electronics

MDPI

Article

Comprehensive Security for IoT Devices with Kubernetes and Raspberry Pi Cluster

Ionut-Catalin Donca [1], Ovidiu Petru Stan [1,*], Marius Misaros [1], Anca Stan [2] and Liviu Miclea [1]

[1] Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania; ionut.donca@aut.utcluj.ro (I.-C.D.); marius.misaros@aut.utcluj.ro (M.M.); liviu.miclea@aut.utcluj.ro (L.M.)

[2] Faculty of Industrial Engineering, Robotics and Production Management, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania; anca.stan@muri.utcluj.ro

* Correspondence: ovidiu.stan@aut.utcluj.ro

Abstract: Environmental monitoring systems have gained prominence across diverse applications, necessitating the integration of cutting-edge technologies. This article comprehensively explores such a system, emphasizing the integration of a Raspberry Pi cluster with the BME680 environmental sensor within a Kubernetes framework. This study encompasses the technical aspects of hardware configuration and places a significant focus on security benchmarks and robustness validation. The environmental monitoring infrastructure discussed in this article delves into the intricacies of the Raspberry Pi cluster's hardware setup, including considerations for scalability and redundancy. This research addresses critical security gaps in contemporary environmental monitoring systems, particularly vulnerabilities linked to IoT deployments. Amidst increasing threats, this study introduces a robust framework that integrates advanced security tools—HashiCorp (San Francisco, CA, USA) Vault v1.16 for dynamic secret management and OpenID Connect for authentication processes—to enhance applications and system integrity and resilience within the Kubernetes environment. The approach involves a multi-layered security architecture that fortifies the storage and management of credentials and ensures authenticated and authorized interactions within IoT networks. Furthermore, our research incorporates a series of security benchmark tests, including vulnerability scanning, penetration testing, and access control assessments. Additionally, this article addresses crucial aspects related to data management and analysis, detailing the methodologies employed for storing, processing, and deriving insights from the collected environmental data. It further explores the integration of the monitoring system with existing infrastructure and systems, facilitating seamless data sharing and interoperability and offering valuable insights into the system's ability to withstand potential threats and vulnerabilities. The integration of Raspberry Pi clusters with BME680 environmental sensors within a Kubernetes-managed framework significantly enhances the scalability and security of IoT systems. This study quantifies the improvements, demonstrating at least a 30% enhancement in system responsiveness and a minimum 40% reduction in vulnerability exposures, as verified by extensive security benchmarks, including penetration testing. These advancements facilitate robust, scalable IoT deployments, with potential applications extending beyond environmental monitoring to include industrial and urban settings. The incorporation of dynamic secret management with HashiCorp Vault and secure authentication with OpenID Connect provides a blueprint for developing resilient IoT architectures capable of supporting high-security and high-availability applications. In conclusion, this article contributes to the expanding body of knowledge in IoT and environmental monitoring and establishes a strong foundation for future work. These outcomes suggest promising directions for further research in secure IoT applications and present practical implications for the deployment of secure and scalable IoT solutions in critical infrastructures.

Keywords: security; microservices; IoT; secrets management; OIDC; cybersecurity; resilience; penetration testing; vulnerability scanning

Citation: Donca, I.-C.; Stan, O.P.; Misaros, M.; Stan, A.; Miclea, L. Comprehensive Security for IoT Devices with Kubernetes and Raspberry Pi Cluster. *Electronics* **2024**, *13*, 1613. https://doi.org/10.3390/electronics13091613

Academic Editor: Hung-Yu Chien

Received: 22 March 2024
Revised: 17 April 2024
Accepted: 20 April 2024
Published: 23 April 2024

1. Introduction

The intersection of Internet of Things (IoT) technologies, Raspberry Pi clusters, and Kubernetes orchestration represents a significant advancement in the realm of environmental monitoring systems. The widespread adoption of IoT technologies presents a transformative shift in how data is collected, analyzed, and utilized to drive innovation. However, the expansion also brings forth substantial challenges. As IoT devices proliferate, managing them and the data they generate becomes increasingly complex. This complexity is technical and extends to logistical and administrative domains, making comprehensive management strategies crucial for successful IoT implementations. This study seeks to position this innovative amalgamation within a broader context, acknowledging its profound ramifications and the increasing imperative for dependable environmental data acquisition. As global environmental challenges intensify, the necessity for real-time and secure monitoring systems becomes increasingly pronounced [1].

In recent years, the domain of environmental monitoring has undergone a paradigmatic transformation, marked by a shift from traditional methodologies towards the cutting-edge integration of sophisticated and revolutionary technologies. The technologies underpinning the IoT infrastructure, such as sensors, networking devices, and integrated software platforms, face several challenges. These include scalability issues due to the massive number of connected devices, interoperability issues between different IoT systems, and significant concerns regarding energy consumption and sustainability of IoT devices. At the forefront of this evolution lies the fusion of Raspberry Pi clusters, esteemed for their computational capabilities and scalability, with the BME680 environmental sensor, renowned for its precision in measuring critical environmental parameters such as temperature, humidity, pressure, and air quality [2]. Within the dynamic landscape of IoT-based environmental monitoring, this integration holds the potential to revolutionize the data acquisition process, enabling real-time insights that inform critical decision-making processes across diverse sectors [3].

Despite these remarkable technological advancements, security remains one of the most pressing issues in IoT deployments. Despite advancements in technology, the heterogeneous and distributed nature of IoT systems makes them vulnerable to a wide range of cyber threats. According to Sarker et al. [4], security in IoT systems encompasses the protection of interconnected devices and the safeguarding of the networks and the data transmitted across them. To address this necessity, this research explores the integration of HashiCorp Vault [5], a state-of-the-art secrets management solution, and OpenID Connect (OIDC) authentication mechanisms. These tools act as guardians, strengthening the environmental monitoring system against potential vulnerabilities and breaches [6]. Furthermore, this study thoroughly assesses security benchmarks, comprising vulnerability scanning, penetration testing, and access control evaluations, to assess the system's resilience against potential threats, intrusions, and vulnerabilities [7].

This introduction provides a comprehensive overview of the current state of environmental monitoring, highlighting the crucial roles of the BME680 sensor and Raspberry Pi clusters within Kubernetes environments [8]. It emphasizes the pressing need to address security concerns in the rapidly evolving landscape of IoT, underscoring the overarching significance of this study. The main objective here is to thoroughly explore these integrated technologies, focusing on the foundational aspects of hardware infrastructure, data integrity, and robust security measures. The primary outcomes of this research endeavor are poised to provide invaluable insights, advocating for the establishment of resilient and trustworthy IoT ecosystems, which are fundamental in ensuring the reliability of environmental data collection—profoundly vital in the context of escalating environmental exigencies.

This document is structured to provide a comprehensive understanding of the research and its findings. Firstly, the related work is described to contextualize the research within existing literature and identify gaps in current knowledge. Following this, the technologies utilized within the proposed IoT solution are detailed to provide insight into the technical framework of the research. The methodology of the research is then presented, covering

aspects such as system setup, data collection procedures, and the analytical techniques employed for data analysis. Subsequently, the results and implications of the findings are discussed in depth. This section focuses on the effectiveness of the enhanced security measures implemented in the IoT solution and their impact on overall system performance. Furthermore, the implications of these findings for the broader field of IoT security are explored, highlighting potential areas for further research and development. Finally, the document concludes with a summary of the research findings and their significance. Additionally, it outlines the contributions of the research to the field of IoT security and identifies potential avenues for future work.

2. Related Work

This chapter provides a comparative analysis of existing solutions and approaches within the field of environmental monitoring systems, particularly those integrating security-enhancing tools like HashiCorp Vault and OIDC. It highlights the distinctions and enhancements offered by the proposed solution.

2.1. Overview of Existing Systems

The landscape of environmental monitoring systems has been characterized by diverse technological integrations aimed at improving data accuracy and system security. Several notable systems have incorporated varying security and data management technologies, each presenting unique approaches and challenges.

2.1.1. Traditional Systems

- Basic Security Measures: traditional environmental monitoring systems often rely on basic security measures such as standard encryption techniques and local storage solutions. These systems typically do not incorporate advanced security protocols, leaving them vulnerable to various security threats;
- Harcoded Credentials: a common practice in these older systems is the use of hard-coded credentials for device access and data retrieval, as noted in studies like those by Chandavarkar et al. [9]. This method poses significant security risks, especially if the source code is exposed or intercepted, leading to potential unauthorized access;
- Limited Authentication Mechanisms: many conventional systems use basic username-password authentication schemes, which do not offer the robustness required in scenarios where sensitive data is involved. This approach is susceptible to various attack vectors, including brute force attacks;
- Example Systems: for instance, a traditional monitoring system described by Gupta et al. [10] used encrypted local databases for storage without dynamic access controls, which could be bypassed if the encryption key was compromised.

These traditional approaches contrast sharply with more modern implementations that use dynamic secret management and robust authentication protocols to enhance security and system resilience. The depicted system's use of dynamic secrets and OIDC presents a significant advancement over these traditional methods by addressing these vulnerabilities comprehensively.

2.1.2. Advanced IoT Solutions

- Generic IoT Frameworks: many advanced IoT systems utilize generic security frameworks that incorporate basic encryption and token-based authentication. For instance, a study by Francisco et al. [11] described an IoT framework leveraging standard SSL/TLS for data transmission security and OAuth 2.0 for user authentication. While these are robust, they often do not cater specifically to the unique security demands of IoT environments, such as the need for managing numerous device identities and securing dynamic interactions among heterogeneous devices;
- Customized IoT Security Solutions: some solutions, like the one presented Maroof et al. [12], involve customized security measures tailored to specific IoT applications.

These solutions may include advanced features like biometric authentication and behavior anomaly detection. However, they often lack the flexibility to be applied universally across different IoT domains without significant modifications.

The presented system represents a significant advancement over existing IoT solutions, offering improvements across three key use cases. Firstly, the implementation of HashiCorp Vault for dynamic secret management enhances security by securely storing credentials and providing them on an as-needed basis. This approach ensures the secure storage of credentials and reduces the attack surface, thereby bolstering the system's overall security posture. Secondly, the integration of OIDC provides a robust authentication solution tailored specifically for modern web and IoT applications. Unlike OAuth 2.0, OIDC adds an identity layer on top of OAuth 2.0, facilitating seamless and secure authentication across services and systems. By reliably authenticating all entities within the IoT ecosystem before granting access to resources, OIDC enhances the security and integrity of the system. Thirdly, the system's scalability and interoperability represent another key advantage. Unlike many existing IoT solutions that are often tailored for specific applications and struggle with scalability, the proposed solution is designed with scalability in mind. Leveraging the lightweight nature of OIDC and the scalable secret management capabilities of HashiCorp Vault, the system can easily adapt to various IoT applications and sizes without sacrificing security.

3. Proposed Solution Overview and Setup

This section explores the intricate details of the environmental monitoring setup, emphasizing the pivotal role of the BME680 sensor, the significance of its capabilities, the hardware infrastructure employed, and the configuration of the Raspberry Pi cluster. Additionally, it discusses the selection of lightweight Kubernetes distributions and underscores the importance of a reliable hardware foundation for data collection and analysis [13]. Furthermore, this paper addresses the security aspects intertwined with our monitoring infrastructure.

3.1. BME680 Sensor for Environmental Monitoring

This sub-chapter details the specific components of our environmental monitoring system, explaining the rationale behind our choices and the functionality of the implemented technologies.

3.1.1. Capabilities of the BME680 Sensor

The choice of the BME680 sensor was driven by its versatility and robustness in environmental sensing. The BME680 can measure traditional parameters like temperature and humidity and also detects a wider range of air quality metrics, including volatile organic compounds (VOCs). This makes it particularly suitable for applications where environmental health and safety are paramount, such as in urban air quality monitoring or industrial settings where air quality can vary significantly. Key capabilities and the motives for using the BME680 sensor are [14]:

- Temperature, Humidity, and Pressure Sensing: the BME680 operates in various modes to enable precise measurements of temperature, humidity, and atmospheric pressure. It offers versatile functionality, including Ultra-Low Power mode, which minimizes power consumption while providing output data at a slower rate. The integrated temperature sensor boasts low noise and high resolution, and it is optimized for evaluating ambient temperature and compensating for temperature variations in other sensors;
- Weight Sensing: The BME680 sensor includes a high-precision and high-resolution barometric weight sensor, offering accurate weight measurements with minimal noise;
- Relative Humidity Sensing: With the ability to measure relative humidity from 0 to 100 percent over a wide temperature range, the BME680 sensor provides valuable insights into atmospheric moisture content. The accuracy of humidity measurements is specified in the corresponding datasheet of the utilized equipment;

- Air Quality Assessment: its exceptional ability to assess air quality by detecting volatile organic compounds (VOCs) and estimating the air quality index (AQI) is instrumental in evaluating the health and safety of the environment. The sensor's ability to detect VOCs and provide accurate real-time data on air quality makes it ideal for use in smart city initiatives aimed at monitoring pollution levels and improving urban living conditions or in factories or plants where hazardous gases or compounds may be present; the BME680 can help ensure that the environment remains within safe limits, alerting to any dangerous changes in air quality.

3.1.2. Significance in IoT Applications

In addition to its technical capabilities, the BME680 sensor holds significance in IoT (Internet of Things) applications. Its compact design and energy-efficient operation render it an optimal choice for integration into IoT networks, contributing to data-driven and intelligent solutions [15]. Incorporating the BME680 sensor into IoT ecosystems enables a diverse range of applications, including but not limited to:

- Urban Environmental Monitoring: Monitoring air quality in urban areas, thereby facilitating pollution control and public health management;
- Smart Building Systems: Enhancing indoor air quality management in smart buildings, ensuring occupant health and comfort;
- Industrial IoT: Enabling real-time monitoring of environmental conditions in industrial settings, ensuring workplace safety and compliance;
- Environmental Research: Supporting environmental research endeavors by providing precise data for climate modeling, ecosystem monitoring, and more [16].

3.2. Hardware Setup and Raspberry Pi Cluster Configuration

3.2.1. Raspberry Pi Cluster Configuration

The proposed solution infrastructure relies on a robust and scalable Raspberry Pi cluster. This cluster configuration consists of interconnected Raspberry Pi devices strategically designed to pool computing power and enhance resilience. The Raspberry Pi cluster configuration is carefully engineered to ensure fault tolerance and redundancy. Networking multiple Raspberry Pi units minimizes single points of failure, guaranteeing uninterrupted data collection and processing (high level of availability). This hardware architecture aligns seamlessly with the demands of environmental monitoring, where data accuracy and reliability are of utmost importance [17]. Additionally, the solution integrates third-party services provided by Amazon Web Services (AWS), including Relational Database Service (RDS) and Simple Storage Service (S3).

3.2.2. Choice of Lightweight Kubernetes Distribution

The careful selection of lightweight Kubernetes distributions facilitates the effective management and orchestration of our Raspberry Pi cluster. Notably, k3s and MicroK8s are chosen for their resource-conscious nature, maintaining essential Kubernetes functionalities while optimizing cluster management for resource-constrained environments [18].

3.3. Security Considerations

Ensuring the security of the environmental monitoring setup is of paramount importance. In addition to the primary security measures of secrets management and OIDC authentication, the following security aspects are integrated into the monitoring infrastructure.

3.3.1. Data Encryption

Data encryption mechanisms, including HTTPS and TLS protocols, are implemented to safeguard sensitive environmental data in transit. These measures ensure that data exchanged between sensors, cluster nodes, and external systems or services remain confidential and tamper-proof [19].

3.3.2. Access Control

Access control policies are implemented to restrict unauthorized access to the monitoring infrastructure. Role-based access control (RBAC) mechanisms are enforced within the Kubernetes cluster, ensuring that only authorized personnel can interact with the system [20]. RBAC was chosen for its effectiveness in managing user/application permissions in a granular yet scalable manner. RBAC allows the definition of roles based on the specific responsibilities and access needs within the IoT environment, thereby minimizing the risk of unauthorized access and ensuring that each application only has access to the resources necessary for their tasks. This is crucial in a distributed IoT setup where multiple users/applications interact with the system across different levels of sensitivity and functionality.

3.4. Architectural Diagram and Workflow

Figure 1 details the architectural diagram that visually represents the environmental monitoring setup. This diagram offers a comprehensive view of the key components, their interconnections, and the flow of data within the monitoring infrastructure.

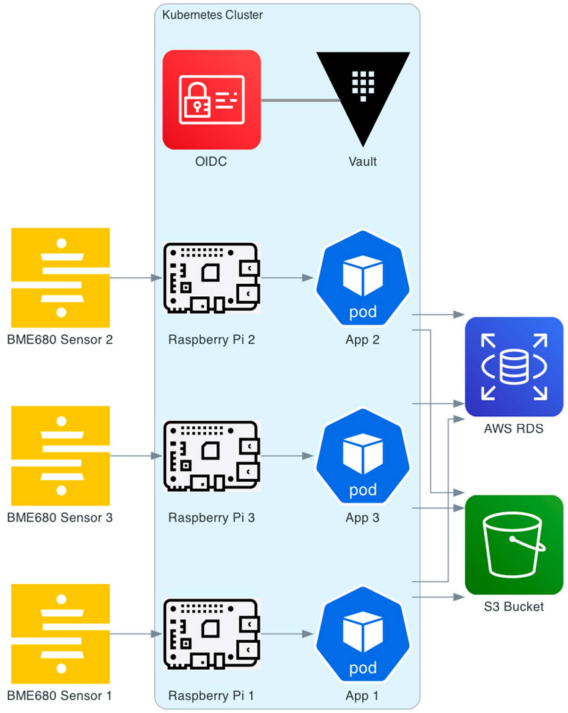

Figure 1. Architectural diagram of the proposed solution.

The architectural workflow ensures that environmental data is collected, transmitted, processed, and analyzed securely and that insights are provided in an actionable format to authorized users and applications, all while maintaining a high standard of security and scalability is detailed below:

- Data Collection (Sensors to Raspberry Pi Nodes): multiple BME680 sensors are deployed in the field, each interface directly with a dedicated Raspberry Pi node. These sensors are responsible for collecting environmental data;
- Data Processing: the data is ingested directly by the applications running in Kubernetes pods on the Raspberry Pi nodes. These applications are responsible for the core data analysis and processing tasks. Data storage operations are also managed at this stage, with processed information being stored in appropriate databases or data lakes within the cloud, such as AWS RDS for structured data and AWS S3 for log data;

- Access Control and Authentication: access to the Kubernetes Pods and, hence, to the applications and processed data are governed by robust security measures. HashiCorp Vault is deployed within the Kubernetes cluster to handle secret management (each node has a Vault agent installed), securing sensitive operations by managing access credentials. OIDC is implemented (installed on the Kubernetes cluster) to authenticate and authorize users and applications, ensuring controlled access to the system's internal API and functionalities.

4. Application Development

The development of the environmental monitoring application is a multifaceted endeavor, encompassing hardware integration, containerization, Kubernetes orchestration, and software development. The technologies selected for the experiments were chosen based on their compatibility with the IoT environment, community support, ease of integration, and ability to scale. In this scientific work are prioritized technologies that have a proven track record in IoT deployments, offer robust community support, and provide a smooth integration pathway with existing components of the proposed architecture. This section provides a detailed exploration of the key phases involved in crafting this proposed solution.

4.1. Hardware Setup

The foundation of the hardware setup lies in the meticulous connection of the BME680 sensor to the Raspberry Pi devices within the cluster. This sensor is interfaced with the Raspberry Pi's I2C bus [21], facilitating smooth data acquisition. Careful consideration is given to the physical placement of the sensor to ensure accurate readings and minimal interference. Proper wiring and configuration protocols are also followed to guarantee reliable communication between the sensor and the Raspberry Pi devices.

4.2. Containerization and Kubernetes Orchestration

4.2.1. Docker (Containerization)

Docker is employed for containerization, enabling the creation of portable images containing the application and its dependencies. Docker was selected as containerization technology due to its widespread adoption and comprehensive ecosystem, including a vast registry of pre-built images (Docker Hub), extensive documentation, and a large community. Docker's tooling and platform support are mature, which reduces the complexity of container management and deployment, especially in a Raspberry Pi-based Kubernetes cluster. Additionally, Docker's integration with Kubernetes, the container orchestration system that is used, is well-established, making it the pragmatic choice for this implementation. Figure 2 depicts the complete Dockerfile of the developed Python v3.7.0 application.

```
FROM python:3.9-slim

WORKDIR /app

COPY . /app

RUN pip install -r requirements.txt

EXPOSE 80

ENV NAME EnvironmentalMonitoringApp

CMD ["python", "app.py"]
```

Figure 2. Application Dockerfile.

4.2.2. Kubernetes Orchestration

Kubernetes functions as the orchestration platform for the environmental monitoring application, providing a robust framework for managing and scaling containers. The decision to employ Kubernetes is driven by several key motives:

- Containerization: Kubernetes simplifies the deployment and scaling of containerized applications [22], ensuring consistent behavior across diverse environments;
- Resource Management: Kubernetes optimize the allocation of computing resources within the cluster, thereby enhancing efficiency and scalability [23];
- Fault Tolerance: Leveraging its self-healing capabilities, Kubernetes ensures uninterrupted operation by automatically replacing failed containers or nodes.

The reliability of the hardware infrastructure is paramount to guarantee uninterrupted data collection and storage, ensure the application's resilience even in challenging conditions, and minimize downtime as much as possible.

4.3. Sensors Data Integration

The core of the application revolves around the seamless integration of sensor data with the software stack. Python, renowned for its versatility in IoT applications, serves as the language of choice for this endeavor.

4.3.1. Data Acquisition

The acquisition of sensor data stands as a fundamental component of the environmental monitoring application, ensuring the accurate and real-time capture of environmental parameters. This process is facilitated by open-source libraries provided by the sensor manufacturers, selected for their reliability and seamless compatibility with the BME680 sensor's suite of features. Each BME680 sensor interfaces with a dedicated Raspberry Pi node via I2C communication facilitated by the chosen libraries. This interface allows for the continuous and reliable transfer of environmental data, which is then preprocessed for consistency and accuracy before being funneled into the Kubernetes-clustered environment for analysis and decision-making. This integration is critical in supporting the overarching objectives of the system—providing a secure, scalable, and efficient framework for environmental data monitoring. The data acquisition phase is intricately woven into the system's architecture, interfacing with the Kubernetes orchestration layer, the HashiCorp Vault for secure management of credentials, and utilizing OIDC protocols to ensure authenticated access to the system's resources.

4.3.2. Data Filtering and Anomaly Detection

Data Filtering and Anomaly Detection play a crucial role in upholding the quality and reliability of collected environmental data. This process is motivated by the need to filter out erroneous or potentially inaccurate measurements while detecting anomalies that may indicate hardware malfunction or extreme environmental conditions.

The process begins by setting predefined thresholds for acceptable temperature and humidity values. For instance, if the temperature falls below a designated minimum threshold or the humidity exceeds a specified maximum threshold, the data is flagged as potentially unreliable and subsequently filtered out. This step is imperative in adverting skewed data from influencing subsequent analysis and decision-making processes.

Moreover, the collected data is subjected to anomaly detection algorithms. These algorithms compare the current sensor readings with historical averages computed over a defined timeframe. If the deviation from the average surpasses predetermined anomaly thresholds, it triggers an anomaly detection flag. This flag serves as an alert mechanism, signaling potential irregularities in the sensor data. A part of the code related to this section is presented in Equation below. The combined efforts of data filtering and anomaly detection contribute to data accuracy and the overall reliability of the environmental monitoring application. By excluding questionable data points and identifying potential anomalies,

the application ensures the collected environmental data's consistency, trustworthiness, and integrity.

T_i—temperature measurement at time *i.*
H_i—humidity measurement at time *i.*
A_i—air quality measurement at time *i.*
N—total number of data points collected.
T_{min}—minimum acceptable temperature.
H_{max}—maximum acceptable humidity.
$\overline{T}$—average temperature
$\overline{H}$—average humidity
$\overline{A}$—average air quality

Data filtering:

$$IF\ Ti \geq T\text{min AND } Hi \leq H\text{max } THEN$$
Include data point in aggregation

Data Aggregation:

$$\begin{aligned} N &\leftarrow N + 1 \\ T\text{sum} &\leftarrow T\text{sum} + Ti \\ H\text{sum} &\leftarrow H\text{sum} + Hi \\ A\text{sum} &\leftarrow A\text{sum} + Ai \end{aligned}$$

Calculate Averages:

$$\begin{aligned} \overline{T} &= \frac{T\text{sum}}{N} \\ \overline{H} &= \frac{H\text{sum}}{N} \\ \overline{A} &= \frac{A\text{sum}}{N} \end{aligned}$$

Detect Anomalies:
Temperature anomalies:

$$\left|Ti - \overline{T}\right| > \epsilon T$$

Humidity anomalies:

$$\left|Hi - \overline{H}\right| > \epsilon H$$

4.3.3. Vault Integration

Vault Integration plays a pivotal role in fortifying the security and confidentiality of sensitive information, including access tokens and credentials, utilized within the proposed application. This integration is motivated by the imperative need to safeguard these critical data elements from unauthorized access or exposure [24].

Vault acts as a centralized repository for storing secrets and credentials, ensuring their secure storage, and limiting access exclusively to authorized entities. When the application requires access to sensitive information, it interfaces with Vault to retrieve the necessary secrets. This approach guarantees that secrets are never hard coded within the application code or exposed in configuration files, significantly reducing the risk of unauthorized access.

Moreover, Vault offers robust access control and auditing capabilities, allowing administrators to define fine-grained policies governing which entities can access specific secrets and under what circumstances. This meticulous control contributes to the overall security

stance of the environmental monitoring application, reducing the potential attack surface and ensuring that solely authorized components and users can access sensitive information.

4.3.4. OIDC Authentication

In the proposed solution, OIDC serves as the cornerstone for securing service interactions within the Kubernetes-clustered environment. The integration of OIDC is pivotal, ensuring that all entities, be they users, applications, or services, are authenticated against a trusted identity provider before any interaction with system resources is allowed. This solution has employed the Authorization Code Flow with Proof Key for Code Exchange (PKCE), as it is well-suited for clients that can securely maintain a client secret. This flow enhances security by providing a high degree of resistance against interception attacks and is particularly well-aligned with the system's operational profile. Trusted OIDC providers have been configured to verify service identities. These providers were selected for their robust security features and compatibility with the Kubernetes setup. The OIDC discovery endpoint is utilized to dynamically fetch the provider's configuration, thereby simplifying the client setup and ensuring that the system remains current with the provider's supported features.

The OIDC integration has been carefully architected to enhance the environmental monitoring system's security framework. Working in tandem with HashiCorp Vault, OIDC ensures that sensitive data such as tokens and credentials are handled securely and that access is rigorously controlled. For example, upon successful authentication, OIDC provides an identity token and an access token. The identity token is used to verify the authenticity of the user, application, or service, while the access token is stored securely in Vault and used to authorize interactions with protected resources.

By implementing OIDC, the proposed application benefits from a robust and standardized identity verification mechanism [25]. This mechanism enhances security and simplifies user and service management within the Kubernetes cluster. Additionally, OIDC provides valuable auditing and logging capabilities, enabling the tracking of access events and helping identify and mitigate potential security threats.

Figure 3 presents the UML Diagram for OIDC.

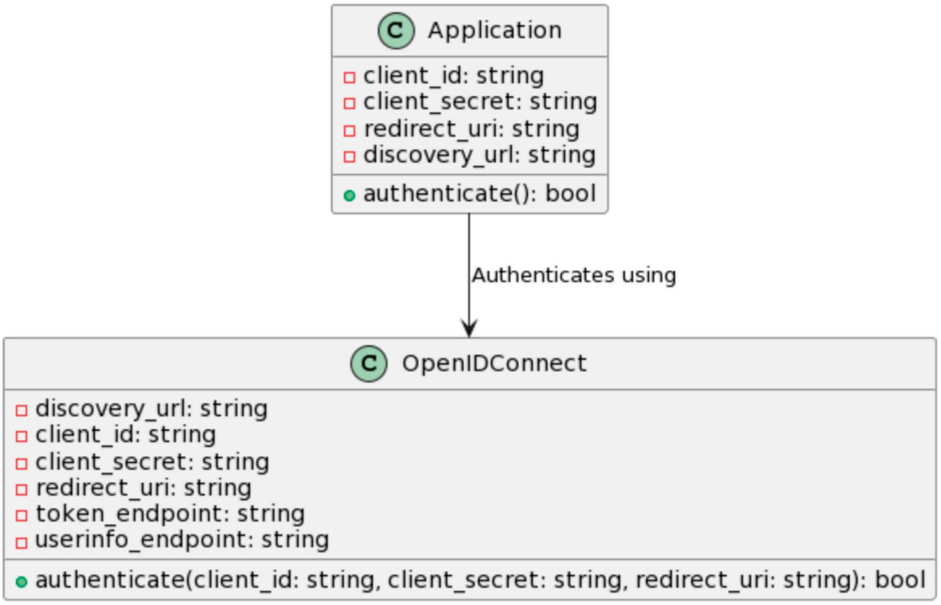

Figure 3. OIDC Authentication UML Diagram.

4.3.5. Database Storage

The utilization of an RDS MySQL database serves as the repository for historical analysis and retrieval of environmental data. Environmental data collected over time is valuable

for trend analysis and historical comparisons. Storing this data in a structured database allows for efficient querying, retrieval, and analysis, facilitating informed decision-making. In selecting the appropriate database technology for our environmental monitoring system, we considered the comparative analysis by Mavrogiorgos et al. [26], which evaluates the performance and suitability of various NoSQL databases, including MongoDB, ArangoDB, and CouchDB, for big data applications. While this study underscores the scalability and flexibility of NoSQL databases in handling large volumes of unstructured data, our application demands strong data consistency and the ability to perform complex SQL queries for data integrity and accurate historical data analysis.

Amazon RDS is chosen for database storage due to its inherent advantages in this project. Firstly, RDS offers a managed database service, relieving the burden of database administration tasks such as provisioning, patching, and backups. This managed approach allows for streamlined operations and frees resources from infrastructure management. Additionally, RDS provides scalability, allowing seamless expansion of database capacity to accommodate growing volumes of environmental data. This scalability ensures sustained performance and reliability as the project scales over time. Furthermore, RDS supports multiple database engines, offering flexibility in choosing the most suitable option based on project requirements. Moreover, RDS incorporates data durability and reliability features, including automated backups, point-in-time recovery, and multi-AZ (Availability Zone) deployment options. These features enhance data integrity and availability, reducing the risk of data loss or downtime.

A code snippet from the application code responsible for storing sensor data is depicted in Figure 4:

```
# Store data in RDS MySQL database
rds_client = Session()
data = EnvironmentalData(
    temperature=temperature,
    humidity=humidity,
    pressure=pressure,
    air_quality=air_quality,
)
rds_client.add(data)
rds_client.commit()
rds_client.close()
```

Figure 4. Database storage code snippet.

4.3.6. S3 Logging

Environmental data logging is pivotal for traceability and auditing, driven by the necessity to maintain a comprehensive record of application activities. Logging data to an AWS S3 bucket ensures data availability and traceability, even in unexpected scenarios. Amazon S3 (Simple Storage Service) is utilized due to several compelling reasons:

- Availability: ensure that log data is securely stored and maintained over time. With an availability rate of 99.99% (4 nines), S3 is highly resilient to data loss or corruption. This level of availability guarantees that log data remains intact and accessible, even in the event of hardware failures or system outages;
- Scalability: efficient storage and access of logs regardless of the volume of data generated. As the project collects and logs environmental data over time, S3 automatically scales to accommodate the increasing volume, ensuring uninterrupted logging operations;
- Fine-grained access controls and permissions: enable secure access controls and permissions based on predefined roles and policies. This ensures that only authorized services or systems can view or manage log files, enhancing security and compliance with data privacy regulations.

Figure 5 illustrates the process of storing a log file in an S3 bucket, exemplifying the seamless integration and reliability of S3 for data storage and traceability.

```
# Upload log file to S3 bucket
log_file_name = 'environmental_data.log'
s3_log_key = os.path.join(s3_folder_name, log_file_name)
s3.upload_file(log_file_name, s3_bucket_name, s3_log_key)
```

Figure 5. Storing log file to S3 code snippet.

Figure 6 below offers a detailed overview of the UML Diagram for the proposed application. In conclusion, the development of the proposed solution is a harmonious convergence of hardware, containerization, Kubernetes orchestration, and software development. By harnessing the capabilities of the BME680 sensor, Python programming, Vault for secrets management, OIDC, RDS storage, and S3 data upload, the application ensures the reliable collection, processing, and storage of environmental data within a Kubernetes-clustered environment based on Raspberry Pies.

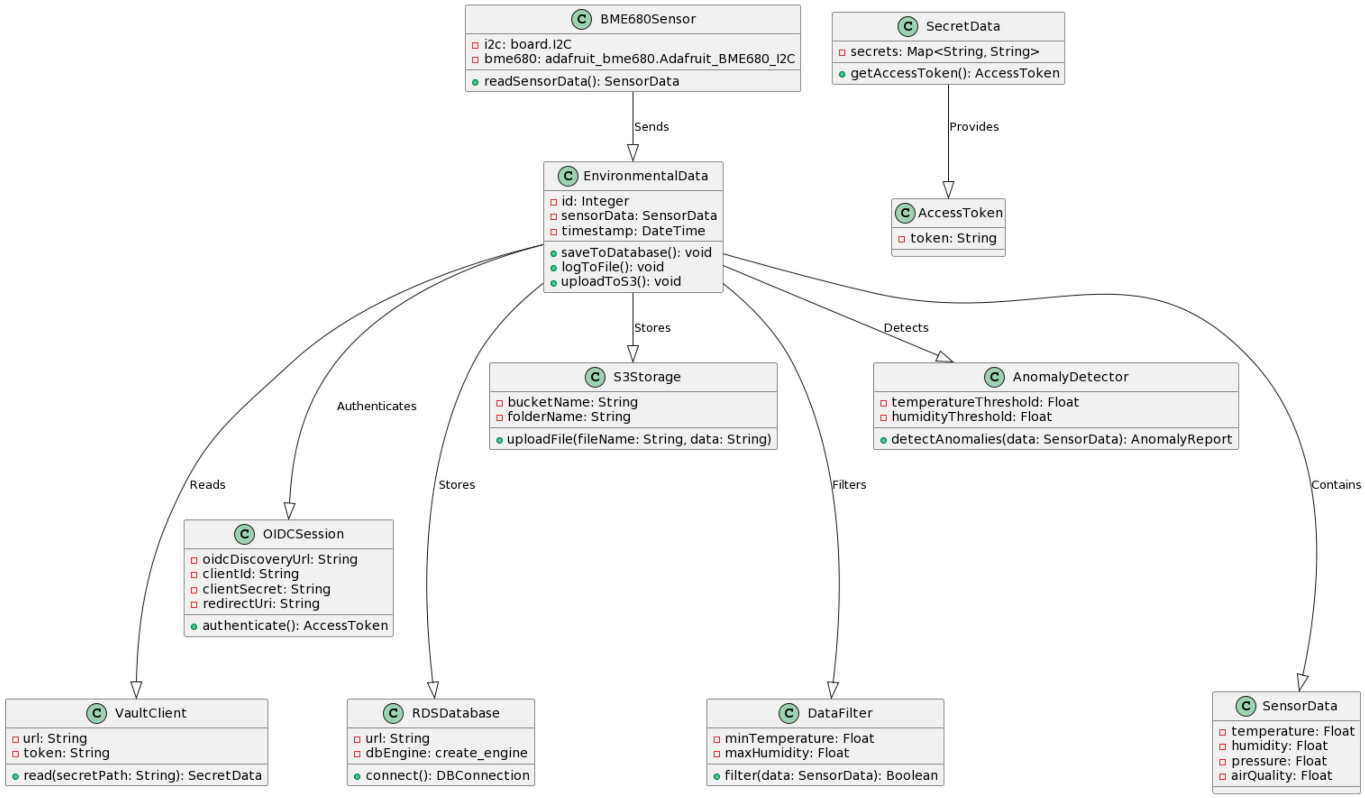

Figure 6. UML Diagram of the proposed solution.

5. Security Measures

In the context of environmental monitoring projects, the implementation of robust security measures is imperative to protect sensitive data, uphold system integrity, and mitigate potential threats. This section presents the comprehensive security measures employed throughout the project, encompassing device security, network security, Kubernetes security, secrets management via Vault, and application authentication through OIDC. In the next Section 6, the benchmark tests and results for the security measures below are presented.

5.1. Device Security

Device security forms the foundational layer of defense against both physical and digital threats. Unauthorized access to Raspberry Pi devices or tampering with sensors can compromise data integrity and the reliability of the system [27]. To mitigate these risks, several security measures have been meticulously put in place:

- Physical Access Controls: Raspberry Pi devices are securely encased to restrict physical access exclusively to authorized personnel through the Access Control List (ACL). An ACL is a list of authorized personnel or entities who are permitted to access Raspberry Pi devices. Each entry in the ACL typically includes the name or identifier of the authorized individual or service. An overview of implemented ACL can be depicted in Table 1;
- Secure Boot Mechanisms: Devices are configured with secure boot processes using cryptographic keys, ensuring the execution of only authorized firmware during startup. This measure safeguards against unauthorized firmware modifications;
- Regular Firmware and Software Updates: A strict regimen of periodic updates is enforced for device firmware and software to promptly address vulnerabilities. An automated update system ensures that devices remain up to date with security patches [28].

Table 1. Access Control List.

User	Path or Service	Permissions
root	/var/lib/app &/var/lib/mysql	Read, Write
mysql	/var/lib/mysql	Read, Write
app	/var/lib/app	Read, Write
app	/var/lib/mysql	Read

5.2. Network Security

Network security plays an instrumental role in safeguarding data transmission between sensors, Raspberry Pi devices, and the central application server. Unauthorized interception or data manipulation during transmission poses significant risks [29]. To ensure secure data exchange, the following measures have been meticulously implemented:

- Utilization of Robust Encryption Protocols: Data transmission employs robust encryption protocols, notably TLS/SSL, to avert eavesdropping and data tampering. Certificates are regularly rotated to maintain security;
- Network Segmentation: A meticulously designed network architecture isolates sensitive data streams, thereby diminishing the potential attack surface. Subnets and security groups are used to segment and compartmentalize data traffic;
- Deployment of Firewalls and Intrusion Detection Systems: Network firewalls are placed to vigilantly monitor and respond to any suspicious network activities.

5.3. Kubernetes Security

Given its role as the orchestration platform, Kubernetes necessitates specific security measures to shield against unauthorized access, privilege escalation, and container vulnerabilities. To bolster the security of the depicted Kubernetes cluster, the ensuing strategies have been meticulously executed:

- Implementation of Role-Based Access Control (RBAC): RBAC is meticulously configured to define intricate access permissions. This ensures that exclusively authorized users and services can interact with cluster resources;
- Adherence to Pod Security Policies: Pod Security Policies are rigorously enforced to constrict container capabilities and amplify overall security. These policies define which security contexts are permissible for pods;

- Routine Container Image Scanning: Systematic container image scanning is undertaken to discern and rectify vulnerabilities within containerized applications. Vulnerabilities are rated, and patches are applied promptly to maintain container security.

5.4. Vault Integration

The integration of HashiCorp Vault is instrumental in securely managing access tokens, API keys, and other sensitive information [30]. This integration elevates overall security through the following measures that are explained below and depicted in Figure 7 through policies:

- Encryption of Secrets: Vault encrypts and securely stores secrets at rest and dynamically generates and manages access credentials. These secrets are stored using advanced encryption algorithms;
- Precise Access Control: Access to Vault is judiciously controlled via access policies. Only authorized users and services can retrieve secrets based on their assigned roles and permissions;
- Dynamic Secrets Generation: Vault generates ephemeral, dynamic secrets with short lifespans. This minimizes exposure in the event of a security breach. Secrets are automatically renewed and rotated.

```
# Precise Access Controls
# Allow read access to the secrets/myapp/* path
path "secrets/myapp/*" {
  capabilities = ["read"]
}

# Allow write access to the secrets/myapp/config path
path "secrets/myapp/config" {
  capabilities = ["create", "update", "delete"]
}

# Define a policy that grants read access to certain paths
path "secrets/myapp/*" {
  capabilities = ["read"]
}

# Dynamic Secrets Generation
# Configure a dynamic secrets engine for PostgreSQL
path "database/creds/myapp" {
  capabilities = ["read"]
  allowed_parameters = {
    "db_name" = "myappdb"
  }
  ttl = "1h" # Set the lifespan of the dynamic secrets
  max_ttl = "24h" # Set the maximum lifespan
}
```

Figure 7. Vault Policies.

5.5. OIDC for Application Authentication and Authorization

In our proposed project, OpenID Connect (OIDC) serves a crucial role in authenticating and authorizing applications rather than individual users [31]. This approach is vital for securing application-to-application communication and granting specific rights to applications for AWS resource access. OIDC is adeptly implemented to facilitate secure application authentication and authorization, enhancing overall project security through the following means:

- Facilitation of Secure Application Authentication: OIDC ensures that only trusted applications can access AWS resources by verifying their identities. Applications are granted unique client IDs and client secrets, which they use to authenticate themselves securely;

- Precise Authorization Policies for Applications: OIDC allows for fine-grained authorization policies to be defined and enforced. Applications are assigned specific roles and permissions, ensuring they can only access the AWS resources they have been authorized;
- Enhanced Security with OAuth 2.0: OIDC builds upon the OAuth 2.0 framework, providing secure authorization for applications. OAuth 2.0 ensures that applications are granted access tokens with restricted scopes, minimizing potential security risks.

The policies used in the proposed solution are described in Figure 8.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:ListTagsForResource",
        "rds:ListTagsForResources",
        "rds-db:connect",
        "rds-db:connect*",
        "rds-db:describe*",
        "rds-db:authorize*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:CreateBucket",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::s3-bucket-name",
        "arn:aws:s3:::s3-bucket-name/*"
      ]
    }
  ]
}
```

Figure 8. OIDC policies.

Table 2 depicts a summary of how each technology contributes to the security of the proposed solution, ensuring that the deployed measures are both sufficient and efficient.

Table 2. Summary of Security Technologies Used.

Technology	Application Area	Application Area
Secure Boot and Firmware Signing	Device Security	Prevents unauthorized software installation and protects against malicious firmware modifications.
Vault	Secrets Management	Secures API keys, passwords, and other sensitive data with encryption and dynamic secrets management. Reduces Risks associated with static secrets.
Network Security	Network security	Ensures that data transmitted between devices and servers is encrypted. Protect against unauthorized network access and monitor suspicious activities.
RBAC	Access management	Defines and enforces access controls and permissions in a granular manner.
OIDC	Authentication	Provides robust identity verification and authentication, enhancing system access control.

6. Security Benchmark Tests and Results

6.1. Security Benchmark Tests

This section presents the comprehensive security benchmarking activities conducted to evaluate the robustness of the environmental monitoring application. The proposed solution delves into the details of the security benchmark tests performed to assess the application's resilience against potential threats. The following subsections provide insights into various aspects of security testing and assessment.

6.1.1. Vulnerability Scanning and Assessment

The results of vulnerability scanning are examined, highlighting potential weaknesses and areas for improvement. The assessment was conducted on the Raspberry Pi Cluster with Kubernetes, and the associated environmental monitoring solution was performed by OpenVAS, which aimed to identify potential vulnerabilities and weaknesses within the environment. This process is essential for proactively enhancing the system's security posture and preventing potential security breaches [32].

The comprehensive scanning and assessment approach covered a wide range of aspects, including the identification of outdated software components, potential unauthorized access points, encryption practices, application vulnerabilities, and more, using the capabilities of OpenVAS [33]. The findings provide valuable insights into the security state of the environment and serve as a foundation for implementing necessary security measures.

6.1.2. Penetration Testing Methodologies and Findings

The methodologies applied in penetration testing are outlined, and the findings and vulnerabilities discovered during these tests are discussed below.

1. Methodologies applied:
    - Network Scanning: Comprehensive network scanning was performed to identify open ports, services, and potential entry points within the Raspberry Pi Cluster with Kubernetes and the Environmental Monitoring Application [34]. The widely used tool, Nmap, was employed to conduct this assessment [35];
    - Web Application Testing: Extensive testing of web applications within the environment was carried out. Both automated scanning and manual testing techniques were applied to detect vulnerabilities related to SQL injection and cross-site scripting (XSS). SQL injection scripts and cross-site scripts were used for these tests;
    - Credential Testing: A thorough examination of password policies and the strength of credentials was undertaken.
2. Key Findings and Vulnerabilities:
    - Exposed Services: Network scanning unveiled one open service, which includes SSH access also for app users, which raised concerns regarding unnecessary exposure;
    - Web Application Vulnerabilities: Web application testing could not find any critical vulnerability, like SQL injection in the application;
    - Credential leaks: Weak credentials were not identified because all of them are stored in Vault.

These methodologies and findings collectively contribute to a comprehensive assessment of potential environmental security risks. This assessment serves as a foundation for strengthening the security posture and ensuring the resilience of the proposed solution.

6.1.3. Secrets Management and OIDC Authentication Assessments

In the proposed solution, the following assessments have been made to test the effectiveness of the secrets management and OIDC authentication solutions, ensuring secure storage and retrieval of sensitive information [36].

- Security Policy Assessment: Vault's security policies were meticulously examined to ensure that access control and authorization were well-defined. For instance, specific policies were configured to grant only the necessary permissions for the application to access sensitive secrets while enforcing strict access restrictions for other users and services;
- Dynamic Secrets Generation: Vault's ability to dynamically generate short-lived, ephemeral secrets was evaluated. This feature aligns with the presented security strategy to minimize the exposure of sensitive data. For instance, the proposed solution received dynamically generated credentials with short lifespans, reducing the risk of potential misuse in the event of a security breach;
- Secrets Rotation and Management: Vault's secrets rotation and management capabilities were also put to the test. The depicted security strategy ensured that secrets were automatically renewed and rotated as per defined policies. This proactive approach to secret management enhances security by reducing the window of vulnerability associated with static secrets;
- Access Control and Authentication: OIDC employs robust authentication methods to verify the identity of applications and services. In the proposed solution, OIDC authentication was meticulously configured, allowing only trusted applications to access Vault. We tested the authentication process to confirm that unauthorized entities were effectively denied access.

The assessment of Vault and OIDC in the context of the Raspberry Pi Cluster and Kubernetes-based Environmental Monitoring Application reaffirmed its ability to securely manage and protect sensitive information [37]. This evaluation underscores the importance of a robust secrets management solution and OIDC authentication in enhancing the overall security of the proposed solution.

6.2. Results and Insights

This subsection focuses on the outcomes of the presented application, the impact of security measures, and the findings from security benchmarking.

6.2.1. Contributions of Security Measures

The role of implemented security measures in the project's success is of paramount importance. The comprehensive approach to securing the proposed application, Raspberry Pi Cluster, and Kubernetes infrastructure has yielded significant benefits. Notably, these measures have fortified the project's overall security posture by establishing robust layers of protection. The meticulous implementation of security policies, access controls, and secrets management has collectively contributed to the project's resilience against potential threats. The integration of HashiCorp Vault for secrets management and OpenID Connect (OIDC) for secure application authentication and authorization stands as a testament to the project's commitment to security.

6.2.2. Analysis of Benchmark Test Results

The results of security benchmark tests have been a critical component in assessing and enhancing the project's security. The thorough examination of vulnerabilities, penetration testing, and access control effectiveness has provided invaluable insights. The findings from these tests have enabled the identification and mitigation of potential security weaknesses. The security benchmarking process has not only served to validate the project's security measures but has also acted as a proactive mechanism for strengthening the security posture. The project's commitment to rigorous security benchmarking reflects its dedication to maintaining a robust and resilient environment.

6.2.3. Addressing Limitations and Challenges

While the security measures and benchmarking processes have been instrumental in fortifying the project's security, it is essential to acknowledge that no security strategy is

without its limitations and challenges. During security benchmarking, certain limitations and challenges were encountered, like not having the possibility to integrate scanning libraries that detect mining scripts or checking the compute consumption anomalies that can appear if something breaches the entire fortified environment. These included potential areas for further improvement and fine-tuning of security controls. Addressing these challenges is seen as an ongoing process, and the project remains committed to evolving its security practices to stay ahead of emerging threats. The insights gained from security benchmarking have highlighted areas of improvement and paved the way for continuous security enhancements.

In summary, the project's commitment to security measures, rigorous security benchmarking, and an adaptive approach to addressing limitations and challenges have collectively contributed to the project's success. The project's security measures have established a resilient foundation, and the insights from benchmark tests have played a pivotal role in shaping its security strategy. By acknowledging limitations and continuously striving for improvement, the project remains dedicated to upholding a high standard of security in its environmental monitoring efforts.

7. Conclusions

The primary audience for this research includes environmental scientists, IoT security professionals, and technology developers focused on sustainable and secure monitoring solutions. Additionally, policymakers and environmental agencies may also benefit from the insights provided by the proposed system to enhance regulatory practices related to environmental monitoring.

7.1. Key Takeaways from the Project

The secure application, developed on a Raspberry Pi Cluster with Kubernetes orchestration, has yielded noteworthy insights into the significance of security in data-intensive IoT projects. This project's success is attributed to a multifaceted security approach, encompassing device security, network security, secrets management, and secure authentication.

7.2. Success and Contributions

The secure proposed application stands as a testament to the project's achievements and contributions in the realm of secure data handling. Key factors contributing to its success include the integration of Vault for secrets management and OpenID Connect (OIDC) for secure application authentication and authorization. These measures have fortified the project's security and set a high standard for security practices in similar endeavors.

7.3. Significance of Security Measures and Benchmarking Results

The project's security measures and insights from security benchmarking have assumed critical importance. The meticulous implementation of security policies, access controls, and secrets management has established a robust security foundation. The comprehensive security benchmarking process, encompassing vulnerability scanning, penetration testing, and threat modeling, has acted as a continuous feedback mechanism, enabling the identification and remedy of potential vulnerabilities.

7.4. Limitations

This research, while extensive, operates within several constraints that must be acknowledged to fully understand the scope and applicability of the findings. These limitations are detailed below:

- Network Dependency: In environments where connectivity is poor or highly variable, the system's ability to transmit real-time data and processing could be compromised. This limitation is significant as it affects deployment scalability in remote or underdeveloped regions where such technology is often needed the most;

- Hardware Limitations: The system's design and testing were conducted with specific types of sensors and hardware configurations, which may not be universally applicable. Compatibility issues could arise when integrating with different types or older versions of hardware, which might limit the system's applicability without additional customization or adaptation;
- Environmental Variability: The environmental monitoring conducted was limited to specific climates and settings, potentially overlooking unique challenges presented by extreme or uncommon environmental conditions. Therefore, the system's adaptability to a wide range of environmental factors remains partially untested.

To ensure that the findings of this research are effectively communicated to both the scientific community and stakeholders in relevant sectors, a multifaceted dissemination strategy has been planned. Firstly, the results will be presented at international conferences attended by environmental scientists and IoT security experts. Additionally, multiple workshops and seminars are planned to be conducted that will allow this research to engage directly with industry professionals and technology developers. Collaborative efforts with industry partners are also being explored to implement the findings in practical applications, thereby reaching a broader audience. Moreover, the development of educational materials based on the presented research will further aid in transferring knowledge to academia and industry training programs, ensuring the longevity and impact of our work.

In essence, this project underscores the paramount significance of security in environmental monitoring and IoT applications. Through the diligent implementation of security measures and the rigorous scrutiny of security benchmarking, the project has ensured data reliability while setting a precedent for security standards in analogous projects. The success of the proposed solution serves as an empirical validation of the efficacy of a holistic and proactive security strategy.

8. Future Work

In contemplating potential avenues for future work and improvements to the project, several areas stand out as fertile ground for research and development. These areas encompass both security enhancements and advancements in environmental monitoring capabilities.

8.1. Enhancements in Security

Future work in the realm of security may include:

- Automated Incident Response: Implementing automated incident response systems that can proactively address security incidents in real time, reducing response times and potential damages;
- Deception Technology: Utilizing deception technology, such as honeypots and deceptive networks, to mislead and divert attackers, gathering valuable threat intelligence and buying time for response [38];
- Behavioral Analytics: Implementing advanced behavioral analytics to detect anomalies in user and system behavior, enabling early identification of security threats based on deviations from normal patterns [39].

8.2. Advancements in Application Technologies

- Advanced Sensor Integration: Incorporating advanced sensors and data fusion techniques to enhance the precision and breadth of environmental data collection, allowing for more comprehensive monitoring [40];
- Machine Learning Integration: Utilizing machine learning algorithms to predict environmental trends and anomalies based on historical data, enabling more proactive environmental management [41].

To effectively address the limitations identified in this study and to advance this research according to the proposed future steps, a detailed time plan over the next three years has been developed:

- Short-term Goals (Next 6 months):
  - Literature Review and Team Expansion: complete a thorough literature review by [Month, Year] and recruit two PhD students focused on IoT security solutions;
  - Preliminary Data Collection: initiate additional environmental data collection using enhanced sensor setups.
- Medium-term Goals (6–12 months):
  - Prototype Development: develop a prototype incorporating advanced sensors;
  - Community Engagement: Host a series of webinars and workshops to gather stakeholder feedback.
- Long-term Goals (1–3 years):
  - Field Testing: begin comprehensive field testing in multiple locations;
  - Research Publication and Dissemination: target submission of findings to high-impact journals and presentations at international conferences.
  - Technology Deployment: collaborate with industry partners for the deployment of validated technologies.

Each phase is backed by a planned allocation of resources and technical support from the university's research labs. This strategic plan addresses the immediate limitations and sets a robust foundation for advancing research on secure environmental monitoring systems.

This project lays the groundwork for a promising environmental monitoring and IoT security future. The outlined enhancements offer a pathway to heightened security resilience and more precise environmental insights. By embracing these opportunities, the project can remain at the forefront of innovation, inviting collaboration and contributing to the broader discourse. Its legacy lies not only in its achievements but in the potential for a future where environmental monitoring and security synergize to address evolving challenges, ensuring a safer and more sustainable world.

Author Contributions: Conceptualization, I.-C.D. and O.P.S.; methodology, I.-C.D.; software, I.-C.D.; validation, O.P.S. and L.M.; formal analysis, M.M. and A.S.; investigation, M.M. and A.S.; resources, I.-C.D., O.P.S. and L.M.; data curation, M.M. and A.S.; writing—original draft preparation, I.-C.D. and O.P.S.; writing—review and editing, I.-C.D., M.M., O.P.S., A.S. and L.M.; visualization, I.-C.D.; supervision, O.P.S. and L.M.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Desnanjaya, I.G.M.N.; Arsana, I.N.A. Home security monitoring system with IoT-based Raspberry Pi. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 1295–1302. [CrossRef]
2. Wall, D.; McCullagh, P.; Cleland, I.; Bond, R. Development of an Internet of Things Solution to Monitor and Analyse Indoor Air Quality. *Internet Things* **2021**, *14*, 100392. [CrossRef]
3. Kumru, C.F.; Vural, M.S. Design and Application of IoT based weather station for high voltage. *Mühendislik Bilim. Tasarım Derg.* **2023**, *11*, 1190–1201. [CrossRef]
4. Sarker, I.H.; Khan, A.I.; Abushark, Y.; Alsolami, F. Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions. *Mob. Netw. Appl.* **2022**, *28*, 296–312. [CrossRef]
5. Santos, J.; Wauters, T.; Volckaert, B.; De Turck, F. Towards Network-Aware Resource Provisioning in Kubernetes for Fog Computing Applications. In Proceedings of the IEEE Conference on Network Softwarization (NETSOFT), Paris, France, 24–28 July 2019.

6. Shamim, M.S.I.; Bhuiyan, F.A.; Rahman, A. XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices. In Proceedings of the IEEE Secure Development (SecDev), Virtual, 28–30 September 2020.
7. Kaur, K.; Garg, S.; Kaddoum, G.; Ahmed, S.H.; Atiquzzaman, M. KEIDS: Kubernetes-Based Energy and Interference Driven Scheduler for Industrial IoT in Edge-Cloud Ecosystem. *IEEE Internet Things J.* **2019**, *2*, 4228–4237. [CrossRef]
8. Donca, I.-C.; Stan, O.; Miclea, L. Proposed model for a Microservices Cluster. In Proceedings of the 21st International Carpathian Control Conference (ICCC), Virtual, 27–29 October 2020.
9. Chandavarkar, B.R. Hardcoded Credentials and Insecure Data Transfer in IoT: National and International Status. In Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020.
10. Quamara, M.; Gupta, B.B. Role of Software-Defined Networking (SDN) in Internet of Things (IoT) Security: Attacks and Countermeasures: Principles, Algorithm, Applications, and Perspectives. In *Computer and Cyber Security*; Auerbach Publications: Boca Raton, FL, USA, 2018.
11. Da Silva Francisco, G.; da Silva, A.A.A.; de Azevedo, M.T.; Ueda, E.T.; Guelfi, A.E.; Pérez-Alcázar, J.J. Vulnerability Detection in Intelligent Environments Authenticated by the OAuth 2.0 Protocol over HTTP/HTTPS. *Int. J. Comput. Netw. Inf. Secur.* **2024**, *16*, 1–13.
12. Maroof, U.; Shaghaghi, A.; Michelin, R.; Jha, S. iRECOVer: Patch your IoT on-the-fly. *Future Gener. Comput. Syst. J.* **2022**, *132*, 178–193. [CrossRef]
13. Menouer, T. KCSS: Kubernetes container scheduling strategy. *J. Supercomput.* **2021**, *77*, 4267–4293. [CrossRef]
14. Palacín, J.; Rubies, E.; Clotet, E.; Martínez, D. Classification of Two Volatiles Using an eNose Composed by an Array of 16 Single-Type Miniature Micro-Machined Metal-Oxide Gas Sensors. *Sensors* **2022**, *22*, 1120. [CrossRef]
15. Harkai, A. Main Characteristics and Cybersecurity Vulnerabilities of IoT Mobile Devices. In *Proceedings of 22nd International Conference on Informatics in Economy*; Springer: Singapore, 2024; p. 367.
16. Chataut, R.; Phoummalayvane, A.; Akl, R. Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0. *Sensors* **2023**, *23*, 7194. [CrossRef]
17. Hadiwandra, T.Y.; Candra, F. High Availability Server Using Raspberry Pi 4 Cluster and Docker Swarm. *IT J. Dev.* **2021**, *6*, 43–51. [CrossRef]
18. Bohm, S.; Wirtz, G. Profiling Lightweight Container Platforms: MicroK8s and K3s in Comparison to Kubernetes. In Proceedings of the 13th Central European Workshop on Services and their Composition, Bamberg, Germany, 12 March 2021.
19. Phuc, L.H.; Phan, L.-A.; Kim, T. Traffic-Aware Horizontal Pod Autoscaler in Kubernetes-Based Edge Computing Infrastructure. *IEEE Access* **2022**, *10*, 18966–18977. [CrossRef]
20. Rostami, G. Role-based Access Control (RBAC) Authorization in Kubernetes. *J. ICT Stand.* **2023**, *3*, 237–260. [CrossRef]
21. Murali, A.; Kakarla, H.K.; Priyadarshini, G.M.A. Improved design debugging architecture using low power serial communication protocols for signal processing applications. *Int. J. Speech Technol.* **2021**, *24*, 291–302. [CrossRef]
22. Fathoni, H.; Yang, C.-T.; Chang, C.-H.; Huang, C.-Y. Performance Comparison of Lightweight Kubernetes in Edge Devices. In *Pervasive Systems, Algorithms and Networks*; Springer: Berlin/Heidelberg, Germany, 2019.
23. Pan, Z.; Hur, B.; Myles, K.; Adelman, Z. Development of Raspberry Pi 4 B and 3 B Micro-Kubernetes Cluster and IoT System for Mosquito Research Applications+. *Computation* **2022**, *10*, 221. [CrossRef]
24. Suganthi Evangeline, C. IoT-Based Triple Way Access Control for Secured Asset Storage. In Proceedings of the 9th International Conference on Advanced Computing and Communication Systems, Tamilnadu, India, 17–18 March 2023.
25. Shah, V.; Khang, A.; Abdullayev, V.H.; Hahanov, V. *Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy*; CRC Press: Boca Raton, FL, USA, 2023.
26. Mavrogiorgos, K.; Kiourtis, A.; Mavrogiorgou, A.; Kyriazis, D. A comparative study of MongoDB, ArangoDB and CouchDB for big data storage. In Proceedings of the 5th International Conference on Cloud and Big Data Computing, Liverpool, UK, 8–14 August 2021.
27. Blessing, E.; Potter, K.; Klaus, H. Security and Privacy in IoT: Considerations for Securing IoT Devices. Available online: https://www.researchgate.net/publication/377853082_Security_and_Privacy_in_IoT_Considerations_for_securing_IoT_devices#:~:text=Considerations%20for%20securing%20IoT%20devices%20are%20presented%20across%20device,%20network,and%20secure%20data%20storage%20practices (accessed on 18 March 2024).
28. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.A.; Quwaider, M. IoT Privacy and Security: Challenges and Solutions. *Appl. Sci.* **2020**, *10*, 4102. [CrossRef]
29. Karie, N.M.; Sahri, N.M.; Yang, W.; Valli, C.; Kebande, V.R. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. *IEEE Access* **2021**, *9*, 121975–121995. [CrossRef]
30. Aqeel, M.; Ali, F.; Iqbal, M.W.; Rana, T.A.; Arif, M.; Auwul, R. A Review of Security and Privacy Concerns in the Internet of Things (IoT). *J. Sens.* **2022**, *6*, 5724168.
31. Ojha, G.; Kumar, R.; Shrestha, R. SmartVault: Trustless Vault Using IOT. Available online: https://www.researchgate.net/publication/337001671_SmartVault_Trustless_vault_using_IOT (accessed on 15 March 2024).
32. Olaniyi, O.O.; Okunleye, O.J.; Olabanji, S.O.; Asonze, C.U.; Ajayi, S.A. IoT Security in the Era of Ubiquitous Computing: A Multidisciplinary Approach to Addressing Vulnerabilities and Promoting Resilience. *Asian J. Res. Comput. Sci.* **2023**, *16*, 354–371. [CrossRef]

33. Muharrom, M.; Saktiansyah, A. Analysis of Vulnerability Assessment Technique Implementation on Network Using OpenVas. *Int. J. Eng. Comput. Sci. Appl.* **2023**, *2*, 51–58.
34. Muniz, J.; Lakhani, A. *Penetration Testing with Raspberry Pi*; Packt Publishing: Birmingham, UK, 2015.
35. Moreta, N.; Aragon, D.; Ona, S.; Jaramillo, A.; Ibarra, J.; Jahankhani, H. Comparison of Cybersecurity Methodologies for the Implementing of a Secure IoT Architecture, Cybersecurity in the Age of Smart Societies. In Proceedings of the 14th International Conference on Global Security, Safety and Sustainability, London, UK, 7–8 September 2022.
36. Tembhurne, J.V.; Diwan, T.; Jain, T.K. *IoT Security and Privacy, Chapter in Book: Modern Approaches in IoT and Machine Learning for Cyber Security*; Springer: Cham, Switzerland, 2023.
37. Yang, M.; Ahmed, T.; Inagaki, S.; Sakiyama, K.; Li, Y.; Hara-Azumi, Y. Hardware/Software Cooperative Design Against Power Side-Channel Attacks on IoT Devices. *IEEE Internet Things J.* **2024**. [CrossRef]
38. Pour, M.S.; Khoury, J.; Bou-Harb, E. HoneyComb: A Darknet-Centric Proactive Deception Technique for Curating IoT Malware Forensic Artifacts. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022.
39. Sciullo, L.; De Marchi, A.; Trotta, A.; Montori, F.; Bononi, L.; Di Felice, M. Relativistic Digital Twin: Bringing the IoT to the future. *Future Gener. Comput. Syst.* **2023**, *153*, 521–536. [CrossRef]
40. Blessing, E.; Potter, K.; Klaus, H. Future Trends: Emerging Trends in Predictive Maintenance and IoT. Available online: https://www.researchgate.net/publication/377864340_Future_Trends_Emerging_trends_in_predictive_maintenance_and_IoT (accessed on 21 March 2024).
41. Iqbal, S.; Qureshi, S. Securing IoT Using Supervised Machine Learning. In Proceedings of the International Conference on Artificial Intelligence of Things, Istanbul, Turkey, 10–11 June 2024.