АҚПАРАТТЫҚ-КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР / ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ / INFORMATION AND COMMUNICATION TECHNOLOGIES

DOI 10.54596/2958-0048-2024-3-155-164
UDK 504.064.3
IRSTI 87.01.73

DESIGN AND IMPLEMENTATION OF A SCALABLE IOT-BASED REAL-TIME ENVIRONMENTAL MONITORING AND ALARM SYSTEM: AN EXPERIMENTAL STUDY

Celal Ceken[1*]
[1]Sakarya University, Sakarya, Türkiye*
[*]Corresponding author: celalceken@sakarya.edu.tr

Abstract

Digital transformation is crucial for organizations to survive, be competitive, and grow in the modern age. IoT is the key to enabling this transformation by connecting devices and optimizing processes. This paper presents the design and implementation of a generic IoT-based real-time environmental monitoring and alarm system. The platform is validated by applying it to a manufacturing plant scenario, where various sensors simulate industrial conditions. Scalable message distribution systems such as MQTT and Apache Kafka facilitate reliable data transmission. A microservice architecture is constructed for the backend services to ensure uninterrupted and high throughput services in the application domain. Instead of deploying a real WSN, traffic generation services were chosen to minimize costs, provide greater control and flexibility, and facilitate faster, scalable testing in a controlled environment. The platform also features an integrated alarm system with an event definition module, which allows users to define custom action rules. This flexible, scalable, and resilient architecture can be used across a wide range of application domains that require digital transformation. The experimental study demonstrates the platform's capabilities and great potential for broader IoT applications.

Keywords: digital transformation; Internet of Things; IoT-based monitoring; alarm system; scalable web architecture; fault-tolerant web architecture.

ЗАТТАР ИНТЕРНЕТІНЕ НЕГІЗДЕЛГЕН МАСШТАБТАЛАТЫН ҚОРШАҒАН ОРТАНЫ БАҚЫЛАУ ЖӘНЕ НАҚТЫ УАҚЫТТАҒЫ ДАБЫЛ ЖҮЙЕСІН ӘЗІРЛЕУ ЖӘНЕ ЕНГІЗУ: ЭКСПЕРИМЕНТТІК ЗЕРТТЕУ

Celal Ceken[1*]
[1]Сакария университеті, Сакария, Түркия*
[*]Хат-хабар үшін автор: celalceken@sakarya.edu.tr

Андапта

Цифрлық трансформация қазіргі дәуірдегі ұйымдардың өмір сүруі, бәсекеге қабілеттілігі және өсуі үшін өте маңызды. IoT-бұл трансформацияны құрылғыларды қосу және процестерді оңтайландыру арқылы қамтамасыз етудің кілті. Бұл мақалада IoT негізіндегі әмбебап қоршаған ортаны бақылау және нақты уақыттағы дабыл жүйесін жобалау және енгізу ұсынылған. Платформа әр түрлі датчиктер өндірістік жағдайларды имитациялайтын өндірістік кәсіпорынның сценарийіне қолдану арқылы тексеріледі. MQTT және Apache Kafka сияқты масштабталатын хабар тарату жүйелері деректердің сенімді берілуін қамтамасыз етеді. Ішкі қызметтер үшін қолданбалы салада үздіксіз және жоғары өнімді қызметтерді қамтамасыз ететін микросервистік архитектура құрылады. Нақты WSN орналастырудың орнына шығындарды азайту, көбірек бақылау мен икемділікті қамтамасыз ету және бақыланатын жүйеде жылдам масштабталатын тестілеуді жеңілдету үшін трафикті генерациялау қызметтері таңдалды.

Кілт сөздер: IoT тарату жүйелері, цифрлық трансформация, микросервис архитектурасы, діріл датчиктері, дабыл жүйесі.

РАЗРАБОТКА И ВНЕДРЕНИЕ МАСШТАБИРУЕМОЙ СИСТЕМЫ МОНИТОРИНГА ОКРУЖАЮЩЕЙ СРЕДЫ И СИГНАЛИЗАЦИИ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ ИНТЕРНЕТА ВЕЩЕЙ: ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

Celal Ceken[1*]
[1]Университет Сакарья, Сакарья, Турция*
[*]Автор для корреспонденции: celalceken@sakarya.edu.tr

Аннотация

Цифровая трансформация имеет решающее значение для выживания, конкурентоспособности и роста организаций в современную эпоху. IoT является ключом к обеспечению этой трансформации путем подключения устройств и оптимизации процессов. В этой статье представлены проектирование и реализация универсальной системы мониторинга окружающей среды и сигнализации в реальном времени на основе IoT. Платформа проверена путем ее применения к сценарию производственного предприятия, где различные датчики имитируют промышленные условия. Масштабируемые системы распределения сообщений, такие как MQTT и Apache Kafka, обеспечивают надежную передачу данных. Для внутренних служб создается микросервисная архитектура, обеспечивающая бесперебойные и высокопроизводительные услуги в прикладной области. Вместо развертывания реальной WSN были выбраны службы генерации трафика для минимизации затрат, обеспечения большего контроля и гибкости, а также упрощения более быстрого масштабируемого тестирования в контролируемой среде. Платформа также имеет интегрированную систему сигнализации с модулем определения событий, который позволяет определять пользовательские правила действий. Эта гибкая, масштабируемая и устойчивая архитектура может использоваться в широком спектре прикладных областей, требующих цифровой трансформации. Экспериментальное исследование демонстрирует возможности платформы и большой потенциал для более широких приложений IoT.

Ключевые слова: распределительные системы IoT, цифровая трансформация, архитектура микросервисов, датчики вибрации, система сигнализации.

Introduction

In today's rapidly evolving technological landscape, digital transformation is crucial for organizations to maintain competitiveness and efficiency. The Internet of Things (IoT) plays a critical role in driving this transformation by enabling the interaction of devices and systems to facilitate real-time data collection, monitoring, and control. While IoT has a broad range of applications across various sectors, its potential impact on industrial environments is also significant. IoT-based monitoring systems can enhance safety, optimize processes, and reduce operational costs, making them vital components of modern industrial operations.

The construction of an IoT architecture that is resilient, highly available, fault-tolerant, and scalable is a big challenge due to the IoT ecosystem's complex nature. These systems involve numerous interconnected devices, each generating continuous data streams, which need to be processed, stored, and monitored in real-time. Ensuring high availability and fault tolerance requires a robust infrastructure capable of handling network interruptions, device failures, and varying loads without compromising performance. Scalability adds another layer of difficulty, as the architecture must accommodate the seamless addition of new devices, services, and data streams while maintaining system reliability. Balancing these factors while maintaining optimal efficiency and responsiveness presents a significant technical challenge in IoT system design.

Over the past decade, the importance of IoT has increased rapidly, leading to a growing body of research in the literature. The authors in [1] present a cloud-based architecture for livestock health monitoring using AWS and Python for real-time data analytics. They concluded that integrating IoT Cloud and IoT devices enhances farm efficiency and animal welfare, promoting sustainable farming practices and setting a benchmark for technology-driven livestock management. Article [2] presents a remote monitoring and data acquisition system for predictive maintenance, along with a real-time observation strategy for DG parameters and an analysis of key metrics such as engine speed, voltage, current, power factor, coolant, fuel consumption, and battery health. The research in [3] focuses on using IoT to develop a smart greenhouse control system, integrating sensors, actuators, and a cloud platform to optimize agricultural production. Authors in [4] present a flood monitoring and warning system (FMWS) that uses an HC-SR04 ultrasonic sensor with an Arduino microcontroller to measure flood levels. Articles [5-7] feature similar architectures, utilizing IoT platforms and sensor-based monitoring systems to achieve real-time data collection and processing in various applications. However, there remains a gap in addressing cost-efficient, scalable, and simulated testing environments, which offer fault tolerance, high availability, greater control, and flexibility, as explored in this paper.

This paper presents the design and implementation of a generic IoT-based real-time environmental monitoring and alarm system. The platform has been designed to be flexible enough to be applied across multiple domains rather than tailored explicitly to any industry. The system's core architecture includes scalable messaging protocols, such as Message Queuing Telemetry Transport (MQTT) and Apache Kafka in order to ensure reliable data transmission. A microservice backend has also been incorporated to guarantee high scalability and modularity.

A manufacturing plant scenario is used as a case study to investigate the system's performance under industrial conditions. Rather than actual sensors, the system employs simulated sensors for the case study to provide a more cost-effective and flexible method for testing and validation. In this scenario, various environmental factors such as temperature, humidity, and vibration are monitored, while alarms are generated in case of threshold violations. The case study demonstrates the platform's capability to handle real-time data flows and alarms in an industrial setting while showcasing its potential for broader applications beyond the manufacturing industry. The system's ability to adapt to various use cases, combined with its real-time monitoring and alarm capabilities, makes it suitable for any environment where continuous monitoring is critical.

This study contributes to the existing literature by designing and implementing an IoT platform with real-time monitoring and alarm generation capabilities. This platform is generic in structure and can, therefore, be used across a wide range of application domains that require digital transformation.

The remainder of the paper is organized as follows: Section 2 overviews the system architecture and introduces the essential components. In section 3, a case study based on a manufacturing plant scenario is conducted to evaluate the performance of the platform developed. The paper concludes with Section 4, which provides the final remarks and a discussion.

1. The Proposed System Architecture

The architecture of the IoT-based real-time environmental monitoring and alarm system is designed to be resilient, scalable, and adaptable to various use cases. While the system is generic in nature, it is validated through a case study in a manufacturing plant. This section presents a comprehensive overview of the system architecture, focusing on fundamental operations such as data collection, real-time monitoring, alarm, messaging, and backend processing.

2.1. System Overview

The proposed architecture is built using a microservice-based backend, which allows for flexibility, scalability, and high availability. Data from various sensors is transmitted via MQTT and Apache Kafka, ensuring reliable, low-latency, and highly scalable communication between devices and the backend services. The core system processes the sensor data, stores it in a PostgreSQL database, provides a real-time user interface to monitor critical devices and environmental conditions, and manages alarms based on defined thresholds.

Figure 1 illustrates the overall structure of the proposed system and highlights the interaction between its components.

2.2. System Components

Through the aforementioned case study, the platform collects data from a variety of sensors, including:

- Temperature and Vibration Sensors on Servo Motor SM0001 that drives Conveyor Belt CB001.
- Vibration Sensors on Forklift VL0001.
- Temperature and Humidity Sensors in Storage Room R0001.

These sensors simulate industrial conditions and feed real-time data into the system, albeit using a data traffic generator service rather than a physical wireless sensor network (WSN). This approach allows for more controlled and scalable testing while reducing hardware deployment's overall cost and complexity.

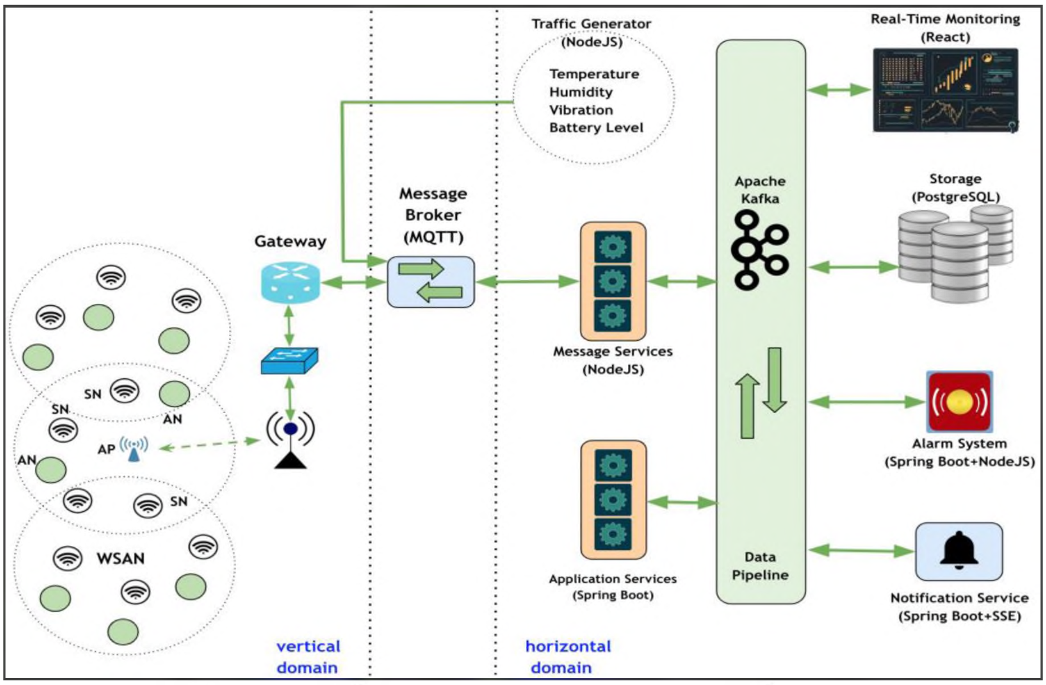

Figure 1. Outline of the proposed system architecture.

The system leverages MQTT [8] for efficient and lightweight message queuing, ensuring reliable delivery of sensor data to the cloud. MQTT is a lightweight communication protocol designed for low bandwidth and real-time messaging, making it ideal for IoT applications that require efficient data transfer between a large number of devices. Its publish-subscribe model provides scalability and reliability, both of which are critical for real-time monitoring and control in an industrial IoT environment. The platform uses the open-source Mosquitto [9] implementation of MQTT to facilitate communication between real or virtual devices and the cloud.

To support large-scale, high-throughput operations, Apache Kafka [10] is used as the core messaging system of our platform, providing a resilient and fault-tolerant infrastructure. Kafka's distributed architecture enables clustering, where topics are partitioned across multiple nodes, allowing for greater scalability. Multiple consumers in a consumer group can process data in parallel. Each group can subscribe to specific topics independently. This approach ensures efficient load balancing and parallel data processing. The robust structure allows the system to scale seamlessly as more devices are added while maintaining high availability and throughput for real-time operations. Data from the sensors is published to specific topics in the Kafka ecosystem, where backend services subscribe and process the data accordingly.

The backend architecture is based on a microservice approach, where each service is responsible for a specific task, ensuring modularity, flexibility, and ease of maintenance. The platform's main services, developed using the Spring Boot framework, include:

- API Gateway: Acts as the entry point for all incoming requests, applying access control mechanisms to filter and manage traffic. It prevents direct external access to internal services, ensuring secure communication. The API Gateway also orchestrates service requests and routes them to the appropriate microservices.
- Auth Service: Responsible for user authentication and management; it handles the generation of JWT (JSON Web Tokens) for authenticated users. This service ensures only authorized users can access the platform's functionalities, managing credentials and permissions.
- Notification Service: This service manages the incoming real-time sensor data and correspondingly sends notifications to the monitoring system. It also sends alerts generated from predefined event rules (e.g., threshold violations) to the users or related systems via push notifications or other communication channels.
- Main Processing Unit: The core service is responsible for processing and storing incoming sensor data in PostgreSQL in BSON format to handle semi-structured data efficiently. It is also responsible for defining and managing event action rules. In addition, it coordinates the interaction between services to ensure seamless data flow and system functionality.

The microservice architecture ensures that each component operates independently, facilitating scalability and fault tolerance. Services can be scaled horizontally based on demand, enabling the system to handle varying workloads seamlessly. It also provides easier maintenance and deployment flexibility, as each service can be independently developed and updated.

The platform provides a real-time monitoring interface where users can interact with the system and visualize sensor data as it arrives. An illustration of this interface is given in Figure 2. All user interfaces are developed with React and designed to be responsive, ensuring compatibility with mobile devices. Users can search for devices and sensors, view real-time graphs for parameters like temperature, humidity, and vibration, and receive alerts when thresholds are violated. The interface is designed for ease of use, with real-time updates pushed via Server-Sent Events (SSE) to ensure low latency and timely feedback.

The system includes a flexible alarm system with an integrated event definition module, as seen in Figure 3. This module allows users to set up custom action rules, enabling dynamic response to sensor data anomalies. For example, users can define an action rule that triggers an alarm or sends a push notification when the temperature exceeds a predefined threshold. This feature enhances the platform's ability to maintain real-time safety and operational efficiency.

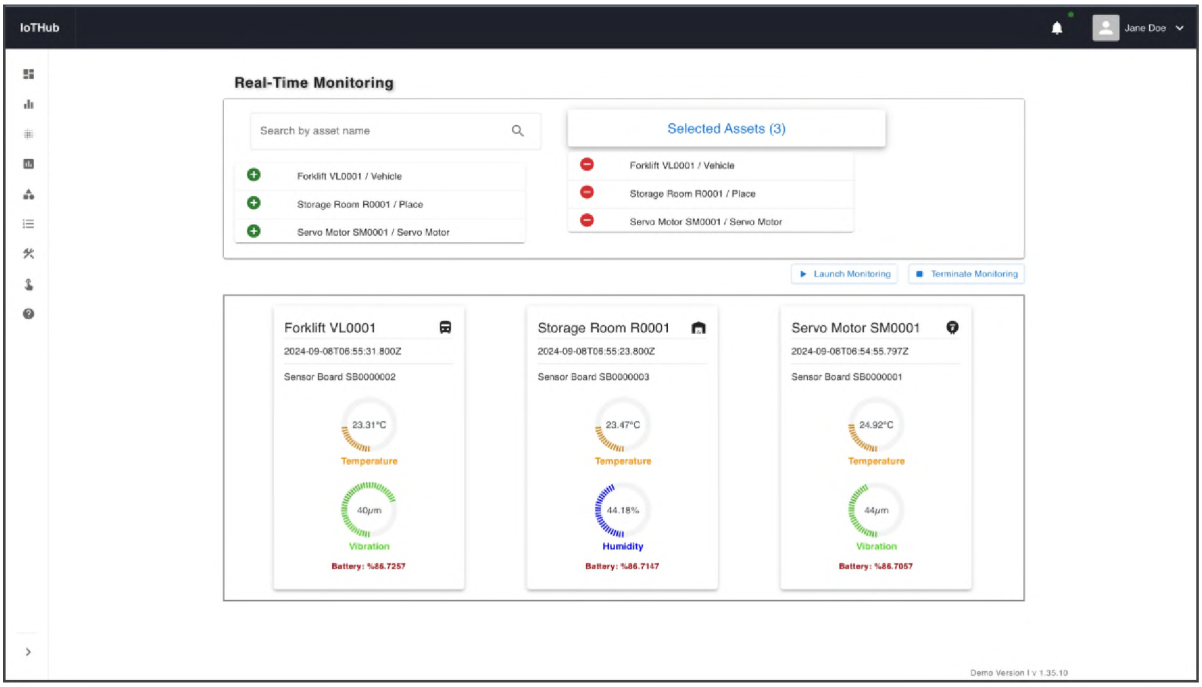

Figure 2. Real-time monitoring interface.

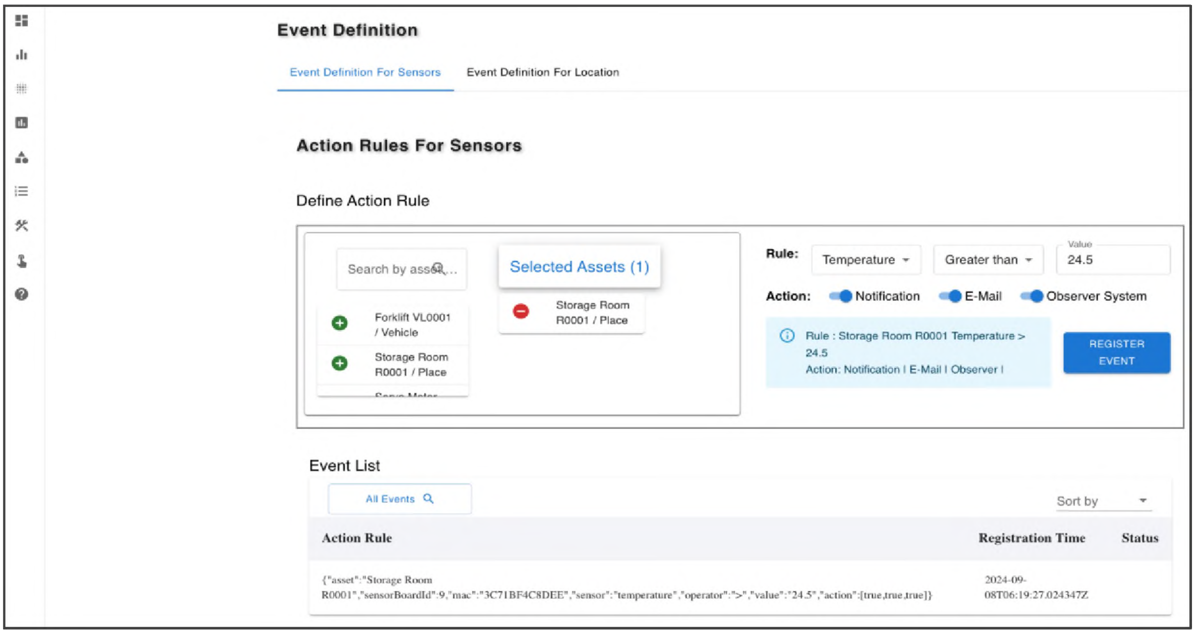

Figure 3. Event definition module of the Alarm System.

3. Experimental Results and Discussions

To validate the performance of our system, we conducted a case study based on a manufacturing plant scenario. The study involved integrating the following components into our IoT platform:

- Servo Motor SM0001, connected to Conveyor Belt CB001.
- Forklift VL0001.
- Storage Room R0001.

These components were equipped with various sensors:

- Temperature and vibration sensors for the Conveyor Belt CB001.
- Vibration sensor for the Forklift VL0001.
- Temperature and humidity sensors for the Storage Room R0001.

For this case study, sensor values were simulated using a backend service developed with NodeJS, which generates random data following a uniform distribution, adhering to specific patterns for temperature and humidity to replicate realistic conditions. These simulated sensors can be considered as virtual "things" in the IoT ecosystem, allowing the platform to interact as if it were receiving data from actual devices. This simulation enabled comprehensive testing and observation of the system's performance in a controlled environment, providing valuable insights into its functionality and reliability. The system's ability to monitor real-time data, generate alarms based on predefined thresholds, and visualize sensor values was effectively evaluated through this virtualized scenario.

The data collected from these sensors is instantly delivered to the real-time monitoring platform, which provides instant graphical representations of the sensor values belonging to the assets. Additionally, the system includes a threshold-based event definition and alarm mechanism. If sensor data exceeds predefined thresholds, the system generates alarms according to the specified action rules. This functionality is illustrated in the provided real-time monitoring screen, showcasing how alerts are triggered in response to data anomalies.

For a clear understanding of the working sequence of the real-time monitoring and alarm generation processes described below, please refer to Figure 1. The real-time monitoring interface subscribes to the notification service, allowing it to instantly receive state updates, such as sensor values and alarm signals. Traffic generators produce these sensor values and push them into the MQTT system. Related message services retrieve this data, transform it properly, and forward to the Apache Kafka message distribution system.

One copy of this data is transferred to the Main Processing Unit service for persistent storage, while the other is conveyed to the Notification Service. The Notification Service then delivers the data to the real-time monitoring interface via server-sent events (SSE). As illustrated in Figure 2, the incoming sensor values for all assets in the case study are displayed in the interface.

As soon as a sensor value reaches MQTT, a message service responsible for evaluating alarm rules (e.g., an action rule like "temperature > 24.5" as defined in Figure 3) assesses the value and executes the required action. As shown in Figure 4, when the temperature of the storage room exceeds the defined threshold, an alert is sent to the real-time monitoring system for immediate attention. The alarm system can also initiate further actions, such as sending an email notification, triggering an air conditioning system, or logging the event in the audit system, depending on the predefined action rules in the system.

Integrating an alarm system into IoT-based monitoring platforms is crucial for critical environments where timely intervention can prevent operational failures, safety hazards, or equipment damage. In industrial scenarios, such as manufacturing plants, real-time monitoring of temperature, humidity, and vibration is essential to maintaining equipment health and ensuring optimal working conditions. For instance, in the case of a malfunctioning cooling system in a storage facility, an undetected temperature rise could lead to spoiled goods or damage to sensitive equipment. With an integrated alarm system, immediate actions can be taken as soon as the temperature exceeds a predefined threshold, such as triggering cooling systems, alerting maintenance teams, or halting operations to prevent further damage.

By enabling automated responses and sending notifications to key personnel, this system minimizes response times and mitigates the risk of human error. The flexibility to customize action rules, such as setting thresholds and defining corresponding actions, allows organizations to tailor the system to their specific operational needs. It is highly beneficial for ensuring system resilience and operational continuity in mission-critical environments.

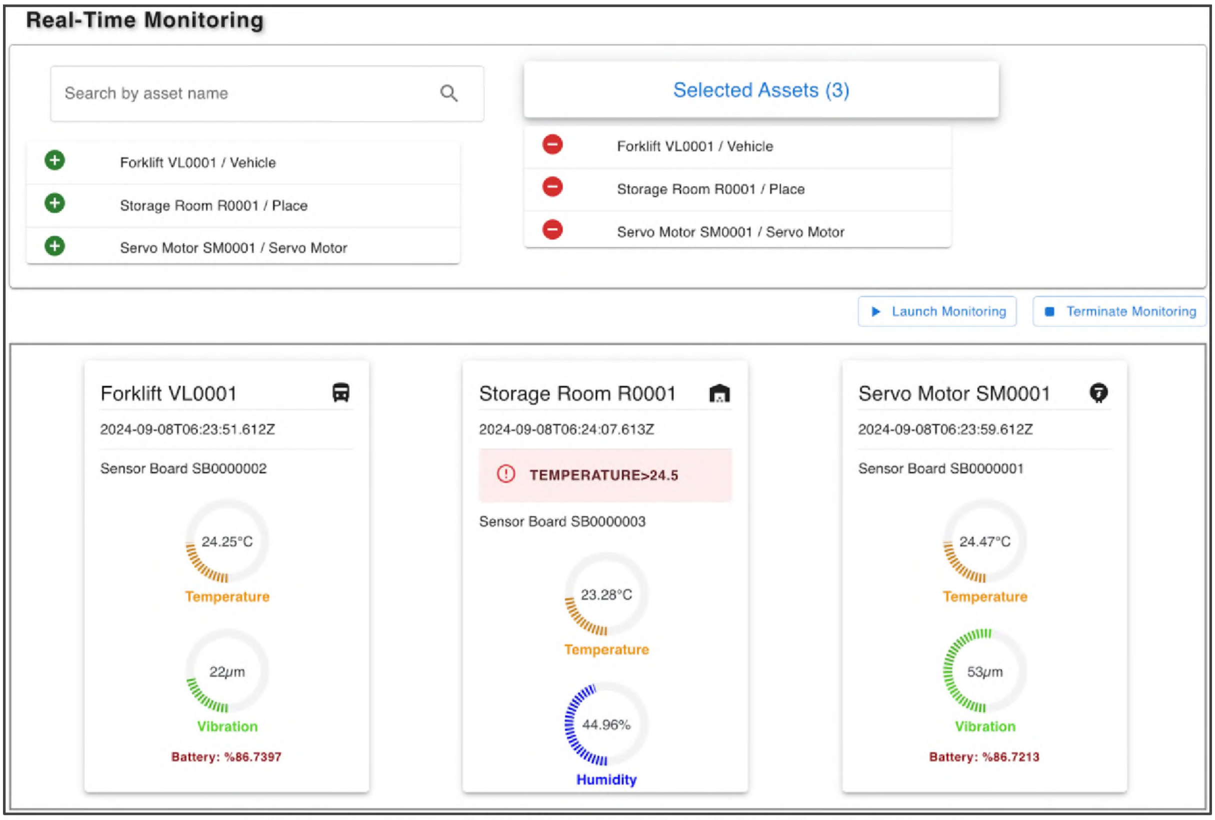

Figure 4. Event definition module of the Alarm System.

In the platform, all incoming sensor values are stored in BSON format within a PostgreSQL database to facilitate further analysis and long-term data retention. Figure 5 illustrates the temperature and vibration values captured by sensors placed in the storage room over a specific period. The traffic generator is configured to produce sensor readings for each asset every 4000ms. As shown in the figure, the sensor values recorded between 04 PM and 01 AM are represented as a line graph, providing a continuous and detailed visualization of environmental conditions in real time.

Analyzing this type of sensor data is vital for organizations, particularly in manufacturing plants, where operational efficiency and safety rely on monitoring environmental factors like temperature and vibration. By collecting and storing these values over time, organizations can identify trends, detect anomalies, and anticipate potential equipment failures before they occur. Such insights enable proactive decision-making, reducing downtime and improving overall productivity. In addition, advanced analytics performed on this data can optimize processes, enhance equipment performance, and ensure regulatory compliance, ultimately contributing to the long-term sustainability and profitability of the organization.

Figure 5. Event definition module of the Alarm System.

4. Conclusions

As demonstrated in this study, implementing a real-time environmental monitoring and alerting system provides significant benefits to industrial operations by ensuring continuous monitoring of critical environmental parameters. The system's scalable and fault-tolerance architecture makes it adaptable to various application domains. Its scalable nature ensures the system can handle increasingly connected devices and data streams without compromising performance or reliability. This is a critical feature for organizations or facilities with expanding operational needs.

The platform is also designed with future enhancements in mind. By integrating a robust data streaming system, the architecture could be prepared for further development, including the addition of advanced data analytics modules. Such a module could analyze incoming data in real time or over time to provide predictive insights. This capability would enhance the system's ability to monitor conditions and provide actionable intelligence, making it a valuable tool for industries looking to leverage IoT for operational excellence and strategic decision-making.

Acknowledgment

The commercial implementation of the IoTHub platform represented in this article has been carried out by Orka İnternet Teknolojileri Tic. Ltd. Şti. (Orca Internet Technologies Trading Limited Company). Special thanks to Sakarya University in Türkiye and Kozybayev University in Kazakhstan for their invaluable support and contributions to this research.

References:

1. Harini Shree Bhaskaran, Miriam Gordon, Suresh Neethirajan, "Development of a cloud-based IoT system for livestock health monitoring using AWS and python", Smart Agricultural Technology, Volume 9, 2024, 100524, ISSN 2772-3755.

2. Ambarish Gajendra Mohapatra, Anita Mohanty, Nihar Ranjan Pradhan, Sachi Nandan Mohanty, Deepak Gupta, Meshal Alharbi, Ahmed Alkhayyat, Ashish Khanna, An Industry 4.0 implementation of a condition monitoring system and IoT-enabled predictive maintenance scheme for diesel generators, Alexandria Engineering Journal, Volume 76, 2023, Pages 525-541, ISSN 1110-0168.
3. Yongchao Song, Jiping Bi, Xuan Wang, Design and implementation of intelligent monitoring system for agricultural environment in IoT, Internet of Things, Volume 25, 2024, 101029, ISSN 2542-6605.
4. Muhammad Izzat Zakaria, Waheb A. Jabbar, Noorazliza Sulaiman, Development of a smart sensing unit for LoRaWAN-based IoT flood monitoring and warning system in catchment areas, Internet of Things and Cyber-Physical Systems, Volume 3, 2023, Pages 249-261, ISSN 2667-3452.
5. Alessandro Zivelonghi, Alessandro Giuseppi, Smart Healthy Schools: An IoT-enabled concept for multi-room dynamic air quality control, Internet of Things and Cyber-Physical Systems, Volume 4, 2024, Pages 24-31, ISSN 2667-3452.
6. Sangeethalakshmi K., Preethi Angel S., Preethi U., Pavithra S., Shanmuga Priya V., Patient health monitoring system using IoT, Materials Today: Proceedings, Volume 80, Part 3, 2023, Pages 2228-2231, ISSN 2214-7853.
7. Celal Çeken, Mohammed Al-Hubaishi "Integrating SDN-Enabled Wireless Sensor Networks Into the Internet", The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Metz, France, 2019, pp. 1090-1094.
8. R. Coppen, A. Banks, E. Briggs, K. Borgendale, R. Gupta," MQTT Version 5.0" Standards Track Work Product, 2019.
9. R. Light," Mosquitto man page." [Online]. Available: https://mosquitto.org/man/mosquitto-8.html
10. Apache Software Foundation: A distributed streaming platform. [Online]. Available: http://kafka.apache.org

Information about the author:

Celal Ceken – corresponding author, PhD, Professor, Department of Computer Engineering, Faculty of Computer and Information Sciencesc hair, Sakarya, Türkiye; e-mail: celalceken@sakarya.edu.tr.