this system could be deployed, highlighting its potential impact across different domains. Several examples of the system's web portal user interface are shown in Chapter 6 in connection with the usability study designed and conducted for this thesis.

5.1 Academic Research and Environmental Monitoring

The primary users of the system during its initial phase are academic researchers - particularly graduate students, professors, and researchers within the University of Nevada, Reno's College of Agriculture, Biotechnology & Natural Resources (CABNR), as well as the Desert Research Institute and other organizations within the Nevada System of Higher Education (NSHE). Their primary use of the platform involves studying the environmental conditions throughout the Lake Tahoe basin. The Lo-

RaWAN sensors deployed within the region collect a variety of climate metrics - including air temperature, humidity, soil moisture, soil temperature, snow depth, and wind-driven tree sway. These metrics provide a detailed picture of both short-term changes and long-term trends in the weather and climate of the area.

5.1.1 Climate Research in the Lake Tahoe Basin

Lake Tahoe is an ecologically sensitive area, known for its clear waters and scenic mountain environment. However, it is also highly vulnerable to climate trends, regional weather events, urban development, and other major disturbances. Monitoring the Tahoe basin for predicted impacts such as changing seasonal snowpacks, urban heat sources, flash droughts, and elevation-dependent temperature shifts is of great interest. The deployment of LoRaWAN climate sensors allows researchers to continuously monitor these factors in real-time and with higher spatial diversity, offering direct observational scientific data on local-scale climate conditions.

Graduate students whose research is focused on climate science will be able to deploy sensors at various locations around Lake Tahoe to create an environmental monitoring network. For example, they can install sensors on forested slopes to assess how temperature and humidity fluctuate across elevations or place soil moisture sensors across gradients of forest disturbance to track the movement of water through the soil. This kind of granular data collection enables researchers to refine landscape models that predict how climate and disturbance impacts to Lake Tahoe may impact water availability, vegetation health, and even local wildlife, all of which are directly related to the environmental and economic health of the Tahoe region.

Furthermore, the data generated by these sensors can be used to compare the environmental conditions among different microclimates around the lake. Researchers can study the variation in soil moisture and temperature in changing seasons, evaluate how changes in snowpack depth correlate with location and other environmental events, and understand how snowfall affects the water cycle and overall forest health. These insights will contribute to ongoing research on forest ecology, hydrology, and climate adaptation, and provide empirical evidence that can inform policy decisions on environmental protection and resource management.

5.1.2 Wildland Fire Science

Wildfire management is an important area of research in Northern Nevada and Northern California, particularly due to the increasing frequency and intensity of wildfires driven by climate change, forest management, urban expansion, and other risk factors that may yet be unknown. The Lake Tahoe basin, like the rest of the inland forested regions of the West, is highly vulnerable to high-severity wildfires due to over a century of fire exclusion and unmanaged vegetation growth and fuels accumulation. These conditions, when combined with drought conditions and high-wind events, can be disastrous.

The CSSI LoRaWAN-based sensor network is a valuable tool for researchers who are focused on understanding the environmental conditions that contribute to wildfires and their impacts. By deploying sensors throughout the basin, researchers can analyze real-time data on key variables such as air temperature, humidity, soil moisture, and wind patterns - factors that are directly related to fire risk [10]. For example, prolonged periods of low humidity and high temperatures, coupled with dry soil conditions and high winds, can significantly increase the risk of wildfires [1, 21, 38, 42].

Researchers studying wildfire science could use the system to monitor how these conditions evolve throughout the year, identifying the patterns that precede major wildfire events. By installing sensors in fire-prone areas, such as densely forested regions, the system can provide early warnings about deteriorating conditions, allowing authoritative bodies to take preparatory or preventative measures. Additionally, the data collected can inform long-term land management practices, such as identifying areas that need targeted fuel reduction or controlled burns to mitigate fire risk.

In addition to local monitoring, the CSSI LoRaWAN platform allows researchers to share data with other institutions and agencies working on wildfire prevention.

By collaborating with fire prevention authorities and other research institutions, the platform can contribute to larger-scale efforts aimed at modeling fire behavior and predicting fire impacts to different parts of the landscape. Furthermore, researchers could analyze historical data from the sensor network to retrospectively study the environmental conditions that led to past wildfires, improving their understanding of how to predict and manage future fires.

The integration of climate data into wildfire management and operations is a crucial step towards safeguarding the Lake Tahoe region from the ever-increasing threat of wildfires. With LoRaWAN's ability to provide continuous, real-time data across large areas, the CSSI project equips researchers with the tools they need not only to study wildfire risk factors but also to help proactive strategies that could save lives and protect the environment.

5.1.3 Agricultural Research and Precision Farming

Beyond climate monitoring, the versatility of the system extends to agricultural research. CABNR can use LoRaWAN sensors in experimental agricultural plots to monitor variables such as soil moisture, temperature, and conductivity, enabling a detailed analysis of crop health and water use. This use case is particularly relevant given the increasing importance of precision agriculture, which seeks to optimize farming practices by providing real-time data on environmental conditions. Machine learning is increasingly being used to optimize agriculture, and having extensive monitoring data is a prerequisite to success in that approach.

Researchers could deploy sensors in test plots to monitor how different irrigation techniques affect water efficiency. By comparing soil moisture levels in plots using traditional irrigation methods with those using modern sensor-driven systems, researchers can evaluate the effectiveness of precision irrigation systems. This could potentially lead to significant water savings, especially in arid or semi-arid regions, where water scarcity is an ongoing concern. Furthermore, data collected from these sensors can reveal which crop varieties are better suited to changing climate conditions, contributing to global efforts to ensure food security in the face of climate change.

This precision farming approach has the potential to improve the way agricultural research is conducted, moving away from labor-intensive manual measurements and towards automated, sensor-driven data collection. In turn, these data can be fed into larger-scale predictive models of agricultural productivity, helping researchers understand how changes in temperature, precipitation, and soil conditions will affect crop yields and food production.

5.2 Democratizing Climate Science through Citizen Science

A central goal of this project is to democratize climate science by allowing the public to participate in data collection and analysis. After the initial phase, the platform will be opened to the general public, enabling hobbyists, citizen scientists, and local communities to deploy their own LoRaWAN sensors and contribute to the growing body of climate data. This form of crowdsourced environmental monitoring empowers individuals and communities to take an active role in understanding and addressing the environmental challenges they face.

5.2.1 Backyard Climate Monitoring

One potential use case for the public is the installation of sensors in private properties to monitor local microclimates. For example, a hobbyist gardener could use soil moisture sensors to track the optimal watering schedule for their garden, while simultaneously collecting air temperature and humidity data to predict frost risks. These sensors would feed data directly into the web platform, where users can visualize trends over time and make informed decisions about their gardening practices.

This setup not only provides users with practical benefits; it also contributes to larger citizen science efforts. By sharing their data with local environmental organizations or schools, individuals can help create a more detailed map of the climate conditions in their area. Over time, this data can be aggregated and used to track long-term trends in temperature, precipitation, and soil conditions, providing valuable insights that can inform local environmental planning and management.

5.2.2 Community Environmental Monitoring

Communities living near sensitive environmental areas, such as Lake Tahoe, could also deploy LoRaWAN sensors to monitor specific environmental conditions that directly impact their quality of life. For example, residents in fire-prone areas could install air temperature and humidity sensors to track conditions during fire season, allowing them to better anticipate wildfire risks. Similarly, communities facing water scarcity could deploy soil moisture sensors to monitor the effects of drought on local vegetation.

In this hypothetical scenario, the platform serves as a tool for empowering communities to take control of their environmental futures. Local governments and advocacy groups could use the collected data to support policies that aim to mitigate the effects of climate change, such as stricter water conservation measures or improved wildfire prevention strategies. In addition, the scalability of the system allows the deployment of large sensors networks in wide geographical areas, providing communities with a powerful tool to track environmental changes in real-time. The CSSI project seeks to encourage people to engage with environmental issues on a personal level, contributing to a local cultural shift towards sustainability and conservation.

5.3 Industrial and Commercial Applications

While the initial focus of the project is on climate science and environmental monitoring, the flexibility of the LoRaWAN platform makes it suitable for a wide range of industrial and commercial applications. LoRaWAN's long-range, low-power capabilities have already been adopted in various sectors, from agriculture to datacenters, and the platform developed here can be easily adapted to support these use cases.

5.3.1 Livestock Management

Livestock management is another area where LoRaWAN can have a significant impact. GPS-enabled LoRaWAN collars can be used to track the location and health of livestock in large grazing areas, providing real-time data on animal movement and behavior. This information can help farmers prevent livestock losses, detect early signs of illness, and optimize grazing patterns to improve pasture management.

For example, a farmer could install GPS sensors on livestock to track the movements of herds and quickly identify any animals that get separated from the herd. They could also use historical GPS data to identify which plots of land are grazed the most, providing valuable insights on how to best rotate different plots of land for grazing.

5.3.2 Smart City Infrastructure

LoRaWAN's scalability and low-power characteristics make it an ideal solution for smart city applications, where sensor networks are used to monitor and manage urban infrastructure. In a smart city scenario, LoRaWAN sensors could be deployed throughout a city to monitor air quality, traffic flow, parking space availability, and other key metrics that affect urban life [49].

For example, cities could use LoRaWAN sensors to monitor air quality in real-time, providing residents with up-to-date information on pollution levels. This data could be used to inform public health initiatives, such as issuing warnings during days of high pollution or adjusting traffic patterns to reduce emissions. Similarly, parking sensors could be installed in busy areas to track space availability, helping drivers find parking more quickly and reducing traffic congestion. Many airports have already implemented parking sensors to provide live maps of free parking spots, and the LoRaWAN platform provided by the CSSI project offers a potentially cheaper and more sustainable way to accomplish this.

The open-source architecture of the system allows the integration of a wide variety of sensors, meaning that cities can customize their sensor networks to meet their specific needs. Whether it's monitoring water levels in storm drains to prevent flooding or tracking energy usage in public buildings, the platform provides a scalable and cost-effective solution to manage urban infrastructure.

5.4 Expanding Beyond Climate Science

The LoRaWAN platform developed for the CSSI project has the potential to be used in a wide range of applications beyond climate science. Its modular design allows for the straightforward integration of new sensor types, making it adaptable to various industries and use cases. Some of the most promising areas for expansion include home automation, security, and logistics.

5.4.1 Home Automation and Security

In the growing field of home automation, LoRaWAN sensors can be used to monitor and control various aspects of the home environment. For example, door and window sensors could be installed to provide real-time security alerts to homeowners, while temperature and humidity sensors could be used to optimize home heating and cooling systems. By integrating these sensors with smart home systems, homeowners can automate tasks such as adjusting the thermostat based on outdoor conditions or receiving notifications when a door is left open.

The long-range capabilities of LoRaWAN also make it suitable for use in rural or remote areas, where traditional Wi-Fi or cellular networks may not be available or may not meet sufficient distance requirements. In such areas, homeowners could install security sensors on gates or fences to monitor entry points on large properties, receiving alerts on their mobile devices even when they are far from home.

5.4.2 Logistics and Supply Chain Management

The logistics industry stands to benefit greatly from the deployment of LoRaWAN sensors, particularly to track the movement and condition of goods throughout the supply chain. For example, temperature and humidity sensors could be used to monitor the conditions inside shipping containers, ensuring that perishable goods such as food and pharmaceuticals are transported under optimal conditions. GPS sensors integrated into shipping containers could track the location of goods in real-time, providing valuable data on delivery times and preventing losses or theft.

The flexibility of the platform means that it can be easily integrated into existing logistics systems, providing companies with a cost-effective way to improve supply chain visibility and efficiency. By enabling real-time monitoring of goods and assets, LoRaWAN sensors can help reduce the risk of spoilage, delays, and other issues that can disrupt supply chains.

5.5 Future Directions and Scalability

As the CSSI project progresses and the system is opened to the public, the scalability of the LoRaWAN platform will become increasingly important. The system is designed to support large-scale deployments, with the ability to integrate thousands of sensors across wide geographical areas. Its use of centralized data management and processing tools such as Apache NiFi [15] and TimescaleDB [25] ensures that it can handle large volumes of time-series data efficiently.

One of the key challenges in expanding the platform will be ensuring that it remains user-friendly and accessible to non-expert users. While the platform's current design includes a simple web interface for visualizing and exporting data, additional features will be needed to support more advanced use cases. For example, the ability to create custom dashboards or generate automated alerts based on specific sensor readings would enhance the platform's usability for both public and commercial users.

Looking ahead, the integration of machine learning could further enhance the capabilities of the platform. By analyzing the vast amounts of data generated by the sensors, machine learning models could identify patterns and trends that may not be immediately apparent to human users. This could enable predictive analytics for applications such as climate modeling, infrastructure maintenance, and supply chain optimization, making the platform even more valuable to its users.

Chapter 6

Web Portal Evaluation

6.1 Objective

The primary objective of this study is to evaluate the usability of the web interface designed for the CSSI project. The current interface is based on The Things Stack Open Source Edition [34] (TTS), a platform originally intended for users with prior IoT and LoRaWAN experience. Although the default functionality of TTSS is well suited for advanced users, the interface presents challenges for novice users, particularly those outside the IoT domain. As the target user base transitions to include climate researchers and eventually the general public, it is imperative to ensure that the interface is intuitive and accessible for users with varying levels of technical expertise.

TTSS was forked, and the web interface has been slightly modified to align with the requirements of the CSSI project. Although the author of this thesis designed and implemented the initial version of the web interface, a revised and extended version developed by fellow CSE master's students Kaden Nesch and Nikhil Sharma has been employed for this usability study. The most notable customizations include:

1. **Renaming Applications to Projects**: This change simplifies terminology to match the expectations of the research community, as suggested by a co-PI of the grant.

2. **Addition of an Export Data Tab**: This tab allows users to export data in

CSV or JSON format. Users can specify sensors, date ranges, and sensor fields (e.g., temperature, soil moisture).

3. **Addition of a Data Visualization Tab**: This tab supports customizable line-graph visualizations of sensor data. Users can aggregate data across specified time frames or disable aggregation for detailed observations.

These modifications aim to better serve the needs of climate scientists by focusing on ease of data retrieval and interpretation. However, the core interface remains largely unchanged from the advanced user-oriented design of TTSS. This introduces a critical need to evaluate the platform's usability, focusing on its functionality, efficiency, and user-friendliness for less experienced users.

6.1.1 Importance of Usability

The initial users of the interface will be climate researchers, many of whom have no prior exposure to LoRaWAN technologies. This discrepancy between user background and interface design creates a steep learning curve. Terminologies such as 'uplinks', 'end devices', and 'gateways', while essential in TTSS, can confuse novice users unfamiliar with the technical intricacies of LoRaWAN systems. Furthermore, the research team aims to open the platform to public participation in the future. The ultimate goal is to enable citizen scientists to deploy their own sensors, manage their data, and contribute to climate research initiatives.

This challenge is well-documented in the IoT community. For example, Carthen et al. [7] noted that while LoRaWAN's open standard and energy efficiency make it an ideal candidate for scalable systems, its reliance on complex terminologies and configurations often alienates novice users. These barriers highlight the importance of iterative design improvements to ensure inclusivity and accessibility for broader userbases.

6.1.2 Study Rationale

By evaluating the usability of the current interface, this study aims to identify and address specific pain points experienced by novice users. The insights gained will guide iterative design improvements, ensuring that the platform is aligned with the needs of diverse user groups. Key evaluation goals include:

- Identifying unintuitive interface elements.
- Assessing the effectiveness and limitations of the Export Data and Data Visualization tabs.
- Understanding the extent to which the platform meets the needs of climate researchers and prospective citizen scientists.
- Collecting qualitative feedback to inform design enhancements.

This usability evaluation is structured as part of a long-term plan to iteratively refine the platform. By systematically addressing identified challenges, the research project aims to create a user-friendly interface that bridges the gap between LoRaWAN technology and its application in climate research and public engagement. Broader Impact

As climate change continues to demand innovative solutions, democratizing access to scientific tools is more critical than ever. A user-friendly interface for TTSS not only supports researchers, but also encourages public participation in climate science. Citizen science initiatives thrive on accessible platforms, where users can independently explore, analyze, and share environmental data. By improving the usability of the TTSS-based interface, this project contributes to a vision of inclusive participatory science, making advanced technology accessible to everyone.

6.2 Experimental Setup

This section outlines the experimental setup for evaluating the usability of the CSSI project's web interface. The evaluation was conducted in a controlled environment to systematically analyze the interactions of the participants with the interface, identify usability challenges, and gather actionable feedback for iterative design improvements.

The design of this study was informed by previous research that highlighted the importance of usability in IoT platforms, particularly for users with limited technical expertise. Ramson et al. [39] emphasized that intuitive designs are essential for reducing cognitive load, particularly when users are unfamiliar with the platform's terminology or workflows. Their findings reinforced the need to evaluate the intuitiveness of the CSSI web interface and its ability to guide novice users through tasks such as exporting data, visualizing metrics, and adding sensors.

The combination of quantitative metrics (e.g., navigational errors and participant ratings) and qualitative feedback (e.g., participant comments) ensured a holistic understanding of the platform's strengths and weaknesses. This approach enabled the identification of specific areas for improvement, which will help shape future iterations of the web interface.

6.2.1 Participants

Eighteen participants from the University of Nevada, Reno were recruited for the study. These participants were primarily students and alumni, representing a variety of educational and technical backgrounds. The group consisted of nine participants aged 18-24 and nine participants aged 25-34 years. Educational qualifications ranged from high school diplomas to master's degrees, with three participants reporting high school as their highest level of education, three holding associate's degrees, nine holding bachelor's degrees, and three holding master's degrees. The fields of study reported by the participants included Computer Science & Engineering (12 participants), Mechanical Engineering (3 participants), Electrical Engineering (1 participant), Nursing (1 participant), and Social Work (1 participant).

This pool of participants was chosen to include users with varying familiarity with IoT systems, ranging from experienced data scientists to individuals in nontechnical fields. Such diversity allowed for a comprehensive evaluation of the web interface's usability, with insights that could be generalized to both technically proficient and novice users. It also allowed the study to anticipate challenges that climate researchers and citizen scientists, the primary target audience of the platform, might encounter.

6.2.2 Setup

The study was carried out in the William Pennington Engineering building at the University of Nevada, Reno, under controlled conditions designed to replicate realistic user scenarios while minimizing external distractions. Each participant worked individually on a laptop running Ubuntu 24.04.1 [28]. The laptop featured a standard integrated webcam and a wired USB mouse, ensuring consistent hardware conditions for all participants. SimpleScreenRecorder [2] was installed on the laptop to capture on-screen activity during tasks, enabling detailed post-study analysis of navigation patterns, task completion times, and errors.

The web interface for the CSSI project was hosted on a live production server at `https://cssi.unr.dev`, ensuring that participants interacted with the same environment as future users of the platform. In addition to the digital setup, participants were provided with a printed instruction sheet detailing the tasks they needed to complete. This sheet included a QR code required for one of the tasks, which could be scanned using the laptop's integrated webcam. This setup simulated real-world scenarios where users might reference external documentation while interacting with the platform. These preparations ensured that the study captured realistic user behaviors and interactions with the platform while maintaining consistency across all participant sessions.

6.2.3 Procedure

The procedure for the study was carefully structured to ensure consistency across participants and to collect both quantitative and qualitative data. Each session followed these steps:

1. Pre-study Phase: Participants were briefed on the study objectives and asked to

sign an informed consent form. This was followed by a pre-study questionnaire that collected demographic data, assessed participants' familiarity with IoT platforms, and evaluated their general technical proficiency. The demographic information collected included:

- Age group.
- Highest level of education completed.
- Primary field of research or study.
- Proficiency with computers (rated on a scale from 1 to 7).
- Proficiency with IoT platforms (rated on a scale from 1 to 7).
- Frequency of using data analysis tools (Never, Rarely, Monthly, Weekly; Daily).
- Types of data analysis tools frequently used (free response).

2. Task Phase: After completing the pre-study questionnaire, participants received a brief verbal overview of the CSSI project and instructions for completing three tasks on the web interface. These tasks were:

   - Task A: Export data from a specific sensor within a project, focusing on a specific date range and sensor metric. Participants needed to export the data in CSV format.
   - Task B: Visualize data from a specific sensor over a specific time range with a chosen aggregation level, focusing on the intuitiveness of the data visualization interface.
   - Task C: Add a new LoRaWAN sensor to a project using a QR code, requiring participants to navigate the LoRaWAN Device Repository and input sensor-specific details correctly.

   Task orders were fully counterbalanced to account for potential order effects. This involved assigning three participants to each of the six possible task sequences, ensuring that the order in which tasks were completed did not bias the results.

3. Post-study Phase: Upon completion of the tasks, participants completed a post-study questionnaire. This survey evaluated the ease of navigation, functionality, and usability of specific features of the web interface using a 7-point Likert scale. Participants were also invited to provide open-ended feedback on potential improvements and any additional comments about their experience.

Screen recordings captured during the task phase were analyzed post hoc to identify navigation patterns, task completion times, and errors, offering a detailed understanding of usability challenges.

6.2.4 Tasks

The tasks were designed to evaluate key functionalities of the web interface, focusing on ease of navigation, data handling, and hardware integration. Each task targeted a specific feature of the platform:

Task A: Participants were asked to export data from the sensor *dragino-soil-moisture1* within the *Dascalu House* project. The task required them to select a date range from November 1, 2024 at midnight to November 7, 2024 at midnight, select the *Soil Temperature* field, and export the data in CSV format. This task assessed the intuitiveness of the data export feature. Task B: Participants visualized the *Temperature* field of the sensor *laird-temp2* from the *Dascalu House* project over the past 6 months. They were instructed to use a 7-day aggregation. This task evaluated the functionality and usability of the data visualization tab, focusing on the ease of navigating the data visualization interface and the interpretability of the data. Task C: Participants added a new sensor to the *Zach House* project using a provided QR code. The key sensor details included the brand (Laird Connectivity, Inc.), the model (Sentrius RS1xx Multi Sensor), and the frequency band (United States 902–928 MHz, FSB 1). This task tested the interface's ability to guide users through complex hardware integration steps.

6.3 Results

This section presents the findings of the usability study of the CSSI project's web interface. It includes data collected through pre- and post-study questionnaires, direct observations from participant screen recordings, and analysis of user feedback.

6.3.1 Task Analysis

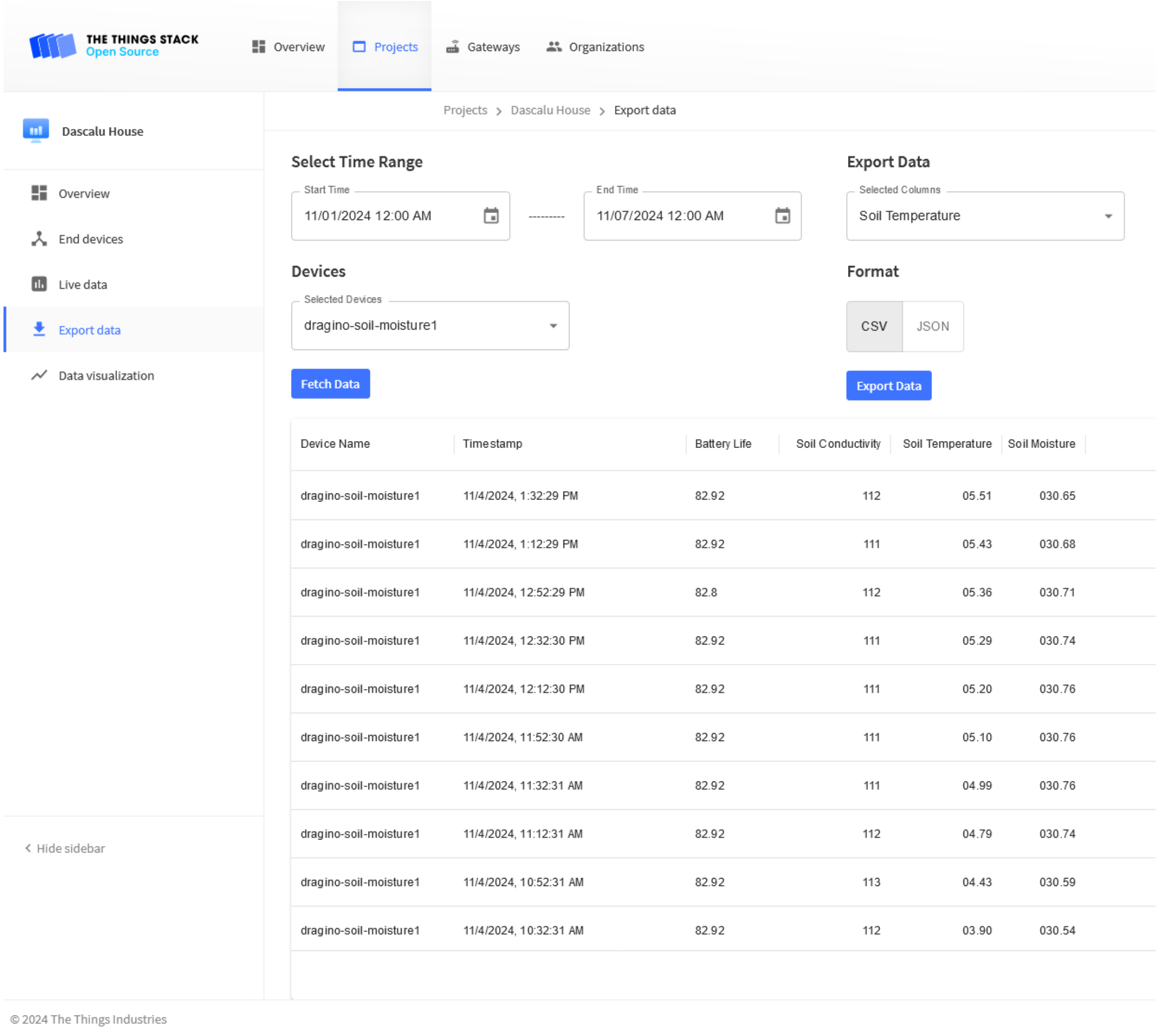
The tasks performed by the participants provided critical insight into the strengths and weaknesses of the current CSSI web interface. The following analysis includes observations from screen recordings and participant feedback for each task.

6.3.1.1 Task A: Exporting Data

Task A required participants to export data from the sensor *dragino-soil-moisture1* within the *Dascalu House* project. Participants were instructed to select a specific date range, include only the *Soil Temperature* metric, and export the data in CSV format. A screenshot of the data export interface can be seen in Fig. 6.1. Observations from screen recordings included:

- **Navigation Challenges**: 7 participants initially navigated to the specific sensor's page instead of using the *Export data* button in the project sidebar menu. Intuitively, it would make sense for an export data button to exist on every sensor's page, and users wanting to export the data of a specific sensor expect to see this functionality on the sensor's page. 10 participants spent a significant amount of time locating the *Export data* button in the project sidebar menu, further highlighting its unintuitive placement.

- **Sensor Field Selection**: - 14 participants exported all sensor fields rather than selecting *Soil Temperature* specifically. This can be attributed to the drop-down menu for sensor field selection being hidden until the device and date/time range



Select Time Range

Start Time: 11/01/2024 12:00 AM

End Time: 11/07/2024 12:00 AM

Devices

Selected Devices: dragino-soil-moisture1

Export Data

Selected Columns: Soil Temperature

Format: CSV

Device Name	Timestamp	Battery Life	Soil Conductivity	Soil Temperature	Soil Moisture
dragino-soil-moisture1	11/4/2024, 1:32:29 PM	82.92	112	05.51	030.65
dragino-soil-moisture1	11/4/2024, 1:12:29 PM	82.92	111	05.43	030.68
dragino-soil-moisture1	11/4/2024, 12:52:29 PM	82.8	112	05.36	030.71
dragino-soil-moisture1	11/4/2024, 12:32:30 PM	82.92	111	05.29	030.74
dragino-soil-moisture1	11/4/2024, 12:12:30 PM	82.92	111	05.20	030.76
dragino-soil-moisture1	11/4/2024, 11:52:30 AM	82.92	111	05.10	030.76
dragino-soil-moisture1	11/4/2024, 11:32:31 AM	82.92	111	04.99	030.76
dragino-soil-moisture1	11/4/2024, 11:12:31 AM	82.92	112	04.79	030.74
dragino-soil-moisture1	11/4/2024, 10:52:31 AM	82.92	113	04.43	030.59
dragino-soil-moisture1	11/4/2024, 10:32:31 AM	82.92	112	03.90	030.54

Figure 6.1: The Export Data page of the project’s web interface.

were entered. While it does make sense for the field selection drop-down menu to be immutable until one or more sensors are selected due to the inability to populate the menu with sensor fields without knowing the respective sensor(s), completely hiding the menu confuses users. In post-study questioning, most users indicated that they did not even notice the menu appear due to it being initially hidden.

- **Date/Time Selection Issues**: - 4 participants failed to notice the interactive calendar button in the date/time selection menu. This led them to manually input the date and time. While this functionally is not a problem, it shows that this functionality is not intuitive.

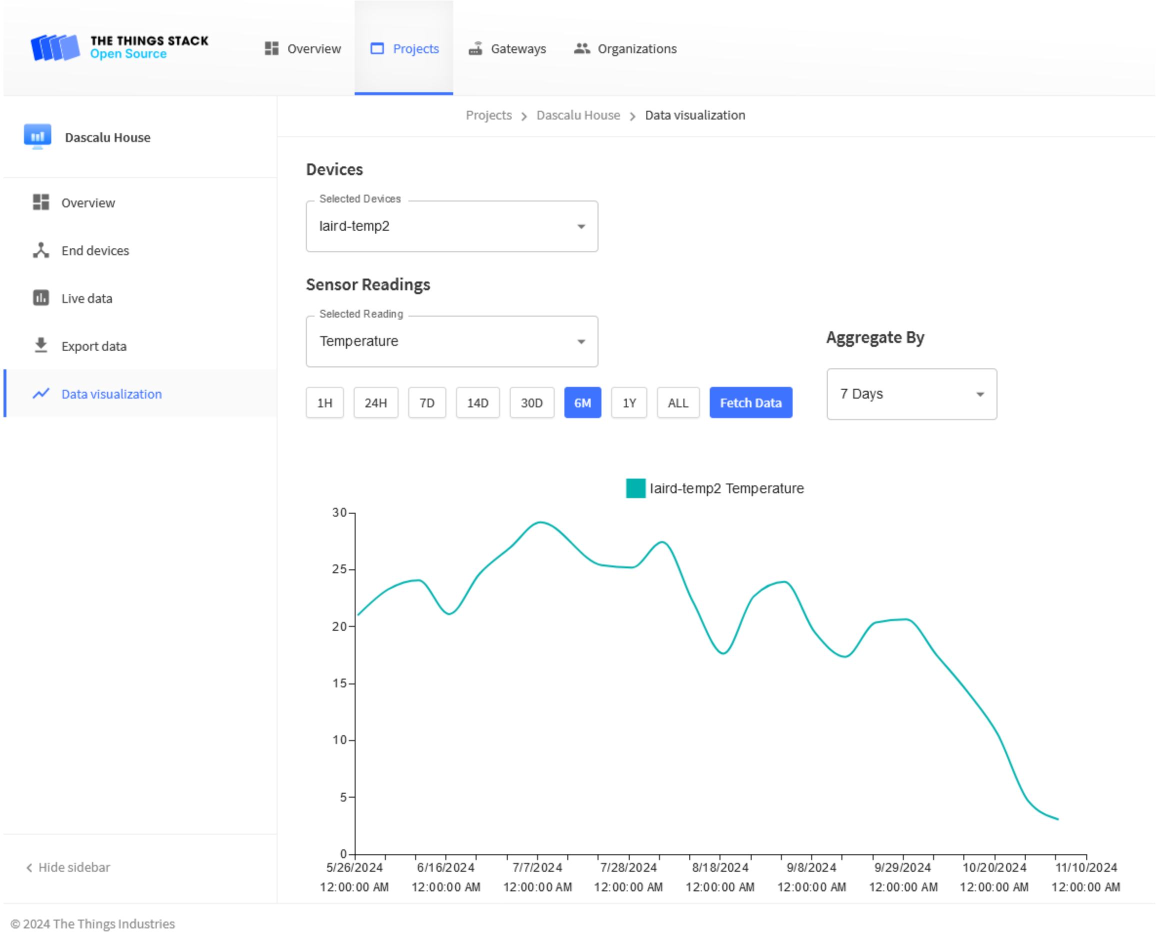
These results suggest that conditionally hidden UI elements hindered task completion and that the project sidebar menu is not the best place for functionality that frequently involves only a single sensor.

6.3.1.2 Task B: Data Visualization

Task B involved visualizing the *Temperature* field from the *laird-temp2* sensor over a 6-month period with a 7-day aggregation. An example of what this looks like in the project's web interface can be seen in Fig. 6.2. This task assessed the functionality and usability of the data visualization feature.

- **Navigation Challenges**: 7 participants initially navigated to the specific sensor's page instead of using the *Data visualization* button in the project sidebar menu. 8 participants spent a significant amount of time locating the *Data visualization* button in the project sidebar menu, further highlighting its unintuitive placement.

- **System Feedback**: 6 participants were confused after clicking the *Fetch data* button due to a lack of loading indicators or other system feedback during graph generation.



Devices

Selected Devices: laird-temp2

Sensor Readings

Selected Reading: Temperature

Aggregate By: 7 Days

Figure 6.2: The Data Visualization page of the project's web interface.

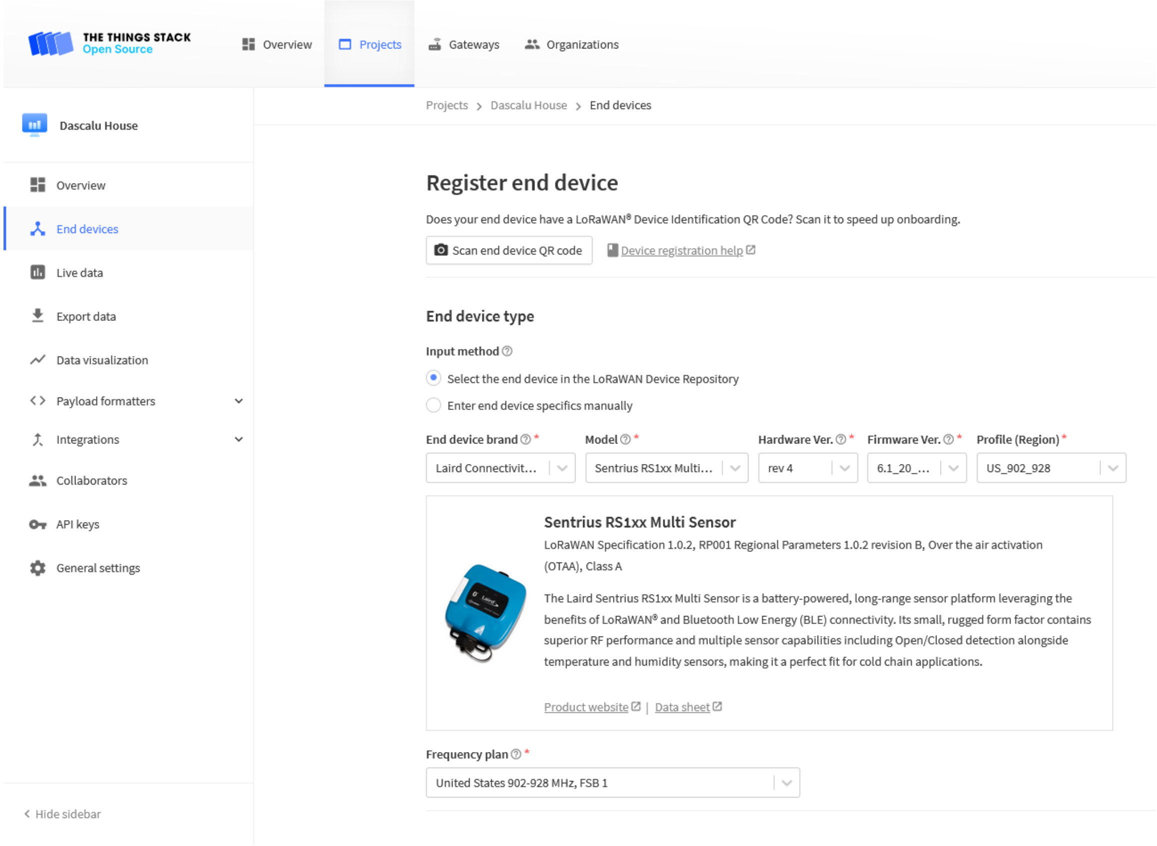
The absence of a progress indicator led to user frustration, as they were unsure if the system was processing their request. The navigational challenges mirrored those in Task A, suggesting that the placement of sidebar buttons needs to be reconsidered. Due to the similarity of these navigational challenges and the immediate proximity of the *Export data* and *Data visualization* buttons, it is likely that the numbers would be even higher if not for the order effect. Task counterbalancing seems to have worked well here in balancing the results across both Task A and Task B.

6.3.1.3 Task C: Adding a New Sensor

Task C required the participants to add a new sensor to the *Zach House* project using a QR code. Because LoRaWAN QR codes do not contain all of the required information, participants were given instructions on how to fill out the remaining fields, including instructions on how to select the correct sensor model from the device repository, which can be seen in Fig. 6.3. Observations included:

- **Finding the *Register end device* Button**: 12 participants struggled to locate this button due to its placement at the bottom right of the project page and its ambiguous label. The placement of the *Register end device* button can be seen in Fig. 6.4.
- **QR Code Feedback**: 6 participants were confused after scanning the QR code because there was minimal visual feedback indicating a successful scan.

These challenges underscore the need for clearer button labeling and immediate feedback mechanisms to guide users through the sensor registration process. Although *Register end device* may be intuitive to those who have prior experience with LoRaWAN platforms, it is clearly unintuitive to people who do not meet those criteria.



Register end device

Does your end device have a LoRaWAN® Device Identification QR Code? Scan it to speed up onboarding.

Scan end device QR code | Device registration help

End device type

Input method

● Select the end device in the LoRaWAN Device Repository

○ Enter end device specifics manually

End device brand *	Model *	Hardware Ver. *	Firmware Ver. *	Profile (Region) *
Laird Connectivit…	Sentrius RS1xx Multi…	rev 4	6.1_20_…	US_902_928

Sentrius RS1xx Multi Sensor

LoRaWAN Specification 1.0.2, RP001 Regional Parameters 1.0.2 revision B, Over the air activation (OTAA), Class A

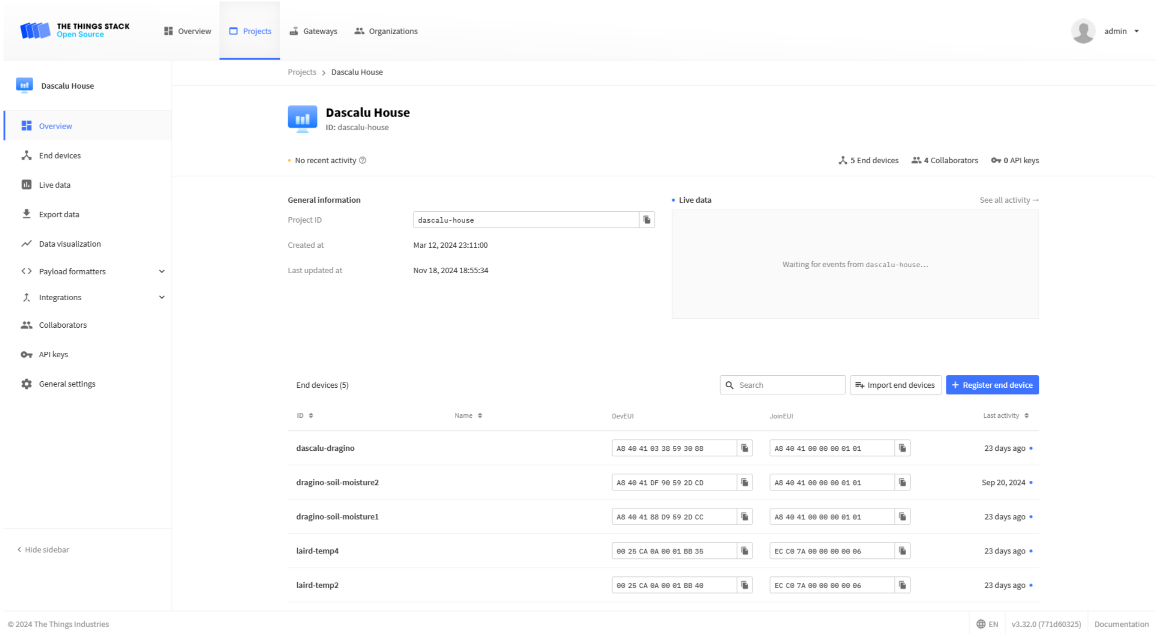
The Laird Sentrius RS1xx Multi Sensor is a battery-powered, long-range sensor platform leveraging the benefits of LoRaWAN® and Bluetooth Low Energy (BLE) connectivity. Its small, rugged form factor contains superior RF performance and multiple sensor capabilities including Open/Closed detection alongside temperature and humidity sensors, making it a perfect fit for cold chain applications.

Product website | Data sheet

Frequency plan *

United States 902-928 MHz, FSB 1

Figure 6.3: The Register End Device page of the project's web interface.



Dascalu House
ID: dascalu-house

General information

Project ID: `dascalu-house`
Created at: Mar 12, 2024 23:11:00
Last updated at: Nov 18, 2024 18:55:34

Waiting for events from `dascalu-house...`

End devices (5)

ID	Name	DevEUI	JoinEUI	Last activity
dascalu-dragino		A8 40 41 03 38 59 30 88	A8 40 41 00 00 00 01 01	23 days ago
dragino-soil-moisture2		A8 40 41 DF 90 59 2D CD	A8 40 41 00 00 00 01 01	Sep 20, 2024
dragino-soil-moisture1		A8 40 41 88 D9 59 2D CC	A8 40 41 00 00 00 01 01	23 days ago
laird-temp4		00 25 CA 0A 00 01 BB 35	EC C0 7A 00 00 00 00 06	23 days ago
laird-temp2		00 25 CA 0A 00 01 BB 40	EC C0 7A 00 00 00 00 06	23 days ago

Figure 6.4: A screenshot of the project's web interface, displaying the unintuitive placement of the *Register end device* button.

6.3.2 Questionnaire Results

The pre-study questionnaire aimed to gather relevant background characteristics about the technical expertise and familiarity of the participants with IoT platforms, as well as their experience with data analysis tools. These data helped contextualize the usability challenges faced by participants during the task by finding correlations between the characteristics of the participants and their subjective experiences using the web interface.

The results of the pre-study questionnaire revealed that most participants (12 of 18) rated their computer proficiency as high (6 or 7 on a 7-point scale). This can be seen in Fig. 6.5. Given that most of the participants had technical backgrounds, this was expected. However, Fig. 6.6 shows that familiarity with IoT platforms varied significantly, with 6 participants rating themselves as 3 or lower, indicating limited exposure. Additionally, in participants reported diverse experiences with both the frequency (see: Fig. 6.7 and types of data analysis tools used. Python libraries were the most commonly used, indicating a strong presence of technically proficient

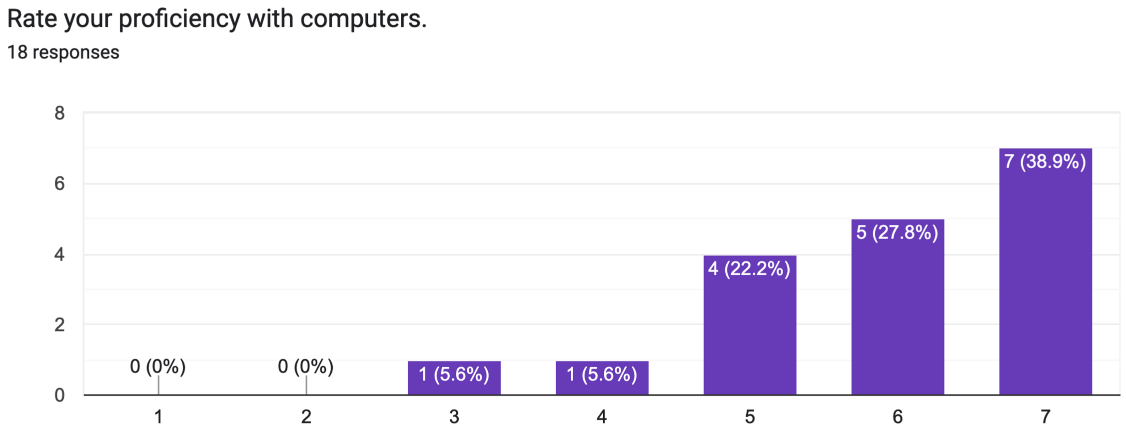


Figure 6.5: Participant self-assessed computer proficiency, rated on a 1–7 scale. (1 = none, 7 = high)

participants familiar with scripting and programming-based tools. Some participants listed simpler tools such as Excel, demonstrating the varied technical backgrounds of the participants.

The results of the post-study questionnaire also show mixed opinions. When asked to rate the functionality of the web interface, most of the participants gave favorable ratings, with half of all participants rating it a 6 or 7, as seen in Fig. 6.8. This matches with post-study verbal feedback indicating that functionality was not a concern, but rather the accessibility of that functionality. TTSS is a powerful platform for managing LoRaWAN networks, but its interface remains daunting to new users.

The remaining questions on the post-study questionnaire were all aimed at analyzing the intuitiveness and understandability of the user interface. The participant ratings for these questions were more nuanced, showcasing a wide range of user experiences. When rating the general ease of navigating the web interface, the participant responses leaned somewhat negative, with half of participants giving a rating of 3 or lower, as seen in Fig. 6.9. This was matched by participant feedback that buttons and menus seems to be placed in confusing or otherwise unintuitive places within the web interface. Furthermore, many participants reported that buttons had confusing

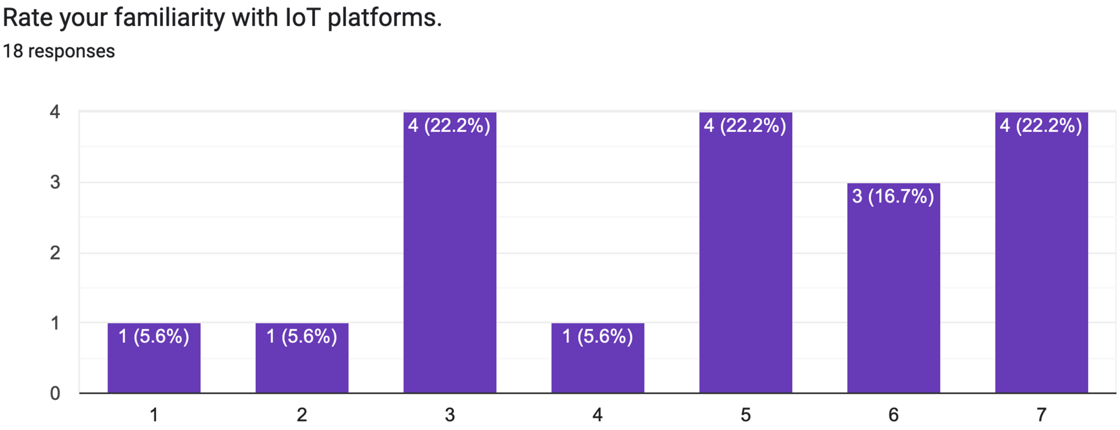


Figure 6.6: Participant self-assessed IoT platform familiarity, rated on a 1–7 scale. (1 = none, 7 = high)

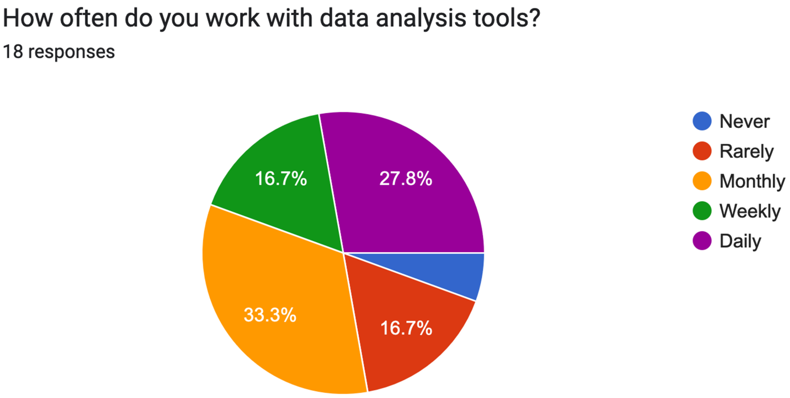


Figure 6.7: Participant self-reported frequency of data analysis tool use.

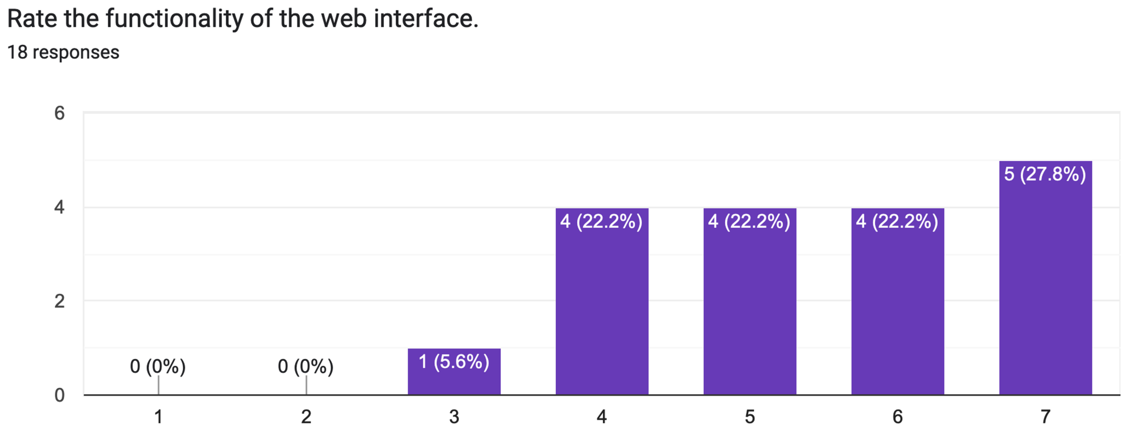


Figure 6.8: Post-study participant ratings of web interface functionality.
(1 = limited functionality, 7 = comprehensive functionality)

names that did not adequately describe their functionality. This led to users having to click buttons that may be incorrect just to figure out what they do. Because this question was asked of the general navigability of the web interface rather than about a specific function, it was not impacted by the order of tasks. While counterbalancing helped offset the order effect for task-specific questions, it could not fully eliminate bias in the responses due to some users having a much easier time with specific tasks depending on the order performed. For this reason, the question of general ease of interface navigability should be regarded higher than the task-specific questions.

The responses were more positive when asked to rate the ease of exporting data, with half of participants rating it a 5 or greater, seen in Fig. 6.10. Unlike the other subjective rating questions on the post-study questionnaire, however, the data export feature is not part of The Things Stack's default functionality. Instead, these data export capabilities were added by research assistants for the CSSI grant. Because this feature was added with the goal of it being used by the general public, it was designed with an emphasis on usability by the general public. This is likely the reason why it was given higher subjective ratings by participants than the other features. That said, there were still a few low ratings that were accompanied by feedback about the

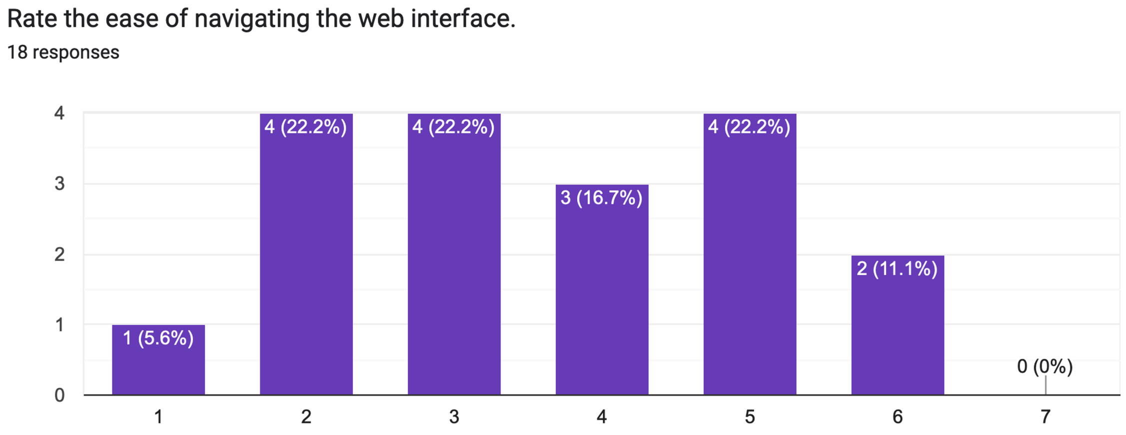


Figure 6.9: Post-study participant ratings of web interface navigation ease.
(1 = very difficult, 7 = very easy)

placement of the relevant data export buttons. It seems that while the data export functionality was considered excellent, the interface for exporting data had room for improvements. One of the most common pieces of feedback was that there should be a data export button on every sensor's page. Currently, the data export button is on the left project sidebar menu. This decision was made because the data export feature includes the option to export the data of multiple sensors at once, but that decision did not take into account the sheer amount of users that would intuitively expect a data export button to exist on the sensor pages themselves. A potential solution to this problem would be to include data export buttons in both locations, with the sensor-specific data export buttons leading to the same page but automatically populating the sensor selection drop-down menu with the respective sensor selected.

The final subjective participant rating requested by the post-study questionnaire was the ease of adding a new sensor. Responses to this question were extremely varied, as seen in Fig. 6.11. There were a handful of obstacles that participants faced during their task to add a new sensor. Most users spent the longest time simply navigating the button to add a new sensor. This can be attributed to a couple reasons. The button's confusing text, which says *Register end device*, may be intuitive to those

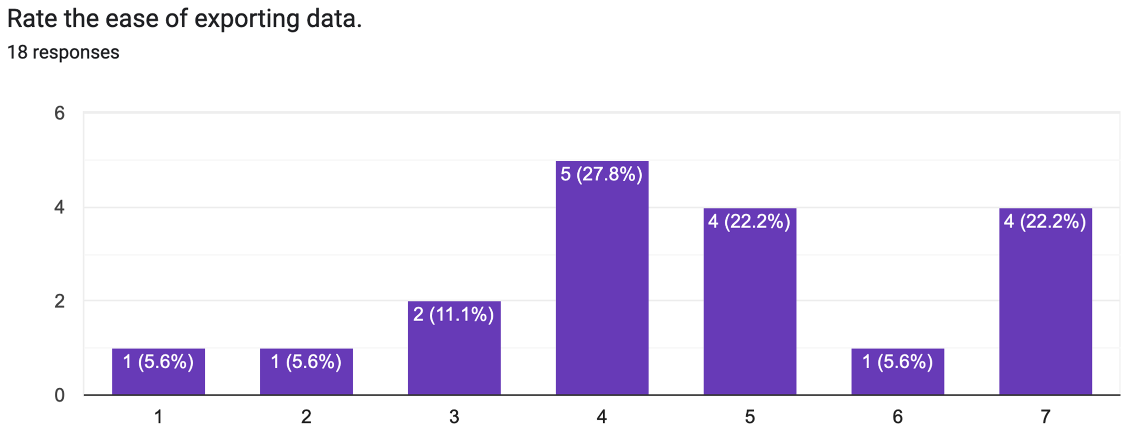


Figure 6.10: Post-study participant ratings of data exporting ease.
(1 = very difficult, 7 = very easy)

with prior LoRaWAN or IoT experience, but the meaning remains ambiguous for those who have not seen the term *end device* used to describe LoRaWAN sensors. Additionally, the placement of the button at the bottom right of the screen goes against existing UI patterns of having important buttons be placed at the top or left side of the screen, leading to further confusion.

An analysis of variance (ANOVA) was performed on all combinations of the aforementioned three pre-study questions with the four post-study questions to explore any possible correlations between participant characteristics and their subjective experiences. The results of this analysis can be seen in Table 6.1. These results indicate that participants' self-rated computer proficiency was the only characteristic to have statistically significant correlations with their subjective ratings of the web interface. This is surprising, because one would expect familiarity with IoT platforms and data analysis tools to also play a role, but none of those relationships reached statistical significance. However, there was a nearly significant correlation between IoT familiarity and ease of navigation, with a p-value of $\sim$0.058.

The statistically significant correlations include positive relationships between computer proficiency and all of the following three subjective rating criteria: ease

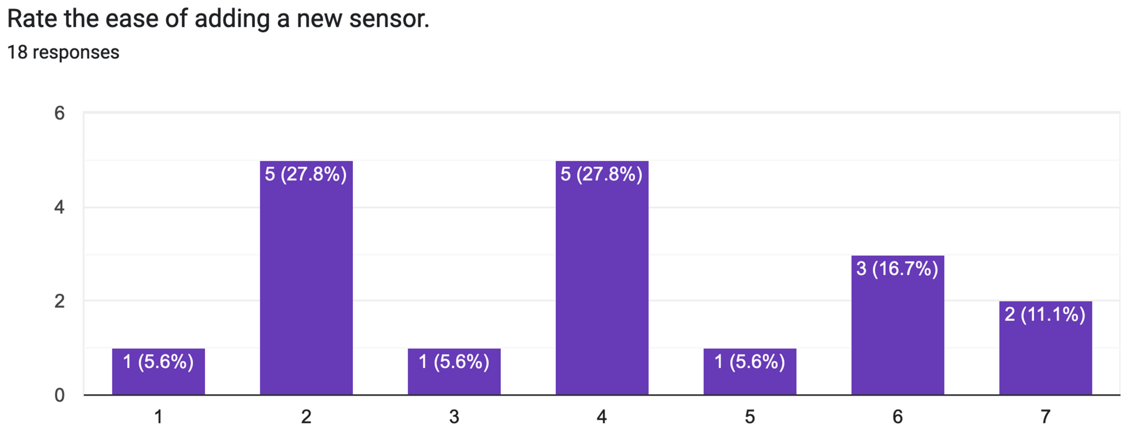


Figure 6.11: Post-study participant ratings of adding a new sensor ease.
(1 = very difficult, 7 = very easy)

of navigation, ease of exporting data, and ease of adding a new sensor. While it may seem obvious that computer proficiency would have a direct correlation with the ease of navigating a user interface, it is important to keep in mind that most popular web interfaces remain very usable by the general public, while the same cannot currently be said for The Things Stack Open Source Edition. Because one of the long-term goals of this project is to open the platform up to the general public, it would be unacceptable for the user interface to remain in a state where only those with substantial computer proficiency can use it.

When analyzing these correlations further, it should be noted that they differed in their significance. Although the correlation between computer proficiency and the ease of exporting data reached $p \leq 0.05$, the two remaining correlations reached a much more substantial threshold of $p \leq 0.001$. This adds a bit more evidence that the data export feature is a bit more usable by the general public than general navigation and adding a new sensor.

Table 6.1: Analysis of variance (ANOVA) between participant characteristics and post-study questionnaire ratings.

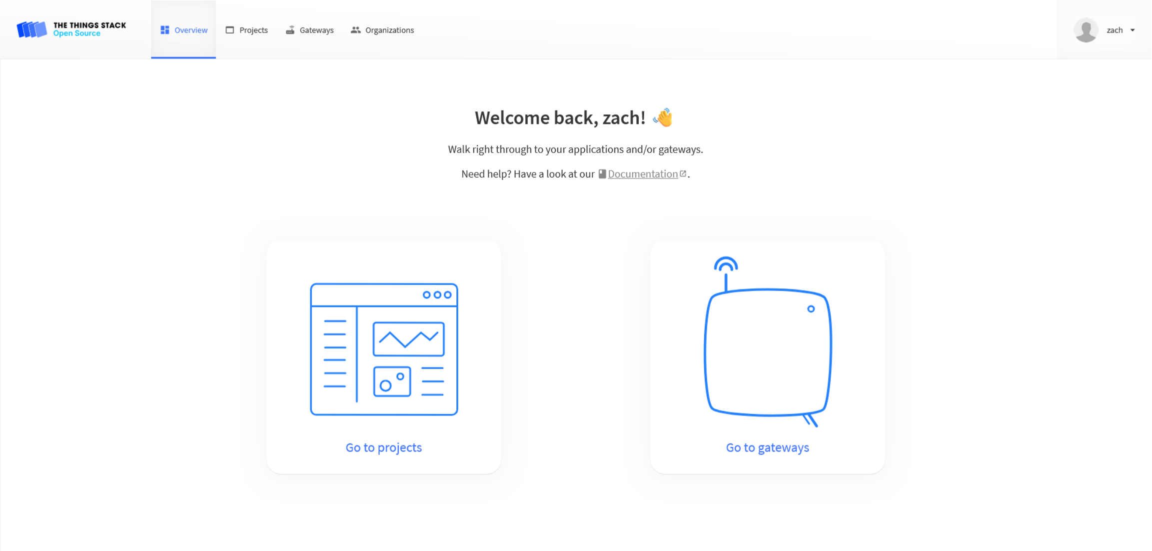
Participant Characteristic	Participant Rating	F-Statistic	P-Value
Computer Proficiency	Rated Functionality	1.157447	0.289571
Computer Proficiency	Ease of Navigation	25.583706	****0.000014**
Computer Proficiency	Ease of Exporting Data	6.569028	***0.014974**
Computer Proficiency	Ease of Adding a New Sensor	13.655738	****0.000768**
IoT Familiarity	Rated Functionality	1.812618	0.187102
IoT Familiarity	Ease of Navigation	3.859251	0.057688
IoT Familiarity	Ease of Exporting Data	0.033764	0.855302
IoT Familiarity	Ease of Adding a New Sensor	1.535484	0.223777
Data Analysis Frequency	Rated Functionality	0.323883	0.856980
Data Analysis Frequency	Ease of Navigation	0.448794	0.771584
Data Analysis Frequency	Ease of Exporting Data	0.495102	0.739737
Data Analysis Frequency	Ease of Adding a New Sensor	1.392253	0.290563

Note: $^*p \leq 0.05$, $^{**}p \leq 0.001$

6.3.3 Participant Comments and Recommendations

The end of the post-study questionnaire included a free response question for feedback and suggestions on further improving the user interface. One of the more common suggestions included the suggestion to remove unnecessary elements such as the LoRa region and frequency sub-band selection menus found when adding a new sensor. This makes sense given that this project will only be used in the US 915 MHz region on frequency sub-band 1. Participants also suggested adding more contrast to the web interface because its current state has a lot of white space with some blue UI elements sprinkled in. This makes the web interface hard on the eyes when viewed on a bright screen at night. Participants also suggested relocating UI elements to more intuitive positions, as well as renaming buttons that had unnecessarily technical names, such as *Register end device.*

One of the non-task related suggestions was to re-haul the overview page. The default overview page in TTSS acts as the home page and is the first page presented to users after logging in. It only contains two buttons - *Projects* and *Gateways*, which can be seen in Fig. 6.12. These buttons are redundant given that they also exist in the top-bar menu, which is always displayed on the overview page. One user suggested



Welcome back, zach! 👋

Walk right through to your applications and/or gateways.

Need help? Have a look at our Documentation.

Figure 6.12: The Overview page of The Things Stack, which acts as its homepage.

that the overview page be changed to a customizable dashboard page, where users can configure default data visualizations and live data outputs as they see fit. The idea of having preconfigured data visualizations was repeated in a few different ways, including suggestions to set a default visualization metric for each sensor so that each sensor's page would include a simple "at-a-glance" graph.

6.4 Discussion

The usability evaluation of the CSSI project's web interface, built upon The Things Stack Open Source Edition, provided insightful findings that highlight both the strengths and areas for improvement within the current design. The study highlights the importance of tailoring the interface to accommodate a diverse user base, including climate researchers and prospective citizen scientists who may possess varying levels of technical expertise. One of the key observations from the study is the placement and labeling of essential functionalities such as the *Export Data* and *Data Visualization* buttons. Participants encountered challenges in locating these features within the project sidebar menu, suggesting that their current positioning may not align with common user interface conventions. This indicates an opportunity to reorganize the interface to improve accessibility, potentially by placing frequently used actions in more prominent locations that are easily discoverable by users.

Additionally, the study identified that certain interface elements, such the sensor field selection drop-down menu, were conditionally hidden until specific criteria were met. While this design choice serves a functional purpose, it inadvertently led to confusion among users who were unaware of how to access these options and did not notice their sudden appearance once the conditions were met. Enhancing the visibility of such elements while still making it clear that they are inaccessible until requisite criteria are met could improve the overall intuitiveness of the interface, making it easier for users to navigate and utilize the platform effectively.

The terminology used within the interface also emerged as a significant factor influencing user experience. Terms inherited from TTSS, such as *Register end device*, were found to be less intuitive for users unfamiliar with IoT and LoRaWAN systems. Simplifying and clarifying button labels to more descriptive terms like *Add New Sensor* can make the interface more user-friendly and reduce the cognitive load on users who are new to these technologies. This adjustment aligns with best practices in user-centered design, which advocate for clear and accessible language to accommodate a broader audience.

Despite some navigational challenges, participants generally provided positive ratings for the functionality of the web interface. This indicates that the platform's core features are robust and meet the needs of technically proficient users. However, to fully realize its potential for a wider audience, it is important to improve the accessibility and discoverability of these features. The positive functionality ratings provide a strong foundation upon which further usability improvements can be built, ensuring that the platform remains both powerful and approachable.

The correlation between participants' computer proficiency and their ease-of-use ratings highlights the importance of designing an interface that accommodates varying levels of technical expertise. Although higher computer proficiency was associated with better usability ratings, it is crucial to ensure that the platform remains accessible to users with a less technical background. Implementing user-centered design principles, such as simplifying workflows and providing intuitive navigation paths, can help achieve a more inclusive interface that serves the entire target audience effectively.

Participant feedback offered several constructive recommendations that can guide future enhancements. Relocating critical UI elements to more intuitive locations, renaming ambiguous buttons to clearly reflect their functions, and enhancing visual feedback mechanisms are among the top suggestions. Additionally, redesigning the overview page into a customizable dashboard would allow users to tailor the interface to their specific needs, thereby improving usability and engagement. Enhancing the visual design by increasing contrast and reducing excessive white space was also recommended to improve readability and reduce eye strain under varying lighting conditions.

In summary, this usability study provided a comprehensive understanding of the current strengths and opportunities for improvement within the CSSI web interface. By addressing the identified challenges through thoughtful redesign and user-centered enhancements, the CSSI project can significantly enhance the accessibility and effectiveness of the platform. These improvements will support the immediate needs of climate researchers and pave the way for meaningful public participation in climate science initiatives, ultimately contributing to the CSSI project's vision of inclusive and participatory research.

Chapter 7

Conclusion and Future Work

This thesis explores the practical implementation of a LoRaWAN-based environmental monitoring platform, showcasing its potential for academic research, citizen science, and industrial applications. The CSSI project illustrates how combining energy-efficient IoT devices with modern data management tools can create a reliable platform for real-time environmental monitoring and analysis. Using components such as The Things Stack [34], Apache NiFi [15], and TimescaleDB [25], the system effectively processes and stores large amounts of time-sensitive data, while remaining adaptable to diverse applications and requirements.

This work highlights several key achievements. First, it showcases the feasibility of deploying LoRaWAN in diverse environments, from rural areas like the Lake Tahoe Basin to industrial settings, providing reliable data transmission over long distances. The integration of open-source technologies such as The Things Stack and Apache NiFi allows for the creation of a modular, extensible open-source system capable of adapting to different use cases. Additionally, the decision to use TimescaleDB for time-series data management allows for efficient data storage and querying, supporting real-time analytics and decision-making processes through a web interface.

The technologies included in the CSSI project have been individually tested in real-world scenarios, including environmental monitoring, agricultural research, and industrial applications. These tests have validated the system's ability to operate under various conditions, demonstrating its robustness and reliability. The use of LoRaWAN technology allows the monitoring of key environmental variables, such as temperature, humidity, and soil moisture, providing critical data for environmental research.

Despite these successes, several challenges have emerged throughout the course of the project. One of the most significant challenges faced was the unstandardized payload formatting for LoRaWAN end devices among different manufacturers, or even sometimes among different devices with the same manufacturer. This limitation has started to lift, however, with the ever-increasing device repository within The Things Stack allowing end devices to be added without manual input of a payload formatter by the user.

Looking to the future, there are several avenues for improving and expanding the system. An important next step involves integrating machine learning techniques to improve data analysis capabilities. These algorithms could help uncover subtle relationships and trends within sensor data, enabling applications like forecasting changes in local climate, optimizing infrastructure maintenance schedules, and improving supply chain operations.

Another area for future work is the development of more advanced user interfaces. While the current system includes a web portal for data visualization and export, there is potential to create more customizable dashboards and tools tailored to specific user needs. For instance, adding features that allow users to set custom alerts based on sensor data would increase the system's usability, particularly in industrial and commercial settings where real-time monitoring and response are critical. Existing open-source technologies, such as MQTT or gRPC and REST APIs, can be utilized for such a notification system.

To support wider adoption and large-scale deployments, the system will need to accommodate thousands of sensors spread over vast areas. Achieving this level of scalability requires infrastructure improvements to ensure reliability under heavy loads. Measures such as distributing traffic across multiple servers, incorporating system redundancies, and implementing failover mechanisms will help maintain continuous operation even during unexpected spikes in usage or temporary hardware failures.

Furthermore, the system could benefit from greater integration with other IoT platforms and services. By connecting the system to cloud-based platforms such as AWS IoT or Google Cloud IoT, it would be possible to leverage additional tools for data processing, storage, and analysis. This would not only increase the system's flexibility but also provide users with access to more advanced features with industry-polished user interfaces, such as real-time machine learning models and predictive analytics.

In addition to technical improvements, there are opportunities to expand the system's use cases beyond environmental monitoring and agriculture. The system's architecture is flexible enough to support a wide range of applications, from smart cities to industrial automation. For example, the system could be adapted to monitor energy consumption in buildings, track the condition of infrastructure such as bridges and roads, or even support the development of autonomous vehicles by providing real-time environmental data.

Finally, the CSSI project has the potential to contribute to the democratization of science by enabling citizen scientists to participate in data collection and analysis. Simplifying the platform for nonexperts will help attract a broader audience to environmental monitoring initiatives. By lowering technical barriers, hobbyists, local communities, and schools can contribute meaningful data to scientific studies, fostering greater public participation in tracking and addressing climate challenges. This collaborative approach can generate richer datasets for researchers while raising community awareness of environmental trends and issues.

In conclusion, this thesis establishes a practical foundation for a LoRaWAN-based IoT platform capable of supporting a variety of applications. While challenges such as improving usability and ensuring system reliability remain, the open and modular design of the platform allows for ongoing enhancements and customization. By continuing to refine its features, such as advanced data analytics and user-friendly interfaces, the system has the potential to benefit fields ranging from climate science to industrial operations. Future work will explore expanding its scale, integrating emerging technologies, and opening new opportunities for collaboration among researchers, businesses, and citizen scientists.

References

[1] Diki Arisandi, Amir Syamsuadi, and Liza Trisnawati. Enhancing peatland fire prevention: an incremental LoRa and mobile-based early warning system. English. *International Journal of Advanced Technology and Engineering Exploration*, 10(108):1368–1391, November 2023. ISSN: 23945443. DOI: `10.19101/IJATEE.2023.10101900`. URL: `https://www.proquest.com/docview/2916907070/abstract/584CEFCD9B8247AFPQ/1` (visited on 11/23/2024). Num Pages: 1368-1391 Place: Bhopal, India Publisher: Accent Social and Welfare Society.

[2] Maarten Baert. Simplescreenrecorder, November 2024. URL: `https://github.com/MaartenBaert/ssr` (visited on 11/26/2024). original-date: 2012-12-18T19:16:16Z.

[3] Konstantina Banti, Ioanna Karampelia, Thomas Dimakis, Alexandros-Apostolos A. Boulogeorgos, Thomas Kyriakidis, and Malamati Louta. LoRaWAN Communication Protocols: A Comprehensive Survey under an Energy Efficiency Perspective. en. *Telecom*, 3(2):322–357, June 2022. ISSN: 2673-4001. DOI: `10.3390/telecom3020018`. URL: `https://www.mdpi.com/2673-4001/3/2/18` (visited on 12/05/2024). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[4] Harvey Bates, Matthew Pierce, and Allen Benter. Real-Time Environmental Monitoring for Aquaculture Using a LoRaWAN-Based IoT Sensor Network. en. *Sensors*, 21(23):7963, November 2021. ISSN: 1424-8220. DOI: `10.3390/s21237963`. URL: `https://www.mdpi.com/1424-8220/21/23/7963` (visited on 11/23/2024).

[5] Mevludin Blazevic and Dennis M. Riehle. Comparing On-Premise IoT Platforms: Empowering University of Things Ecosystems with Effective Device Management. en. In volume 2, pages 310–320. SCITEPRESS, April 2024. ISBN: 978-989-758-699-6. DOI: `10.5220/0012727000003705`. URL: `http://www.scitepress.org/Papers/2024/127270` (visited on 11/23/2024).

[6] Pongsarun Boonyopakorn and Thanamate Thongna. Environment Monitoring System through LoRaWAN for Smart Agriculture. In *2020 - 5th International Conference on Information Technology (InCIT)*, pages 12–16, Chonburi, Thailand. IEEE, October 2020. ISBN: 978-1-72816-694-0. DOI: `10.1109/`

InCIT50588.2020.9310977. URL: https://ieeexplore.ieee.org/document/9310977/ (visited on 11/23/2024).

[7] Chase Carthen, Zach Estreito, Vinh Le, Jehren Boehm, Scotty Strachan, Alireza Tavakkoli, Frederick C. Harris, and Sergiu M. Dascalu. Initial Design and Implementation of an Edge-to-Edge LoRaWAN Data Collection System. en. In Shahram Latifi, editor, *ITNG 2024: 21st International Conference on Information Technology-New Generations*, pages 241–247, Cham. Springer Nature Switzerland, 2024. ISBN: 978-3-031-56599-1. DOI: 10.1007/978-3-031-56599-1_32.

[8] Chase Carthen, Araam Zaremehrjardi, Vinh Le, Carlos Cardillo, Scotty Strachan, Alireza Tavakkoli, Frederick C. Harris, and Sergiu M. Dascalu. Orchestrating Apache NiFi/MiNiFi within a Spatial Data Pipeline. In *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 366–371, Orlando, FL, USA. IEEE, May 2023. ISBN: 9798350345889. DOI: 10.1109/SERA57763.2023.10197731. URL: https://ieeexplore.ieee.org/document/10197731/ (visited on 11/23/2024).

[9] Roger Pueyo Centelles, Felix Freitag, Roc Meseguer, and Leandro Navarro. Beyond the Star of Stars: An Introduction to Multihop and Mesh for LoRa and LoRaWAN. *IEEE Pervasive Computing*, 20(2):63–72, April 2021. ISSN: 1558-2590. DOI: 10.1109/MPRV.2021.3063443. URL: https://ieeexplore.ieee.org/abstract/document/9385408?casa_token=YDNtoUfqqHgAAAAA:Gi9lxqxON8OVm_2Vde2qphgiW3TlMdWsfyz5brpt_Rj107AzHy0ZCClIio_d8pXvPRjjQmAToA (visited on 11/24/2024). Conference Name: IEEE Pervasive Computing.

[10] Chiu Chun Chan, Sheeraz A. Alvi, Xiangyun Zhou, Salman Durrani, Nicholas Wilson, and Marta Yebra. A Survey on IoT Ground Sensing Systems for Early Wildfire Detection: Technologies, Challenges, and Opportunities. *IEEE Access*, 12:172785–172819, 2024. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3501336. URL: https://ieeexplore.ieee.org/abstract/document/10756651 (visited on 11/23/2024). Conference Name: IEEE Access.

[11] Somphop Chanthakit, Phongsak Keeratiwintakorn, and Choopan Rattanapoka. An IoT System Design with Real-Time Stream Processing and Data Flow Integration. In *2019 Research, Invention, and Innovation Congress (RI2C)*, pages 1–5, December 2019. DOI: 10.1109/RI2C48728.2019.8999968. URL: https://ieeexplore.ieee.org/document/8999968 (visited on 11/23/2024).

[12] Bruno Citoni, Francesco Fioranelli, Muhammad A. Imran, and Qammer H. Abbasi. Internet of Things and LoRaWAN-Enabled Future Smart Farming. *IEEE Internet of Things Magazine*, 2(4):14–19, December 2019. ISSN: 2576-3180, 2576-3199. DOI: 10.1109/IOTM.0001.1900043. URL: https://ieeexplore.ieee.org/document/8982742/ (visited on 11/23/2024).

[13] Semtech Corporation. Semtech acquires wireless long range ip provider cycleo. URL: https://www.sec.gov/Archives/edgar/data/88941/000115752312001273/a50196224ex99_2.htm (visited on 11/26/2024).

[14] Jonathan de Carvalho Silva, Joel J. P. C. Rodrigues, Antonio M. Alberti, Petar Solic, and Andre L. L. Aquino. LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities. In *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–6, July 2017. URL: `https://ieeexplore.ieee.org/abstract/document/8019271` (visited on 11/24/2024).

[15] Apache Software Foundation. Apache nifi, November 2024. URL: `https://github.com/apache/nifi` (visited on 11/26/2024). original-date: 2014-12-12T08:00:05Z.

[16] National Science Foundation. Elements: Innovating for Edge-to-Edge Climate Services, August 2022. URL: `https://www.nsf.gov/awardsearch/showAward?AWD_ID=2209806` (visited on 11/27/2024).

[17] Paula Fraga-Lamas, Mikel Celaya-Echarri, Leyre Azpilicueta, Peio Lopez-Iturri, Francisco Falcone, and Tiago M. Fernández-Caramés. Design and Empirical Validation of a LoRaWAN IoT Smart Irrigation System. en. In *The 6th International Electronic Conference on Sensors and Applications*, page 62. MDPI, April 2020. DOI: `10.3390/ecsa-6-06540`. URL: `https://www.mdpi.com/2504-3900/42/1/62` (visited on 11/23/2024).

[18] Piotr Grzesik and Dariusz Mrozek. Comparative Analysis of Time Series Databases in the Context of Edge Computing for Low Power Sensor Networks. en. In Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science – ICCS 2020*, pages 371–383, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-50426-7. DOI: `10.1007/978-3-030-50426-7_28`.

[19] Yuanzhe Hao, Xiongpai Qin, Yueguo Chen, Yaru Li, Xiaoguang Sun, Yu Tao, Xiao Zhang, and Xiaoyong Du. TS-Benchmark: A Benchmark for Time Series Databases. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 588–599, Chania, Greece. IEEE, April 2021. ISBN: 978-1-72819-184-3. DOI: `10.1109/ICDE51399.2021.00057`. URL: `https://ieeexplore.ieee.org/document/9458659/` (visited on 11/23/2024).

[20] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A Survey of LoRaWAN for IoT: From Technology to Application. en. *Sensors*, 18(11):3995, November 2018. ISSN: 1424-8220. DOI: `10.3390/s18113995`. URL: `https://www.mdpi.com/1424-8220/18/11/3995` (visited on 11/24/2024). Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[21] Georgi Hristov, Jordan Raychev, Diyana Kinaneva, and Plamen Zahariev. Emerging Methods for Early Detection of Forest Fires Using Unmanned Aerial Vehicles and Lorawan Sensor Networks. In *2018 28th EAEEIE Annual Conference (EAEEIE)*, pages 1–9, September 2018. DOI: `10.1109/EAEEIE.2018.8534245`. URL: `https://ieeexplore.ieee.org/abstract/document/8534245?casa_token=toC-Nog4QScAAAAA:uz-cj01aJIs5BlI3KuqLWVHx0c0jWY4ylJb_ayoPy`

tkXmcHnrhAaSF_Sfp_JcOSdOE9D2TLf7w (visited on 11/23/2024). ISSN: 2472-7687.

[22] Ahmad Rizan Ibrahim, Nik Hisham Nik Ibrahim, Ahmad Nizar Harun, Mohamed Rawidean Mohd Kassim, Shamsul Effendy Kamaruddin, Liang Kim Meng, Chong Jin Hui, and Gunawan Witjaksono. Automated Monitoring and LoRaWAN Control Mechanism for Swiftlet Bird House. In *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pages 1–5, August 2018. DOI: 10.1109/ICIAS.2018.8540562. URL: https://ieeexplore.ieee.org/abstract/document/8540562 (visited on 11/23/2024).

[23] Waheb A. Jabbar, Thanasrii Subramaniam, Andre Emelio Ong, Mohd Iqmal Shu'Ib, Wenyan Wu, and Mario A. De Oliveira. LoRaWAN-Based IoT System Implementation for Long-Range Outdoor Air Quality Monitoring. en. *Internet of Things*, 19:100540, August 2022. ISSN: 25426605. DOI: 10.1016/j.iot.2022.100540. URL: https://linkinghub.elsevier.com/retrieve/pii/S2542660522000427 (visited on 11/23/2024).

[24] Ala' Khalifeh, Abdullah Al-Qammaz, Khalid A. Darabkh, Laith Abualigah, Ahmad M. Khasawneh, and Zinon Zinonos. An AI Based Irrigation and Weather Forecasting System utilizing LoRaWAN and Cloud Computing Technologies. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 443–448, St. Petersburg, Moscow, Russia. IEEE, January 2021. ISBN: 978-1-66540-476-1. DOI: 10.1109/ElConRus51938.2021.9396431. URL: https://ieeexplore.ieee.org/document/9396431/ (visited on 11/23/2024).

[25] Ajay Kulkarni and Michael Freedman. Timescaledb: an open-source time-series sql database optimized for fast ingest and complex queries. packaged as a postgresql extension. November 2024. URL: https://github.com/timescale/timescaledb (visited on 11/26/2024). original-date: 2017-03-07T20:03:41Z.

[26] Roger Light. Eclipse mosquitto - an open source mqtt broker, November 2024. URL: https://github.com/eclipse-mosquitto/mosquitto (visited on 11/26/2024). original-date: 2016-03-10T20:19:09Z.

[27] Rui Liu, Jun Yuan, and Xiangdong Huang. Benchmarking Time Series Databases with IoTDB-Benchmark for IoT Scenarios, 2019. DOI: 10.48550/ARXIV.1901.08304. URL: https://arxiv.org/abs/1901.08304 (visited on 11/23/2024).

[28] Canonical Ltd. Ubuntu 24.04.1 (Noble Numbat). URL: https://releases.ubuntu.com/noble/ (visited on 11/26/2024).

[29] Marco Mancini, Paola Nassisi, Antonio Trabucco, Alessandro Meloni, Konstantina Toli, Valentina Bacciu, Giovanni Aloisio, Donatella Spano, and Simone Mereu. An Open Source and Low-Cost Internet of Things-enabled Service for Irrigation Management. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1714–1719, October 2019. DOI: 10.1109/SMC.2019.8914230. URL: https://ieeexplore.ieee.org/

abstract/document/8914230?casa_token=KtSx1cQdgwIAAAAA:6kZH89lPmxMw7aKbdIYb7pNkhv6kJpS9U4VLqIgdUaa3y8VISZ6XAPWvlQXcR3jLWr16-9_FEA (visited on 11/23/2024). ISSN: 2577-1655.

[30] Luz E. Marquez, Alfonso Osorio, Maria Calle, Juan C. Velez, Antonio Serrano, and John E. Candelo-Becerra. On the Use of LoRaWAN in Smart Cities: A Study With Blocking Interference. *IEEE Internet of Things Journal*, 7(4):2806–2815, April 2020. ISSN: 2327-4662. DOI: `10.1109/JIOT.2019.2962976`. URL: `https://ieeexplore.ieee.org/abstract/document/8945192?casa_token=4saO5s1rkE4AAAAA:eoThb2I_yFFYeHp_yLQcnNQd2BqFzHvNXGCfGLowsky9POEg4nsE-fYsf-HWxEK62DUB-II62g` (visited on 11/24/2024). Conference Name: IEEE Internet of Things Journal.

[31] Biswajeeban Mishra and Attila Kertesz. The Use of MQTT in M2M and IoT Systems: A Survey. *IEEE Access*, 8:201071–201086, 2020. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2020.3035849`. URL: `https://ieeexplore.ieee.org/document/9247996/` (visited on 11/23/2024).

[32] Jalal Mostafa, Sara Wehbi, Suren Chilingaryan, and Andreas Kopmann. SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things. In *Proceedings of the 34th International Conference on Scientific and Statistical Database Management*, SSDBM '22, pages 1–11, New York, NY, USA. Association for Computing Machinery, August 2022. ISBN: 978-1-4503-9667-7. DOI: `10.1145/3538712.3538723`. URL: `https://dl.acm.org/doi/10.1145/3538712.3538723` (visited on 11/22/2024).

[33] The Things Network. Device Classes. en-us. URL: `https://www.thethingsnetwork.org/docs/lorawan/classes/` (visited on 11/26/2024).

[34] The Things Network. The things stack, an open source lorawan network server, November 2024. URL: `https://github.com/TheThingsNetwork/lorawan-stack` (visited on 11/26/2024). original-date: 2019-01-12T15:52:34Z.

[35] Nik Hisham Nik Ibrahim, Ahmad Rizan Ibrahim, Ibrahim Mat, Ahmad Nizar Harun, and Gunawan Witjaksono. LoRaWAN in Climate Monitoring in Advance Precision Agriculture System. In *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pages 1–6, Kuala Lumpur. IEEE, August 2018. ISBN: 978-1-5386-7269-3. DOI: `10.1109/ICIAS.2018.8540598`. URL: `https://ieeexplore.ieee.org/document/8540598/` (visited on 11/23/2024).

[36] Tommaso Polonelli, Davide Brunelli, Andrea Bartolini, and Luca Benini. A LoRaWAN Wireless Sensor Network for Data Center Temperature Monitoring. en. In Sergio Saponara and Alessandro De Gloria, editors, *Applications in Electronics Pervading Industry, Environment and Society*. Volume 573, pages 169–177. Springer International Publishing, Cham, 2019. ISBN: 978-3-030-11972-0 978-3-030-11973-7. DOI: `10.1007/978-3-030-11973-7_20`. URL: `http://link.springer.com/10.1007/978-3-030-11973-7_20` (visited on 11/23/2024).

[37] Shivananda Poojara, Argo Jõeleht, Pelle Jakovits, and Satish Narayana Srirama. Serverless Outlier Management for Environmental IoT Data - A Case Study of PuhatuMonitoring. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–7, October 2022. DOI: 10.1109/WF-IoT54382.2022.10152102. URL: https://ieeexplore.ieee.org/abstract/document/10152102?casa_token=7A1n5gxx9hsAAAAA:fLH1XaHfW19EH50iluvrkG986kocKCj3vYSXigHrEM04HHbkDb9qQw0IyCUOz2-ux1rW2-rFfg (visited on 11/23/2024).

[38] Montaser N. A. Ramadan, Tasnim Basmaji, Abdalla Gad, Hasan Hamdan, Bekir Tevfik Akgün, Mohammed A. H. Ali, Mohammad Alkhedher, and Mohammed Ghazal. Towards early forest fire detection and prevention using AI-powered drones and the IoT. *Internet of Things*, 27:101248, October 2024. ISSN: 2542-6605. DOI: 10.1016/j.iot.2024.101248. URL: https://www.sciencedirect.com/science/article/pii/S2542660524001896 (visited on 11/23/2024).

[39] S. R. Jino Ramson, Walter D. Leon-Salas, Zachary Brecheisen, Erika J. Foster, Cliff T. Johnston, Darrell G. Schulze, Timothy Filley, Rahim Rahimi, Martin Juan Carlos Villalta Soto, Juan A. Lopa Bolivar, and Mauricio Postigo Malaga. A Self-Powered, Real-Time, LoRaWAN IoT-Based Soil Health Monitoring System. *IEEE Internet of Things Journal*, 8(11):9278–9293, June 2021. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2021.3056586. URL: https://ieeexplore.ieee.org/document/9344804/ (visited on 11/23/2024).

[40] Brecht Reynders and Sofie Pollin. Chirp spread spectrum as a modulation technique for long range communication. In *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, pages 1–5, November 2016. DOI: 10.1109/SCVT.2016.7797659. URL: https://ieeexplore.ieee.org/document/7797659 (visited on 11/24/2024).

[41] Olivier Bernard Andre Seller. Wireless communication method. en. (US20160094269A1). March 2016. URL: https://patents.google.com/patent/US20160094269A1/en (visited on 11/26/2024).

[42] Sandra Sendra, Laura García, Jaime Lloret, Ignacio Bosch, and Roberto Vega-Rodríguez. LoRaWAN Network for Fire Monitoring in Rural Environments. en. *Electronics*, 9(3):531, March 2020. ISSN: 2079-9292. DOI: 10.3390/electronics9030531. URL: https://www.mdpi.com/2079-9292/9/3/531 (visited on 11/23/2024). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[43] Ritesh Singh, Michiel Aernouts, Mats De Meyer, Maarten Weyn, and Rafael Berkvens. Leveraging LoRaWAN Technology for Precision Agriculture in Greenhouses. en. *Sensors*, 20(7):1827, March 2020. ISSN: 1424-8220. DOI: 10.3390/s20071827. URL: https://www.mdpi.com/1424-8220/20/7/1827 (visited on 11/23/2024).

[44] Ritesh Kumar Singh, Mohammad Hasan Rahmani, Maarten Weyn, and Rafael Berkvens. Joint Communication and Sensing: A Proof of Concept and Datasets for Greenhouse Monitoring Using LoRaWAN. en. *Sensors*, 22(4):1326, February 2022. ISSN: 1424-8220. DOI: `10.3390/s22041326`. URL: `https://www.mdpi.com/1424-8220/22/4/1326` (visited on 11/23/2024).

[45] Níbia Souza Bezerra, Christer Åhlund, Saguna Saguna, and Vicente De Sousa. Temperature Impact in LoRaWAN—A Case Study in Northern Sweden. en. *Sensors*, 19(20):4414, October 2019. ISSN: 1424-8220. DOI: `10.3390/s19204414`. URL: `https://www.mdpi.com/1424-8220/19/20/4414` (visited on 11/23/2024).

[46] Michael Stonebraker. Postgresql, November 2024. URL: `https://github.com/postgres/postgres` (visited on 11/26/2024). original-date: 2010-09-21T11:35:45Z.

[47] Antonio Valente, Carlos Costa, Leonor Pereira, Bruno Soares, José Lima, and Salviano Soares. A LoRaWAN IoT System for Smart Agriculture for Vine Water Status Determination. en. *Agriculture*, 12(10):1695, October 2022. ISSN: 2077-0472. DOI: `10.3390/agriculture12101695`. URL: `https://www.mdpi.com/2077-0472/12/10/1695` (visited on 11/23/2024).

[48] Benny Vejlgaard, Mads Lauridsen, Huan Nguyen, Istvan Z. Kovacs, Preben Mogensen, and Mads Sorensen. Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, March 2017. DOI: `10.1109/WCNC.2017.7925510`. URL: `https://ieeexplore.ieee.org/abstract/document/7925510?casa_token=L5w2yVLNM5MAAAAA:aH8S17IEGEz4vMwPvXYkZ95bBT-p3TZQMiyxyezneDIsU_Zvy9Z1vZLszYaI0tvgctyo0KL62Q` (visited on 11/24/2024). ISSN: 1558-2612.

[49] Gonçalo Vítor, Pedro Rito, Susana Sargento, and Filipe Pinto. A scalable approach for smart city data platform: Support of real-time processing and data sharing. *Computer Networks*, 213:109027, August 2022. ISSN: 1389-1286. DOI: `10.1016/j.comnet.2022.109027`. URL: `https://www.sciencedirect.com/science/article/pii/S1389128622001839` (visited on 11/23/2024).

[50] Yansong Wang, Yi Huang, and Chaoyun Song. A New Smart Sensing System Using LoRaWAN for Environmental Monitoring. In *2019 Computing, Communications and IoT Applications (ComComAp)*, pages 347–351, Shenzhen, China. IEEE, October 2019. ISBN: 978-1-72811-973-1. DOI: `10.1109/ComComAp46287.2019.9018829`. URL: `https://ieeexplore.ieee.org/document/9018829/` (visited on 11/23/2024).

[51] Karol Wnęk and Piotr Boryło. A Data Processing and Distribution System Based on Apache NiFi. en. *Photonics*, 10(2):210, February 2023. ISSN: 2304-6732. DOI: `10.3390/photonics10020210`. URL: `https://www.mdpi.com/2304-6732/10/2/210` (visited on 11/23/2024).