*Journal of the Brazilian Computer Society, 2026, 32:1,* doi: 10.5753/jbcs.2026.5471


# Leveraging Fog and Cloud Computing for Continuous Health Monitoring and Data Processing: An Architecture for Outdoor Environments and Variable Connectivity

**Juan Felipe Souza Oliveira** [ **Instituto Militar de Engenharia** | ***oliveira.juan@ime.eb.br*** ]
**Paulo Cesar Salgado Vidal** [ **Instituto Militar de Engenharia** | ***vidal@ime.eb.br*** ]
**Ronaldo Moreira Salles** [ **CIICESI, ESTG,Instituto Politécnico do Porto** | ***rmo@estg.ipp.pt*** ]
**Marcelo Quesado Filgueiras** [ **Universidade Federal de Juiz de Fora**| ***mquesado@uol.com.br*** ]

*Instituto Militar de Engenharia, Praça Gen. Tibúrcio, 80 - Urca, Rio de Janeiro - RJ, 22290-270, Brazil.*

**Received:** 23 January 2025 • **Accepted:** 11 December 2025 • **Published:** 28 April 2026

**Abstract.** A multilayer architecture was developed for real-time health data collection and processing, designed for outdoor environments with high population density and significant network interference. By integrating fog and cloud computing, the system addresses the growing demand for continuous health monitoring driven by the proliferation of Internet of Things (IoT) devices and Wireless Body Area Networks (WBANs) using smartbands. Traditional cloud-centric solutions often face challenges such as high latency and data integrity issues in unstable network conditions. The proposed architecture overcomes these limitations by employing fog computing for edge data preprocessing, reducing reliance on cloud connectivity and enhancing system responsiveness. The architecture was originally evaluated under diverse network conditions (3G, 4G, 5G) and in real-world scenarios such as football stadiums, metro systems, and urban beaches, demonstrating over 96% packet delivery success and significant latency reductions compared to cloud-only approaches. In this extended version, additional real-world scenarios are analyzed, including domestic flights, large-scale events in stadiums with over 60,000 attendees, and new evaluations along urban beachfronts. Furthermore, this version provides a more detailed explanation of key mechanisms, such as the use of the Transactional Outbox pattern to ensure data consistency in unstable networks and the integration of distributed processing techniques for real-time alert generation. These contributions offer deeper insights into the architecture's scalability and reliability, confirming its effectiveness in maintaining data integrity and achieving low latency in connectivity-challenged environments, providing a solution for health monitoring.

**Keywords:** Fog Computing, Cloud Computing, E-Health

## 1 Introduction

The exponential growth of healthcare systems and the increasing life expectancy of the global population bring various challenges, especially concerning the management and control of chronic and epidemic diseases that threaten public health (Do Nascimento *et al*., 2020). Continuously monitoring patients' health parameters, such as heart rate, blood oxygen levels, blood sugar index, and weight, is essential to address these issues. However, this monitoring requires high integrity and reliability of the data transmitted through mobile networks and the internet (Farahani *et al*., 2018).

The Internet of Things (IoT) has emerged as a promising solution to these challenges by integrating sensors, wearable devices, databases, and cloud computing (Tardieu *et al*., 2020). Wireless wearable devices, such as smartbands, enable the monitoring of patients outside healthcare facilities, promoting savings in human and material resources (Manyika *et al*., 2015), as well as enhancing the comprehensive monitoring of patients who need rapid diagnosis of their conditions, particularly those with high disability and mortality rates (Mendonça *et al*., 2022; Ahmed Kamal *et al*., 2023).

Cloud computing offers advantages such as scalability, data processing capacity, and operational cost savings. It allows dynamic allocation of computational resources according to demand, optimizing server and infrastructure usage (Fernando *et al*., 2013). Moreover, it facilitates the implementation of high-availability database solutions, enabling real-time collaboration and mobility, making it easier to access data and applications from anywhere at any time (Stavrinides and Karatza, 2019).

However, it faces some limitations, such as the dependence on a stable internet connection, increased latency due to the distance between the user's device and the cloud data centers (Kashani *et al*., 2021), and potential security vulnerabilities during data transmission (Tina Victoria and Kowsigan, 2022).

To mitigate these limitations, fog computing emerges as an evolution of distributed computing, aiming to overcome the constraints of current cloud-based models (Kashani *et al*., 2021; Angel *et al*., 2022). Data can be processed and storage and communication resources can be obtained at the network edge, closer to the sensors and users. This enables continuous and real-time monitoring, triggers alerts with lower latency, even in the event of network failures (Vilela *et al*., 2018), and pre-processes data before being sent to the cloud layer (Angel *et al*., 2022). This decentralized processing capability is particularly advantageous for healthcare applications requiring fast and reliable responses (Ilyas et al., 2022).

Some works have implemented fog and cloud computing for health monitoring, however, few (Al-Joboury and Al-

Hemiary, 2018; Ahmadi *et al*., 2021) have explored testing and evaluating the integrity of collected data and latency in outdoor environments with high population density and interference, such as beaches, football stadiums, and metro systems. These environments present additional challenges due to the high number of connected devices and network instability.

The authors published the paper at the 30th Brazilian Symposium on Multimedia and the Web (Oliveira *et al*., 2024), proposing a vital sign monitoring architecture using fog computing, called CEN. The architecture consists of three layers (sensors, fog, and cloud) and employs the Message Queuing Telemetry Transport (MQTT) communication protocol. The MQTT protocol is characterized by being lightweight (in terms of processing) and efficient, making it ideal for real-time data transmission in networks with limited and unstable resources (Sasaki and Yokotani, 2019), and was evaluated in outdoor environments.

In this extended version, we expand on the work described in (Oliveira *et al*., 2024) by detailing the architecture implementation, including a broader analysis of data storage in NoSQL databases. These databases were chosen for their flexibility and ability to handle large volumes of data generated continuously by IoT devices, such as heart rate, geolocation, and network status at the time of data collection. The data structure was designed to ensure consistency and integrity even under unstable connectivity conditions, highlighting its suitability for dynamic and high-interference environments.

This paper further explores Quality of Service (QoS) in medical applications, which plays a key role in transmitting sensitive data such as vital signs in real-time. QoS metrics, such as latency, packet loss, and bandwidth, are particularly critical to ensure the reliability and efficiency of continuous monitoring and clinical response in healthcare systems. The architecture was designed to meet these requirements under adverse network conditions, ensuring that the transmitted data maintain high integrity and low latency, even in challenging scenarios such as densely populated environments or unstable connectivity.

A detailed explanation of the developed mobile application is provided, which directly integrates with wearable devices, such as smartbands, and manages the collection and transmission of data to the fog and cloud computing layers. The application was designed to operate in various mobile networks, utilizing local preprocessing strategies and efficient transmission to optimize resource usage and meet QoS requirements.

This version also explores in greater depth the use of the Transactional Outbox design pattern, implemented to ensure consistent data delivery between the architecture layers. Strategies for generating real-time connection and disconnection alerts to the cloud, critical for medical applications, are discussed, highlighting how the system was adjusted to optimize reliability and responsiveness.

Another unique aspect of this version is the inclusion of three new evaluation scenarios, complementing the tests conducted earlier. New evaluations were conducted on domestic flights, in urban beach areas, and at large-scale events in stadiums with more than 60,000 participants. These scenarios validate the architecture under diverse and challenging outdoor conditions, broadening its applicability to real-world contexts. The integration of QoS in these scenarios demonstrates the potential of the architecture to meet the monitoring demands even under adverse conditions.

The paper is organized as follows: Section 2 discusses the technical and operational challenges of cloud and fog computing in healthcare, as well as Quality of Service (QoS) in medical applications. Section 3 reviews related works and compares them with the proposed approach. Section 4 provides a detailed description of the developed architecture and its characteristics. Section 5 addresses the evaluation of implementations and test scenarios. Section 6 presents the results of the experiments, including performance in outdoor scenarios, latency tests, and delays in alert generation. Finally, Section 7 concludes the paper and suggests directions for future work.

# 2 Background

## 2.1 Cloud Computing in Healthcare

Applications in the healthcare field developed and based on cloud computing possess high storage capacity and computational power, making them well-known as efficient ways to transfer, process, and store data (Sharma and Gupta, 2023). However, there are some deficiencies in current models (Angel *et al*., 2022), such as:

- **Network Infrastructure**: The lack of connectivity to servers, data, or applications hosted in the cloud due to failures in communication networks, such as mobile networks, represents a major obstacle. In medical contexts, where continuous access to information is essential for quick decisions, network disruptions can cause harmful delays in patient care.
- **Bandwidth**: The large volume of data traffic can lead cloud providers to apply bandwidth restrictions to ensure the network functions properly. This can negatively impact the performance of critical healthcare applications, such as monitoring systems that require real-time transfer of large data volumes. With bandwidth restrictions, response times can be significantly increased, making it difficult to access the necessary information quickly and effectively for immediate decisions.
- **Reliability**: Issues in the cloud can result in various implications. Data integrity is vital to ensure the accuracy of information and the continuity of services. In the event of failure, data loss or corruption can occur, directly affecting the consistency of medical records and other essential documentation. This can result in incorrect diagnoses or delays in treatment, compromising patient safety and well-being.
- **Response Time**: Medical applications, such as patient monitoring systems, require extremely fast response times to alert healthcare professionals about critical changes in the patient's condition. Latency in communication with the cloud can cause delays in detecting and responding to these changes, potentially compromising healthcare professionals' ability to intervene promptly and effectively.

Considering the aspects observed, IoT models using only the cloud computing layer do not necessarily offer the most viable solution for critical applications (Al-Joboury and Al-Hemiary, 2018) and addressing the IoT challenges related to the healthcare field (Sharma and Gupta, 2023). In this context, fog computing emerges as a distributed computing paradigm that extends the cloud's capabilities to devices located near users or at the network edge, such as sensors, cameras, and wireless body area networks.

## 2.2 Fog Computing in Healthcare

The integration of fog computing technologies with healthcare devices, such as smartbands and smartphones, presents technical and operational challenges that must be overcome to ensure the effectiveness and safety of health monitoring systems (Tina Victoria and Kowsigan, 2022). Fog computing offers significant advantages, such as decentralized processing and improved real-time response capability, but it also presents challenges for successful implementation (Angel *et al*., 2022; Ahmadi *et al*., 2021; Tina Victoria and Kowsigan, 2022), such as:

- **Health Data Security and Privacy**: Protecting health data is critical due to its sensitivity. Communication between the smartband, smartphone (fog node), and the cloud involves multiple steps susceptible to vulnerabilities, requiring end-to-end encryption and strong authentication to ensure data integrity.
- **Connectivity Management and Interference**: Densely populated environments, such as beaches, stadiums, and metro systems, present challenges due to the high density of connected devices, as well as terrain characteristics that lead to network congestion and interference. Using a lightweight communication protocol like MQTT can mitigate these problems.
- **Implementation and Integration Complexity**: Integrating smartbands, smartphones, and cloud computing into an architecture involving fog computing requires coordinating multiple heterogeneous components. Interoperability between devices and platforms, such as Android and Apple, is essential, as well as the development of specialized applications and plugins to manage communication between devices and the cloud computing layer.

In summary, the integration of healthcare devices with fog and cloud computing faces critical challenges: data security and privacy, connectivity management in external environments with multiple connected devices, and the complexity of integrating heterogeneous systems. Overcoming these challenges is essential for a multi-layer architecture, combining sensors, fog layer, and cloud.

## 2.3 Quality of Service (QoS) in Medical Applications

Quality of Service (QoS) is a set of requirements that ensures the optimal performance of networks for specific applications. In medical applications, QoS is particularly critical (Mukhopadhyay, 2017; Sodhro *et al*., 2019), as it involves the transmission of sensitive data, such as vital signs, medical images, and real-time videos, used in continuous monitoring, diagnostics, consultations, and clinical procedures (Rodrigues *et al*., 2022). The reliability and efficiency of communication directly impact the safety and effectiveness of remote healthcare services.

Medical applications have strict requirements for various QoS metrics, depending on the type and criticality of the application:

- **Latency:** The time required for data to be transmitted from the source to the destination. For applications such as videoconferencing or telesurgery, latency must be below 300 ms; for monitoring vital parameters, values below 500 ms are acceptable (Rodrigues *et al*., 2022).
- **Packet Loss:** In the context of networks, a packet is a unit of data transmitted between devices, containing information for delivery and content. During transmission, data is divided into smaller packets, which can be lost due to issues such as congestion or interference. Packet loss can cause gaps in the data, compromising the integrity and analysis of the received information.
- **Bandwidth:** The network's capacity to support the volume of transmitted data is essential for real-time monitoring applications. Sensory data, such as continuously collected vital signs, require sufficient bandwidth to ensure stable and uninterrupted transmission, especially in scenarios with multiple devices connected simultaneously.

### 2.3.1 Mobile Networks and QoS Requirements in Medical Applications

The impact of QoS can be illustrated in medical procedures such as the remote monitoring of patients with chronic conditions. For instance, in patients with heart failure, continuous monitoring of parameters like heart rate significantly contributes to detecting early complications (Sodhro *et al*., 2019). These data are collected by wearable devices, such as smartbands, and transmitted for remote analysis.

The monitoring flow generally follows these steps:

- **Transmission to the Cloud:** Normalized data are sent to a central server, where they are analyzed and integrated into the patient's electronic medical record.
- **Clinical Response:** Alerts regarding detected anomalies are sent to healthcare professionals, who can make immediate decisions, such as adjusting medication or calling the patient in for further evaluation.

### 2.3.2 Network Context in Medical Applications

Network conditions have a direct impact on meeting QoS requirements. Mobile networks such as 3G, 4G, and 5G offer varying performance, as follows (Peralta-Ochoa *et al*., 2023):

- **3G:** Highly susceptible to high latencies (>600 ms) and packet loss, making it unsuitable for real-time applications.

- **4G:** Offers significant improvements in latency (150-450 ms) but can be affected by congestion in densely populated areas.
- **5G:** Promises ultra-low latency (<100 ms) and higher bandwidth, making it ideal for critical applications like telesurgery and high-definition video transmission.

Integrating QoS in medical scenarios allows procedures to be performed with greater efficiency and safety, even under adverse network conditions. For example, in environments such as a crowded stadium or a busy beach, the system's ability to meet QoS requirements can determine the success or failure of a medical application.

The related works and results of this study, presented in the following sections, demonstrate how the proposed architecture integrates fog and cloud computing to optimize QoS levels in networks prone to instabilities. This approach aims to reduce latency and packet loss in real-world tests while ensuring data integrity, such as in public transportation monitoring or areas with large crowds.

# 3 Related Works

The related works in this research area have explored the concepts of latency, fault tolerance (Gomes *et al*., 2020), simulation of data collection, and real data collection in health parameter monitoring systems (Mendonça *et al*., 2022). These studies investigated approaches to minimize latency in data transmission and processing, as well as proposing techniques to deal with failures and instabilities in mobile communication networks (Alshammari, 2023; Bansal and Aggarwal, 2022). Furthermore, they utilized simulations to evaluate system performance in different scenarios and also conducted real data collection to validate their proposals (Shaji *et al*., 2023).

The review of related works covered relevant criteria aligned with the architecture developed in this paper, such as:

- **Latency**: Ensuring that the data transmission and processing time is minimized to provide fast and efficient responses. This requires the optimization of latency both in fog computing and in the cloud, so that the system's agility is maintained.
- **Fault Tolerance**: Developing architectures that can withstand failures in devices, networks, or processes, while maintaining system operation without interruptions. This ensures service continuity and trust, even under adverse conditions.
- **Smartphone as Fog Node**: Smartphones play a key role in the fog computing architecture, acting as peripheral nodes in the network. With their high processing power and storage capabilities, they can perform computational tasks locally, reducing latency and efficiently connecting to mobile networks, Wi-Fi, and other nearby devices.
- **Different Network Communication Protocols**: Application of protocols such as HTTP, MQTT, and CoAP in the healthcare context, exploring different approaches. These protocols support the efficient exchange of clinical data, remote patient monitoring, and integration with wireless body area networks, promoting more effective communication between devices.
- **Data Collection**: Validating and calibrating monitoring systems, enabling verification of system reliability and effectiveness, and optimizing the system as a whole based on real usage conditions.

The comparative studies addressed the challenges faced in indoor and outdoor environments, considering the presence or absence of mobile communication networks such as 2G, 3G, 4G, and 5G, and the use of Wi-Fi (Sharma and Gupta, 2023; Ilyas *et al*., 2022). Indoor environments were identified as offering more stable network infrastructure, facilitating communication between health monitoring devices. In contrast, outdoor environments present greater variability and connectivity challenges.

Solutions have been proposed to handle network outages and instabilities in outdoor environments (Shaji *et al*., 2023), including temporary data storage, local storage technologies, and ad hoc networks (Dar *et al*., 2018). These strategies aim to ensure the reliability of data collection in challenging environments. In summary, the studies highlighted the importance of considering the characteristics of each environment, the presence or absence of mobile networks and Wi-Fi, and the different generations of mobile networks when designing health monitoring systems in various contexts.

The technologies and computational tools used in previous works (Sasaki and Yokotani, 2019; Shaji *et al*., 2023), such as NoSQL databases, messaging systems, and the software architecture design pattern *Transactional Outbox*, contributed to the deepening and implementation of the developed architecture. These technical contributions provide significant improvements, enabling more efficient, reliable, and scalable data collection.

NoSQL databases were chosen for this architecture due to their intrinsic ability to meet the demands of real-time health monitoring systems, which must process large volumes of heterogeneous and semi-structured data from multiple sensors. Unlike traditional relational databases, NoSQL systems offer a flexible schema design, allowing different types of health data — such as heart rate, geolocation, and network state — to be stored without requiring a rigid, predefined structure (Asri and Jarir, 2023). This is particularly important for mobile sensing applications, where the format and frequency of sensor outputs may vary.

Additionally, NoSQL databases provide high ingestion throughput and low-latency writes, enabling rapid recording of patient data in real-time (Shaji *et al*., 2023). Their horizontal scalability allows the backend to dynamically expand its storage and processing capacity as the number of users or devices increases, without requiring significant reconfiguration (Al-Sakran *et al*., 2018). These characteristics make NoSQL databases particularly well-suited to the demands of fog and cloud-based health architectures, supporting both scalability and adaptability in heterogeneous environments.

Regarding messaging systems and distributed data streaming tools to handle continuous real-time data flows, asynchronous communication is used between system components, making data exchange more flexible and scalable (Hiraman *et al*., 2018), playing a key role in monitoring in outdoor environments where network conditions may be unstable. Additionally, the *Transactional Outbox* pattern ensures con-

sistency and atomicity in message sending operations (Sharma and Gupta, 2023; Shaji *et al*., 2023), avoiding data loss or duplication during asynchronous communication.

Finally, for the data traffic to the cloud layer, recent studies show that the *Message Queuing Telemetry Transport (MQTT)* communication protocol exhibits high efficiency and fault tolerance (Al-Joboury and Al-Hemiary, 2018; Alshammari, 2023), enabling automatic reconnection when a disruption or change in connectivity occurs. It can adapt to different levels of network quality, including low bandwidth and unstable connections, such as 2G and 3G, or situations of network overload due to multiple devices being connected simultaneously, optimizing data traffic based on network conditions (Sharma and Gupta, 2023; Ilyas *et al*., 2022).

Despite significant advances and valuable contributions, some limitations were identified. Many studies did not fully explore the integration of different communication technologies, such as Wi-Fi and mobile networks across generations (2G, 3G, 4G, and 5G), creating potential gaps in continuous monitoring under variable connectivity. Although some works proposed fault tolerance and temporary data storage mechanisms, their implementation in real-world scenarios still requires extensive testing and validation. Most approaches focused on indoor environments with stable networks, leaving a notable gap in studies addressing outdoor settings with adverse connectivity conditions.

Another observed limitation is the underutilization of smartphones' computational power as fog nodes. Many studies have not fully explored the potential of these devices to perform computational tasks at the network's edge, which could reduce latency and improve system efficiency. These limitations highlight the need for more integrated approaches and solutions that can be applied in a variety of health monitoring scenarios.

This paper differs by presenting an architecture that will be tested in outdoor environments with adverse connectivity conditions, such as beaches, football stadiums, and public transportation such as metros, providing a more comprehensive and practical evaluation.

The developed architecture utilizes the computational power of smartphones as fog nodes, where preprocessing, encryption, and data anonymization are performed to comply with the General Data Protection Law (LGPD, Portuguese: Lei Geral de Proteção de Dados).

Finally, the application of NoSQL databases and the use of the MQTT protocol for continuous real-time data flow present a perspective of overcoming scalability and efficiency challenges in data transmission. These approaches aim to improve efficiency, reduce latency, and increase system resilience, representing a significant contribution to the field of health monitoring in outdoor environments.

### 3.1 Related Research on IoT and Healthcare Applications - JBCS

Recent publications in the *Journal of the Brazilian Computer Society (JBCS)* highlight advancements in the use of heterogeneous networks and IoT technologies for medical applications, focusing on the integration of connected devices and the efficiency of distributed systems. (Haertel *et al*., 2022) proposes the use of middleware for distributed processing of contextual data, addressing challenges related to connectivity variability in mobile networks and the need for real-time decision making. This work explores the use of sensors and mobile devices in distributed systems, emphasizing the importance of resilient architectures to meet specific QoS demands.

Another relevant study, presented in (da Silva *et al*., 2022), discusses the development of solutions based on the Internet of Human Things (IoHT), with an emphasis on the interaction between devices, local infrastructure, and social dimensions. The study adopts an iterative design approach to adapt the systems to healthcare environments, assessing reliability and network performance. The analysis of these scenarios highlights the value of distributed architectures capable of operating under diverse conditions and meeting the needs of critical medical applications.

These works highlight a growing trend in IoT and heterogeneous network research aimed at healthcare applications, particularly in the context of resilient and scalable systems. The research presented in this paper expands on this approach by implementing an architecture that uses mobile devices as nodes in the fog layer (*fog nodes*), integrating fog computing and mobile networks from different generations (3G, 4G, and 5G). Tests in challenging scenarios such as beaches, stadiums, public transportation, and airplanes aim to validate the architecture under real-world conditions, contributing to the advancement of the state-of-the-art in remote monitoring.

## 4 Developed Architecture

### 4.1 Architecture Design

The development and implementation stages of the architecture are integrated with the fundamental principles of Software Engineering and Software Quality assurance. Each phase, from the literature review to real-world testing, demonstrates a methodological and technical approach to ensuring the complete development of the architecture. The process includes: a) literature review, b) definition of functional and non-functional requirements, c) selection of technologies and frameworks, d) preliminary tests on network variations in external environments, e) architecture development, f) architecture testing in external environments, g) performance evaluation of the architecture.

The health monitoring architecture includes three main layers: sensors, fog computing, and cloud computing. Sensors, such as smartbands, collect real-time vital data. Fog computing, using smartphones as fog nodes, performs preprocessing, encryption, anonymization, and persistence of data in case of connectivity instability, and also transmits data to the cloud via MQTT. Cloud computing is responsible for storage, analysis, generation of tracking dashboards, and issuing alerts, enabling real-time monitoring.

From the Bluetooth connection between the smartband and the patient's smartphone, continuous collection of health parameters will be possible, providing accurate and up-to-date information. This architecture enables more effective and personalized monitoring.

By processing data close to the source through fog com-

puting, the architecture can ensure more efficient and agile data collection and transmission. This can solve challenges related to generating reliable electronic data, meeting time requirements, and fault tolerance, collecting health parameters such as heart rate, step count, and patient geolocation.

The collected and processed data can be used to obtain a clear view of the patients' health condition and will be periodically sent to cloud servers. In the context of e-Health, ambulances, nurses, and doctors will be able to remotely access this information to assess the condition of the monitored patients.

## 4.2 Fog Computing Layer

Figure 1 presents the component and tool diagram of the fog computing architecture, including the sensor layer and the mobile application (Publisher API) developed for Android and Apple, which communicates with the cloud. Using the Bluetooth connection between the smartband and the patient's smartphone, health parameters such as heart rate, step count, and geolocation are continuously collected.

In the implementation of this layer, the smartphone acts as a fog node, performing various critical functions. First, it performs the preprocessing of the collected data. In case of instability in the connection with the cloud, the smartphone uses a local database to temporarily persist the data. This ensures that no information is lost during connectivity interruptions.

The processed data is sent to the cloud via the MQTT protocol, which ensures efficient and resilient delivery of the information. In the fog layer, in addition to receiving data from the sensors, an ETL (Extract, Transform, Load) process takes place, which includes extracting raw data, applying filters, and storing it locally. This process allows for the selection and processing of only the relevant data, reducing the amount of information to be sent to the cloud and optimizing resource consumption.

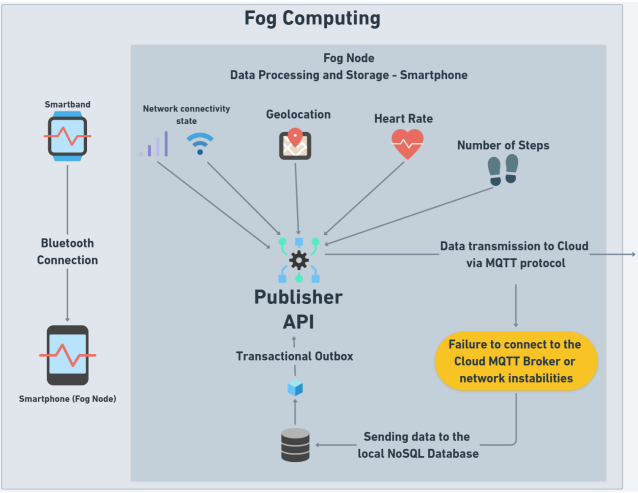


**Figure 1.** Component diagram of the fog computing layer.

### 4.2.1 Transactional Outbox Design Pattern

In addition to the QoS levels implemented by the MQTT protocol, which provide fault tolerance, the developed architecture uses an additional design pattern, the Transactional Outbox, to enhance resilience in the data transmission between the fog layer and cloud resources. This approach is particularly effective in scenarios where cloud services, such as the MQTT Message Broker, may be temporarily offline or experiencing technical failures.

In the context of the fog layer, the Transactional Outbox pattern is implemented on the smartphone device, which acts as an intermediary node between the IoT sensors and the cloud layer. When the device collects data, such as heart rate or geolocation, it initially persists this information in a local database, in the table named outbox. This local database is used to ensure that the data is securely stored, regardless of the availability of the MQTT Message Broker in the cloud.

The flow works as follows:

1. The device attempts to send the data directly to the MQTT broker in the cloud layer.
2. If the broker is unavailable, the data remains stored in the local database of the smartphone in the outbox table, awaiting a reconnection.
3. A background process periodically performs polling on the outbox table to check for pending messages.
4. When the connection to the broker is re-established, the messages are successfully sent and removed from the outbox table. In case of another failure, the message remains in the local database until the delivery is completed.

This design pattern provides resilience to the architecture, ensuring that information is not lost even under adverse connectivity conditions. Additionally, the system performs checks on the data quality before transmission, assessing, for example, whether the sensor is inactive, discharged, or if there were communication failures. These checks allow for a more accurate adaptation to the variability of the operational context and ensure that only valid and complete information is sent to the broker in the cloud layer.

The integration of the Transactional Outbox into the fog layer strengthens the resilience mechanisms and data reliability, allowing the system to maintain data integrity and the continuity of monitoring.

The described flow is illustrated in Figure 2, demonstrating how messages are collected, stored locally, and eventually successfully delivered to the broker.

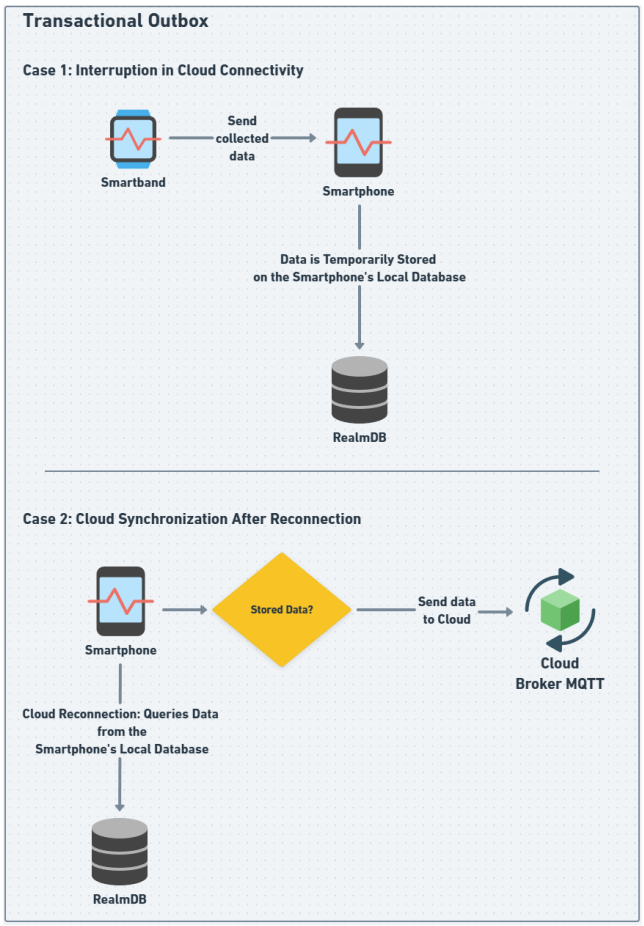


**Figure 2.** Event flow with the transaction outbox pattern.

### 4.2.2 Fog Health IME App Implementation

A specific mobile application was developed using the Kotlin language, integrating with the smartband and the MQTT Broker in the cloud for data transmission via the MQTT protocol, and with the server in the cloud for transmission via HTTP. The application was designed using smartphone system resources and open-source libraries, such as Eclipse Paho MQTT for efficient communication with the MQTT broker, Retrofit for integration with HTTP requests to the cloud, and the Transactional Outbox Pattern to ensure data consistency and reliable delivery even in unstable network conditions.

The Bluetooth Low Energy (BLE) library was also used to establish communication between the app and the smartband. BLE was chosen due to its low power consumption, making it ideal for wearable devices that require long data collection sessions without compromising battery life. This technology enables efficient and stable connection between the smartphone and the smartband, ensuring continuous data transfer with minimal impact on energy consumption.

The app interface is simple and straightforward, facilitating the process of connection, transmission, and monitoring for the user. Figure 3 shows the basic usage flow of the app:

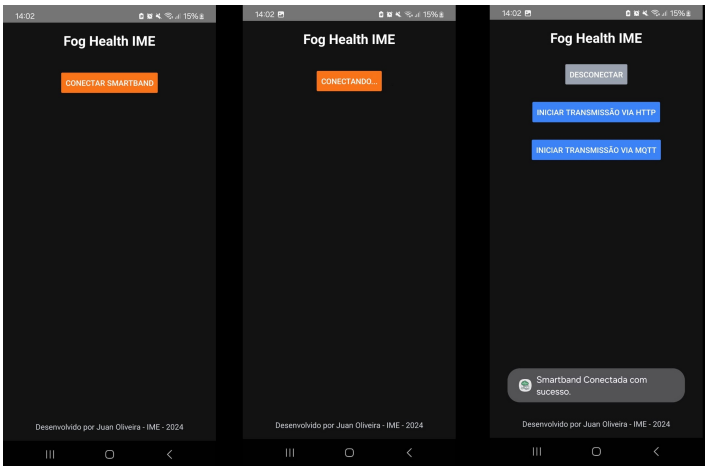


**Figure 3.** User interface of the Fog Health IME app.

- **Connect Smartband**: On the home screen, the app displays a ”Connect Smartband” button, which initiates the Bluetooth connection process between the app and the smartband. When the user selects this option, the app enters the ”Connecting...” state, displayed on the interface, until the connection is successfully established. After the connection, the app is ready to start processing the collected data.
- **Start Transmission**: Once the connection is established, the app provides the functionality to transmit the data collected by the smartband. The user can choose between the MQTT or HTTP protocols, selecting the appropriate transmission method for the experiment or application. This functionality was implemented to integrate flexibility and allow the use of multiple communication technologies, adapting to the network conditions present.
- **Connection Status and Network State**: During use, the app continuously monitors the mobile device’s network state, identifying whether it is operating on 3G, 4G, 5G, or Wi-Fi. This functionality records the connection state at the time of sending each packet, ensuring that data transmission is adapted according to network conditions and providing greater reliability in the process.
- **Data Transmission and Persistence**: After starting the transmission, the health data collected is sent directly to the cloud or fog layer. In the event of a connection failure, such as network interruptions, the data is temporarily stored locally on the device until connectivity is restored. This temporary storage is carried out using the Transactional Outbox design pattern.

This implementation aims to enable a comprehensive comparison between the MQTT and HTTP protocols, focusing on latency, reliability, and delivery rate across the defined test scenarios. By selecting these two protocols, the study seeks to evaluate their performance under adverse connectivity conditions and varying network environments, highlighting their strengths and limitations in challenging contexts.

### 4.2.3 Device Specification and Energy Impact

The experiments were conducted using a Samsung Galaxy A54 smartphone running Android 13. The device features an octa-core Exynos 1380 processor, 8 GB of RAM, and

128 GB of internal storage. These resources were sufficient to support data preprocessing, temporary local storage, and real-time transmission via MQTT and HTTP. The wearable device used was the Xiaomi Mi Smart Band 4, connected to the smartphone using Bluetooth Low Energy (BLE).

The mobile application was tested in continuous operation for periods between 90 and 120 minutes. During these sessions, battery usage was minimal: the device's battery level dropped from 90% to 86%. This result reflects the low energy impact of the system's components — including BLE communication, GPS usage, and MQTT message publishing. Although precise power profiling was not part of the scope of this study, future experiments may include energy instrumentation using tools such as Android Profiler.

## 4.3 Cloud Computing Layer

Figure 4 presents the component and tool diagram of the cloud computing architecture, which complements the fog layer. This layer, hosted on AWS (Amazon Web Services), provides a scalable and secure infrastructure, ensuring compliance with the General Data Protection Law (LGPD) by implementing strict security and privacy policies.

The data processed and transmitted via the MQTT protocol by the smartphone is sent to an MQTT broker, which acts as an intermediary in the communication between the fog layer and the cloud. In case of a disconnection of the smartphone, the broker sends logs to a NoSQL database and triggers Will Messages — messages configured to be sent automatically when a disconnection occurs — to the Subscriber API.

The Subscriber API, implemented in Java with the Spring framework, consumes messages from the MQTT broker. Spring facilitates the creation of Java applications by providing an infrastructure for managing components and configuring functionalities and integrations with the MQTT broker, RabbitMQ, and databases. The Subscriber API queues the messages in RabbitMQ, a messaging system that allows asynchronous and scalable communication between the system's components. Then, the messages are stored in the NoSQL database.

To ensure anonymization, the cloud only receives a unique identifier (ID) associated with the collected data, without personally identifiable information of the patients.

Additionally, the persistence of data in the NoSQL database, infrastructure monitoring, and alert sending are addressed in the following subsections. The first subsection explores how data is stored efficiently and securely in NoSQL, ensuring the integrity and scalability of the information. The second subsection details how Dashboard Frameworks, such as Prometheus and Grafana, are used to monitor system performance and received data, collecting key metrics in real-time to optimize the operation of the architecture. Finally, the third subsection discusses real-time alert strategies, explaining how the system reacts to critical events and sends relevant notifications to users and healthcare professionals.

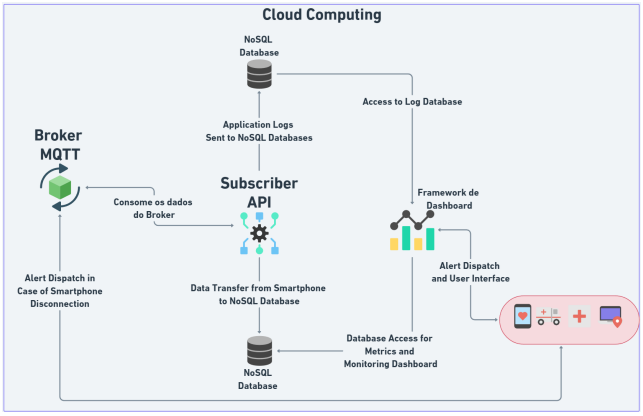


**Figure 4.** User interface of the Fog Health IME app.

### 4.3.1 Data Persistence in the Cloud NoSQL Database

The parameters sent by the smartband are received in the cloud by the Subscriber API application, where they undergo an integrity check before being stored in the NoSQL database. This database is configured to store data in JSON format, providing flexibility and efficiency in managing large volumes of heterogeneous information.

In the implementation, each data packet is stored as a JSON document, allowing fast queries and efficient indexing of key monitoring fields. The structure, illustrated in Figure 5, includes health and contextual information such as client identifier (`clientMqttId`), patient identifier (`patientId`), network state (`networkState`), heart rate, geolocation (latitude and longitude), battery level, and timestamps.

Two timestamp fields are particularly important: `sensorTimestamp`, which records the exact moment the data was collected by the wearable, and `savedAtDbTimestamp`, which marks when the data was stored in the cloud database. These timestamps enable the measurement of latency between collection and persistence.

As shown in Figure 5, each packet has an average size of approximately 300 bytes in JSON format. With a fixed transmission interval of one packet per second, the estimated throughput per device is around 2.4 kilobits per second. This bandwidth usage is minimal, even under constrained network conditions such as 3G, ensuring compatibility with unstable environments while maintaining continuous monitoring capability.

```
db.getCollection("sensor_data_http").find({})
{
    "_id" : ObjectId("6701b9c8fc4c5f31ff34a886"),
    "networkState" : "3G",
    "clientHttpId" : "ime-1-xiaomi",
    "heartRate" : 75,
    "sensorTimestamp" : ISODate("2024-10-06T01:12:23.593+0000"),
    "savedAtDbTimestamp" : ISODate("2024-10-06T01:12:24.541+0000"),
    "_class" : "com.ime.comp.fog_health.http.model.HttpData",
    "batteryLevel" : 21,
    "geoLocation" : {
        "latitude" : -22.9121,
        "longitude" : -43.2302
    },
    "patientId" : "8d6bc9e2-99bf-4a6b-9452-12766ffd7d2f"
}
```

**Figure 5.** Example of a JSON document persisted in the cloud NoSQL database.

### 4.3.2 Cloud Monitoring

With the use of Prometheus, a monitoring and alerting system that collects and stores metrics in time series, providing flexibility in data collection and querying, it is possible to monitor the health of the system and data integrity. Prometheus collects data from various system components, including the MQTT broker, the Subscriber API, and the NoSQL database, and stores these metrics in a database optimized for temporal queries.

Grafana is an open-source visualization and analysis platform that integrates with Prometheus to generate interactive dashboards and control panels. These dashboards allow real-time monitoring of patients' health parameters, trigger alerts when specific metrics reach certain thresholds, and provide a user interface (UI) that can be remotely accessed by healthcare professionals. Grafana allows users to create customized visualizations of data collected by Prometheus, making it easier to detect anomalies and make informed decisions.

In Figure 6, the dashboard generated in Grafana is shown, using data from Prometheus to display visualizations of system metrics. This dashboard provides a comprehensive view of the connectivity status of the smartbands and network conditions, with indicators such as the volume of data received by connection type (3G, 4G, and 5G), average latency per network, and accumulated delay in minutes.

The dashboard allows users to observe disconnection periods of devices, visible in the connectivity status section, where interruptions in communication with specific devices are displayed. These disconnections, monitored in real-time, reflect network intermittencies and instabilities and enable a detailed analysis of the architecture's behavior under adverse connectivity conditions, as well as the persistence of these monitoring logs in the database for further analysis.

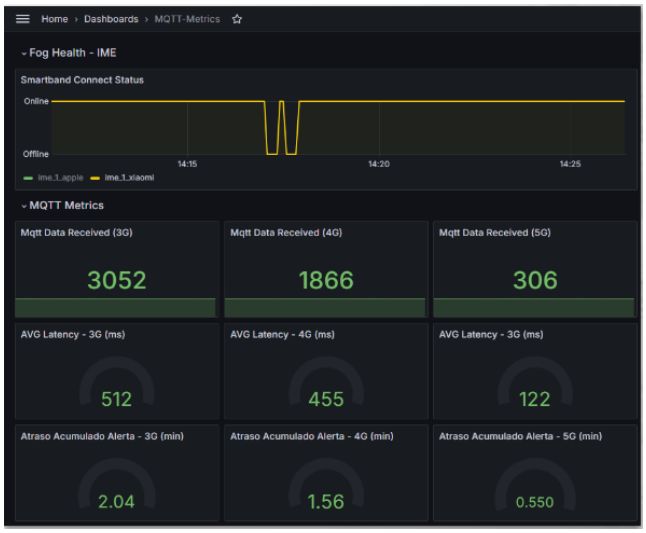


**Figure 6.** Cloud monitoring dashboard.

### 4.3.3 Connection and Disconnection Notification Triggers

Figure 7 illustrates the notification architecture in cloud computing for monitoring connection and disconnection events of devices with the MQTT Broker provided by AWS IoT Core. This service receives the device status data and forwards it to the IoT Rules service. The IoT Rules filters and processes these messages, detecting specific connection and disconnection events.

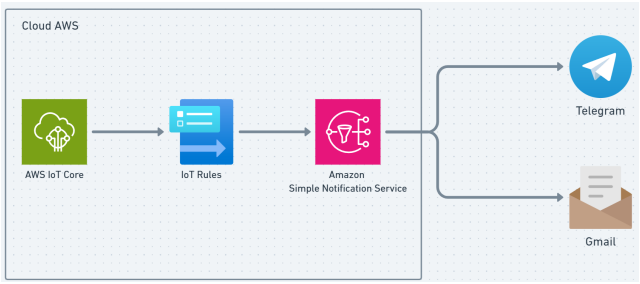


**Figure 7.** Notification flow for device connection and disconnection.

- **AWS IoT Core**: An AWS service that provides the input data for the architecture, responsible for managing the MQTT communication of connected devices and forwarding this data to the next component, *IoT Rules*.
- **IoT Rules**: Responsible for processing and filtering the messages received by AWS IoT Core. The rules configured in *IoT Rules* specifically identify connection and disconnection messages from devices. When a message with the desired status is detected, the rule triggers a notification, sending the event to *Amazon SNS* for notification delivery.
- **Amazon Simple Notification Service (SNS)**: This service manages the delivery of notifications to configured channels. Upon receiving a connection or disconnection event from *IoT Rules*, *SNS* automatically triggers a notification to the configured destinations, which include *Telegram* and *Gmail* channels.
- **Notification Channels (*Telegram* and *Gmail*)**: Notifications are sent to *Telegram* and *Gmail*. The notifications are instantly received, providing a fast and accessible way for the team to monitor the status of patients.

Figure 8 shows the connection and disconnection notifications from health monitoring devices being successfully sent to the channels configured in *SNS*. This allows real-time detection of device connectivity status, ensuring that any changes are instantly recorded and monitored.

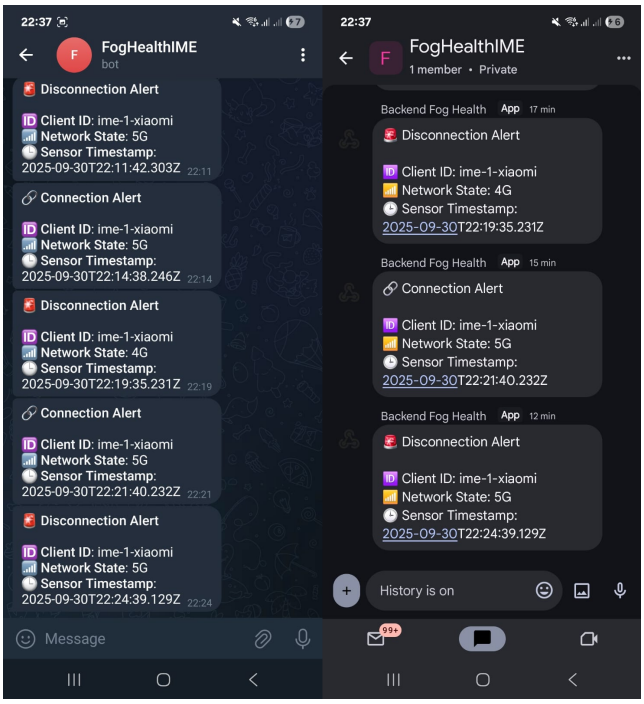


**Figure 8.** Connection and disconnection notification flow delivered through Telegram and Google Chat.

# 5 Security, Privacy and Scalability Considerations

## 5.1 Security Considerations

Security and privacy are critical concerns in the proposed architecture, given the continuous transmission of health-related data. To address these concerns, multiple protection mechanisms were implemented, covering data in transit, data at rest, authentication, and anonymization.

All communication between the smartphone (used as a fog node) and the cloud infrastructure is encrypted using the SSL/TLS protocol. In particular, the MQTT connection to the broker hosted on AWS IoT Core requires client authentication through X.509 digital certificates. This setup ensures mutual authentication and guarantees that only registered and authorized devices are able to publish health data, preventing spoofing or unauthorized access attempts.

In the cloud environment, the persisted data is encrypted using AES-256 algorithms, with encryption keys securely managed via AWS Key Management Service (KMS). Access to the NoSQL databases and log repositories is restricted through granular Identity Access Manager (IAM) policies, which define specific permissions for each service and user role. This guarantees that only legitimate services can read or write sensitive data.

To ensure compliance with privacy regulations such as the Brazilian LGPD and the European GDPR, the system adopts pseudonymization techniques. Each device is associated with a clientHttpId and a patientId, which are anonymized identifiers not directly linked to any personal information. Additionally, no names or sensitive personal identifiers are transmitted or stored as part of the data payload.

The architecture also ensures the authenticity of the device that sends the data. Each smartphone must present a valid client certificate signed by the system's root certificate authority to connect to the MQTT broker. The broker validates the certificate before accepting any data, establishing a secure trust relationship with the device. All received messages are logged with metadata, including the device identifier, timestamp, and IP address, which supports traceability and forensic analysis.

Furthermore, a dedicated NoSQL database stores application logs, including authentication events and access patterns, allowing for security auditing and anomaly detection. This contributes to the overall robustness and transparency of the system.

While the primary focus is on transmission and cloud security, local risks on fog devices are also acknowledged. Potential threats such as smartphone compromise, insecure BLE pairing with the wearable device, or unencrypted local storage may affect data privacy. Although these aspects are beyond the scope of this study, they can be mitigated through best practices such as biometric or PIN-based access control, encrypted BLE communication, and secure local storage using Android EncryptedSharedPreferences or SQLCipher. Future versions of the architecture may incorporate these safeguards to further reinforce device-level security.

This layered approach ensures data confidentiality, integrity, and authenticity while maintaining low latency and compatibility with unstable mobile network environments. Future improvements may include end-to-end encryption (E2EE) and hardware-backed key storage on smartphones to enhance security even further.

## 5.2 Scalability and Device Heterogeneity

While the evaluation focused on a single smartphone, the architecture was designed to support large-scale deployments. Its scalability relies on the elastic nature of the cloud environment, which allows automatic provisioning of resources to meet growing demands. This includes the dynamic allocation of computing power and storage as the number of connected devices increases.

The use of MQTT as a lightweight messaging protocol enables efficient management of high-throughput, low-latency communication. By organizing messages into topics, the system can isolate streams by device, user, or context, allowing fine-grained control and parallel processing of incoming data. Additionally, MQTT brokers can scale horizontally to handle thousands of simultaneous connections, making them well-suited for health monitoring applications with continuous data flow.

On the backend, the architecture incorporates loosely coupled services and asynchronous processing pipelines. This ensures that data ingestion, processing, and persistence can grow independently, without introducing bottlenecks. The NoSQL database supports schema flexibility and high write throughput, which further enhances the system's ability to ingest heterogeneous data types from a variety of mobile devices and sensor configurations.

Together, these characteristics allow the proposed system to maintain performance and reliability even as the number of monitored users increases or device capabilities vary.

Although the current implementation used a single smartphone model, the proposed architecture is designed to support a wide range of mobile devices and operating systems. To ensure interoperability, the application was developed with cross-version compatibility in mind and relies on widely supported protocols such as BLE, GPS, and MQTT. For real-world deployment, we recommend a minimum hardware specification of 4GB RAM, 32GB and Android 9 or higher, which ensures sufficient processing power and background service stability. Additionally, the use of adaptive data collection intervals and modular sensor integration allows the system to accommodate devices with varying capabilities and energy constraints. Future iterations may incorporate runtime profiling to dynamically adjust processing and transmission strategies based on device performance.

# 6 Evaluation

In addition to the main layered architecture developed (fog and cloud), referred to as the Fog Computing Implementation (FCI), a secondary implementation called the Non-Fog Computing Implementation (NFCI) was also developed. The NFCI implementation uses only the cloud computing layer, where the *smartphone* sends the data directly to the cloud via the *HTTP* communication protocol, without utilizing the fog computing layer.

We will evaluate the developed architecture by comparing the two distinct implementations described. This evaluation aims to identify the advantages and limitations of each approach, especially in terms of reliability and efficiency in transmitting health data.

Finally, the evaluation will include an analysis of the architecture as a whole, considering aspects such as latency, alert generation, and the ability to operate in different environments with varying connectivity. Evaluation criteria will include connection stability, data transfer rate, packet loss, and effectiveness in real-time alert generation.

In this context, a data packet refers to the set of information collected from the patient's sensors every second, such as heart rate data, geolocation, and network state at the time of collection. These packets are continuously transmitted every second to the cloud processing layer, where they are analyzed in real-time.

## 6.1 Main Architecture and Secondary Implementation

The use of the *HTTP* protocol presents significant limitations compared to *MQTT*. *HTTP* is a request-response-based protocol, meaning that communication is unidirectional and only occurs when the *smartphone* sends a request and the cloud responds. This can result in higher latency and lower efficiency in real-time data transmission, as each communication requires the establishment of a new connection (Sharma and Gupta, 2023). Additionally, *HTTP* lacks internal mechanisms to ensure data delivery, which can lead to data loss in unstable or intermittent network conditions (Sasaki and Yokotani, 2019). These limitations can negatively impact the reliability and efficiency of transmitting critical health data.

In contrast, *MQTT* is designed for continuous bidirectional communication, allowing data to be sent and received in real-time with lower latency. *MQTT* offers message delivery guarantees through different Quality of Service (QoS) levels, which are essential for the reliability of health data in environments with unstable connectivity (Alshammari, 2023).

In the architecture, we use QoS level 1, which ensures that a message is delivered at least once, with acknowledgment of receipt. This service level strikes a balance between reliability and efficiency, ensuring data packets are delivered even in the case of temporary network failures, without the overhead of multiple redundant transmissions (Alshammari, 2023). This maintains a high message delivery rate with low latency.

## 6.2 Evaluation Criteria

The comparison between Fog Computing (FCI) and Non-Fog Computing (NFCI), focusing on the connection with the *MQTT Broker*, will address limitations identified in previous works, such as the lack of integration between different communication technologies and the underutilization of the computational power of *smartphones*.

The architecture aims to overcome these limitations by using *smartphones* as fog nodes. The application of *NoSQL* databases and the use of the *MQTT* protocol for continuous real-time data flow are central elements of this approach.

The comparison between the FCI and NFCI implementations will be conducted based on the following criteria:

- **Connection Stability**: Evaluate the frequency of disconnections and the ability to automatically reconnect.
- **Latency**: Measure the time taken for data transmission from collection to persistence in the cloud-based databases.
- **Data Transfer Rate**: Evaluate the amount of data transmitted during the experiments.
- **Packet Loss**: Monitor the data lost during transmission.

## 6.3 Test Scenarios

Data collection and measurements were conducted in six distinct scenarios using the *Open Signal* software. This tool tests network signals, including *download*, *upload*, and latency, while also allowing the detection of signal presence and quality as well as network performance evaluation.

## 6.4 Test Scenarios

The scenarios evaluated in this study were chosen to represent environments with challenging connectivity characteristics, such as high population density, environmental interference, and adverse network conditions. Below, we describe the tested scenarios and the observed results:

- **Metro Rio, Line 1 - Rio de Janeiro - RJ:** Constant variations in signal quality were observed, with areas of no connectivity and weak to moderate signal strength. Underground stations exhibited high latency and frequent disconnections, compromising the continuity of real-time health monitoring.

- **Maracanã Stadium, Rio de Janeiro - RJ**: During game days, a high concentration of mobile devices connected simultaneously to the network was observed, causing transmission capacity overload. This congestion resulted in conflicts and interference, negatively impacting network stability and data transmission.
- **Nilton Santos Olympic Stadium (Engenhão), Rio de Janeiro - RJ**: On game days, the environment exhibited similar characteristics to the Maracanã Stadium, with overloaded networks due to the large number of connected devices. This overload caused instability, significant transmission delays, and occasional packet losses.
- **Copacabana Beach, Rio de Janeiro - RJ**: The evaluation on the beach revealed that the high concentration of people in the area resulted in significant variations in network quality, similar to those found in stadiums. Additionally, the proximity to the sea introduced unique instability factors, including environmental interference that negatively impacted signal quality, especially during peak hours.
- **Guanabara Bay, Rio de Janeiro - RJ**: During the ferry transit between the cities of Rio de Janeiro and Niterói, frequent changes in signal quality were observed. This route presented additional challenges due to marine environmental interference and the constant adaptation of networks to coverage areas. As a result, there was an increase in latency, occasional disconnections, and variations in the stability of data transmission.
- **Domestic Flight Brasília - Rio de Janeiro**: In the aerial scenario, monitoring was conducted during a commercial flight aboard an Airbus A320 between the cities of Brasília and Rio de Janeiro. During the route, connectivity was characterized by intermittent periods of connection due to the technical limitations of the Wi-Fi network provided onboard. Latency was significantly higher compared to other scenarios, often exceeding 1000 ms. Additionally, transmission was affected by constant network changes during the flight, depending on the coverage provided by satellites or ground stations.

These test scenarios highlight the significant challenges in maintaining a stable and reliable network connection in urban and high-density population environments. The analysis of this data is essential to evaluate the effectiveness of the FCI and NFCI implementations under different connectivity conditions and to identify potential areas for improvement in the transmission of critical health data.

# 7 Results

This section analyzes the results of two experiments conducted in the test scenarios shown in Figures 9 and 10.

- **Experiment I - Performance Evaluation of NFCI and FCI Architectures**: This experiment was conducted in three environments with distinct connectivity characteristics: the Rio de Janeiro Metro, Maracanã Stadium, and Nilton Santos Olympic Stadium (Engenhão). The aim was to evaluate the efficiency of the NFCI and FCI architectures in handling network intermittencies typical of high-circulation environments with significant connectivity variations. During the tests, data packets were sent every second to observe the capability of both architectures (NFCI and FCI) to maintain data integrity and continuity.
- **Experiment II - Latency and Alert Emission Tests**: This experiment was conducted at the Nilton Santos Olympic Stadium (Engenhão), during the ferry transit across Guanabara Bay connecting the cities of Rio de Janeiro and Niterói, at Copacabana Beach, and on a domestic flight between Brasília and Rio de Janeiro. These scenarios presented varying network conditions due to factors such as natural interference and mobility. In each location, tests were performed to measure the response time between data collection and alert emission, evaluating the system's ability to quickly respond to critical events, even under adverse connectivity conditions. This experiment validates the FCI architecture in situations where latency can directly impact decision-making in remote monitoring.

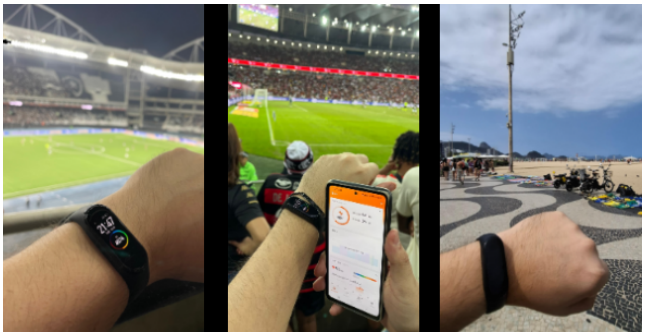

**Figure 9.** Data collection in test scenarios: Nilton Santos Olympic Stadium (Engenhão), Maracanã Stadium, and Copacabana beachfront, using the Xiaomi Mi Smart Band 4 during the experiments.

**Figure 10.** Data collection in test scenarios: Metro Rio, ferry Guanabara Bay crossing, and domestic flight Brasília–Rio de Janeiro, using the Xiaomi Mi Smart Band 4 during the experiments.

## 7.1 Performance of FCI and NFCI Architectures - Metro

During the experiment in the metro, both architectures were used simultaneously, and 12 transmission interruptions were identified, lasting between 10 seconds and 2 minutes, over a total of 90 minutes, with 5400 data packets sent. These interruptions resulted in packet losses. Table 1 presents the

results, highlighting the performance of each architecture in handling these interruptions.

**Table 1.** Comparison of Performance Metrics - metro

| Metrics | NFCI | FCI |
|---|---|---|
| Total Transmitted Packets | 5400 | 5400 |
| Packet Loss | 1680 | 176 |
| Successfully Delivered Packets | 3720 | 5224 |
| Packet Delivery Ratio (%) | 68.89 | 96.73 |
| Packet Loss Ratio (%) | 31.11 | 3.27 |

This NFCI approach, although simple through the HTTP protocol, may introduce additional latency and is more susceptible to data loss in cases of connection interruptions, due to the need to establish a new connection for each data transaction. The values observed in the table for NFCI reflect this dynamic, with a success rate of 68.89% and a failure rate of 31.11%, highlighting potential challenges in terms of latency and data loss during connectivity disruptions.

On the other hand, the FCI architecture, by using the MQTT protocol, supports persistent connections and allows the use of the Transactional Outbox pattern to log locally collected data before sending it to the cloud in cases of disconnection and network interruptions. This approach offers an advantage in outdoor environments, where connectivity has proven to be unstable. The data in the table for FCI show a success rate of 96.73% and a low failure rate of 3.27%, highlighting the efficiency of the FCI architecture in ensuring a quick response to critical events with minimal network overhead and lower susceptibility to data loss.

## 7.2 Performance of the NFCI and FCI Architectures - Maracanã Stadium

The experiment was conducted at Maracanã Stadium during the Flamengo versus Vasco match on June 2, 2024, with an attendance of 62,288 spectators. Over the two-hour event, 7,200 data packets were transmitted. Each data packet sent during both experiments represents a sample containing information such as device battery level, heart rate, and geolocation. The packets were transmitted every second, enabling continuous and real-time monitoring.

A significant degradation in mobile network quality was observed due to the overload of connected devices, leading to increased latency, reduced transfer rates, and packet loss. Table2 presents a comparative analysis of the performance of each architecture under these conditions.

**Table 2.** Comparison of Performance Metrics - Maracanã Stadium

| Metrics | NFCI | FCI |
|---|---|---|
| Total Transmitted Packets | 7200 | 7200 |
| Packet Loss | 2569 | 215 |
| Successfully Delivered Packets | 4631 | 6985 |
| Packet Delivery Ratio (%) | 64.32 | 97.02 |
| Packet Loss Ratio (%) | 35.68 | 2.98 |

The Maracanã Stadium, being an open-air venue and subject to various network conditions, represents a complex scenario for mobile connectivity. The FCI architecture demonstrated greater efficiency in packet delivery, achieving a success rate of 97.02% and a failure rate of only 2.98%. In comparison, the NFCI architecture showed a success rate of 64.32% and a failure rate of 35.68%. These results highlight how the FCI architecture provided lower data loss and better connection stability.

## 7.3 Performance of the NFCI and FCI Architectures - Nilton Santos Olympic Stadium (Engenhão)

Following the evaluation at Maracanã Stadium, a similar experiment was conducted at Nilton Santos Olympic Stadium during the Vasco versus Fluminense match on August 10, 2024, with an attendance of 20,003 spectators (Figure 9).Over the two-hour event, 7,200 data packets were transmitted. As observed previously at Maracanã Stadium, significant degradation in mobile network quality was detected due to the overload of connected devices, as summarized in Table 3.

**Table 3.** Comparison of Performance Metrics - Nilton Santos Olympic Stadium (Engenhão)

| Metrics | NFCI | FCI |
|---|---|---|
| Total Transmitted Packets | 7200 | 7200 |
| Packet Loss | 2231 | 198 |
| Successfully Delivered Packets | 4969 | 7002 |
| Packet Delivery Ratio (%) | 69.01 | 97.25 |
| Packet Loss Ratio (%) | 30.99 | 2.75 |

The FCI architecture demonstrated greater efficiency in packet delivery, achieving a success rate of 97.25% and a failure rate of only 2.75%. In comparison, the NFCI architecture showed a success rate of 69.01% and a failure rate of 30.99%. These results highlight how the FCI architecture provided lower data loss and better connection stability.

These findings emphasize the advantage of the FCI architecture in outdoor environments, where the network experiences significant overload and interference.

## 7.4 Latency Testing and Alert Transmission

Latency tests were conducted in real-world scenarios with highly variable network conditions: the Estádio Olímpico Nilton Santos (Engenhão) during a football event, Copacabana Beach on June 23, 2024—a Sunday with intense public presence, the ferry route between Rio de Janeiro and Niterói across Guanabara Bay, and a domestic flight between Brasília and Rio de Janeiro. These locations were chosen to simulate challenging connectivity environments and evaluate the responsiveness of the system in terms of latency and alert delivery.

The monitoring system is configured to detect continuous 5-minute segments in which the heart rate remains outside normal thresholds. This interval was selected based on medical studies on early warning systems and heart rate monitoring (Ilyas *et al*., 2022; Vilela *et al*., 2018), which indicate that sustained anomalies over this duration can reflect acute or chronic conditions requiring attention. It also aligns with

the design of clinical protocols that avoid false positives by filtering transient fluctuations.

Once an abnormal 5-minute segment is identified, the system triggers alerts to notify designated healthcare professionals or caregivers. These alerts depend on real-time data transmission from the fog layer (smartphone) to the cloud processing layer, and any delay impacts the timeliness of intervention.

Throughout the experiments, network fluctuations led to automatic switching between *3G*, *4G*, and *5G*, particularly in congested areas such as the stadium and beach. These fluctuations revealed latency variations directly related to network quality. The architecture's ability to maintain performance despite degraded connectivity demonstrates its robustness. Comparative latency was also analyzed between the FCI (with fog computing) and NFCI (cloud-only), showing superior responsiveness of the proposed FCI architecture in scenarios with intermittent connectivity.

### 7.4.1 Latency Test Results

The results presented in the table below highlight the performance differences between the architectures:

**Table 4.** Latency Testing - Nilton Santos Olympic Stadium (Engenhão)

| Metrics | NFCI | FCI |
|---|---|---|
| Average Response Time *5G* (ms) | 871 | 98 |
| Average Response Time *4G* (ms) | 1314 | 245 |
| Average Response Time *3G* (ms) | 2608 | 383 |

**Table 5.** Latency Testing - Copacabana Beach

| Metrics | NFCI | FCI |
|---|---|---|
| Average Response Time *5G* (ms) | 1116 | 110 |
| Average Response Time *4G* (ms) | 1814 | 312 |
| Average Response Time *3G* (ms) | 3306 | 409 |

**Table 6.** Latency Testing - Guanabara Bay

| Metrics | NFCI | FCI |
|---|---|---|
| Average Response Time *5G* (ms) | 1951 | 326 |
| Average Response Time *4G* (ms) | 2697 | 754 |
| Average Response Time *3G* (ms) | 4012 | 1229 |

**Table 7.** Latency Testing - Domestic Flight - Brasília - Rio de Janeiro

| Metrics | NFCI | FCI |
|---|---|---|
| Average Response Time *5G* (ms) | 2951 | 433 |
| Average Response Time *4G* (ms) | 3665 | 819 |
| Average Response Time *3G* (ms) | 5542 | 1329 |

### 7.4.2 Calculation of Cumulative Delay for Alert Emission

The cumulative delay is a measure of the total time elapsed from the detection of an event to the issuance of an alert, taking into account network latency. This delay is influenced by the latency in sending each data packet over the network. The formula to calculate the cumulative delay in minutes is given by:

$$\text{Cumulative Delay (min)} = \frac{\text{Average Response Time (ms)} \times 300}{1000 \times 60}$$

Cumulative Delay for Alert Emission results, are presented in the following tables:

**Table 8.** Cumulative Delay Across 3G, 4G, and 5G Networks - Nilton Santos Olympic Stadium (Engenhão)

| Metrics | NFCI | FCI |
|---|---|---|
| Cumulative Delay *5G* (min) | 4.36 | 0.49 |
| Cumulative Delay *4G* (min) | 6.57 | 1.23 |
| Cumulative Delay *3G* (min) | 13.04 | 1.92 |

**Table 9.** Cumulative Delay Across 3G, 4G, and 5G Networks - Copacabana Beach

| Metrics | NFCI | FCI |
|---|---|---|
| Cumulative Delay *5G* (min) | 5.58 | 0.55 |
| Cumulative Delay *4G* (min) | 1.56 | 1814 |
| Cumulative Delay *3G* (min) | 16.54 | 2.04 |

**Table 10.** Cumulative Delay Across 3G, 4G, and 5G Networks - Guanabara Bay

| Metrics | NFCI | FCI |
|---|---|---|
| Cumulative Delay *5G* (min) | 9.76 | 1.63 |
| Cumulative Delay *4G* (min) | 13.49 | 3.77 |
| Cumulative Delay *3G* (min) | 20.06 | 6.15 |

**Table 11.** Cumulative Delay Across 3G, 4G, and 5G Networks - Domestic Flight - Brasília - Rio de Janeiro

| Metrics | NFCI | FCI |
|---|---|---|
| Cumulative Delay *5G* (min) | 14.77 | 2.165 |
| Cumulative Delay *4G* (min) | 18.32 | 4.09 |
| Cumulative Delay *3G* (min) | 27.17 | 6.64 |

### 7.4.3 Analysis of Results

The test results indicate that the FCI architecture exhibits significantly shorter response times compared to the NFCI architecture in both analyzed scenarios. At the Nilton Santos Olympic Stadium (Engenhão), the FCI architecture achieved lower cumulative delays across all tested networks (5G, 4G, and 3G) compared to the NFCI. In contrast, the Guanabara Bay scenario presented higher delays compared to the stadium experiments, likely due to the maritime environment. The movement of the vessel, greater distance from cellular towers, and signal propagation over water may influence connection quality, contributing to increased latency observed throughout the journey.

#### 7.4.4 Impact of Delay on Alert Emission

The delay in alert emission can have a significant impact on the ability of healthcare professionals to respond to critical situations. In the context of continuous patient monitoring, particularly in large-scale and high-density environments such as a stadium, prompt detection and response to heart rate anomalies can be crucial to saving lives.

- **FCI Architecture**: With a cumulative delay of approximately 2 minutes in the worst-case network scenario (*3G*), the architecture enables healthcare professionals to be promptly notified of critical conditions. This agility can be vital for initiating immediate medical interventions and preventing severe complications.
- **NFCI Architecture**: In contrast, it exhibits a cumulative delay ranging from approximately 13 to 27 minutes in the worst-case network scenario (*3G*). This significantly longer response time can delay the notification of healthcare professionals, reducing the window of opportunity for swift interventions and potentially compromising the effectiveness of medical care.

#### 7.4.5 Latency Variability

In addition to average response times, analyzing latency variability is essential to assess the reliability of each architecture under unstable network conditions. Metrics such as standard deviation and interquartile range (IQR) provide a better understanding of performance consistency.

Although this paper focuses primarily on average cumulative delays, preliminary analysis indicates that the FCI architecture not only delivers lower mean latency but also shows reduced variability compared to the NFCI. This is particularly evident in 3G and 4G network scenarios, where packet loss and retransmission events are more frequent.

Lower latency variability suggests greater system predictability a factor in real-time health monitoring systems where consistent delivery is as important as speed. Future versions of this study will include a complete statistical analysis of latency variance to validate this observation across all test environments.

## 8 Discussion

The tests conducted in real-world scenarios demonstrated that the FCI architecture, leveraging the smartphone as a fog node, is more effective in locally processing data, reducing latency, and minimizing dependency on constant cloud connectivity. This approach proved essential for maintaining real-time monitoring continuity, meeting Quality of Service (QoS) requirements in critical healthcare applications.

The integration of fog computing and the use of the MQTT protocol in the FCI architecture provided significant benefits to key QoS metrics:

- **Latency Reduction**: Local processing at the fog node enabled data to be analyzed quickly before being transmitted to the cloud, reducing total delay and improving system responsiveness.
- **Transmission Reliability**: The protocol demonstrated resilience in scenarios with unstable networks, ensuring continuous data transmission despite latency variations or temporary disconnections.
- **Minimization of Packet Loss**: Although marginal packet loss was attributed to interference and network congestion, the packet delivery success rate exceeding 96% highlights the reliability of the implemented architecture.

The utilization of MQTT's *Will Messages* feature was a critical factor in ensuring message delivery during disconnections with the broker. This functionality maintained data analysis continuity and mitigated the impact of temporary failures on monitoring quality. In general medical applications, this capability is vital to ensuring that data gaps do not compromise clinical analysis or delay medical interventions.

The FCI architecture demonstrated substantial advantages over the NFCI, which relies exclusively on cloud processing and storage. During testing, the FCI recorded lower cumulative delays across all evaluated scenarios, even in congested networks, highlighting its superiority in terms of latency. Furthermore, it achieved a significantly higher packet delivery success rate, especially in 3G and 4G networks, showcasing its reliability. The use of fog-based databases and the MQTT protocol optimized network usage, ensuring higher data integrity and reducing dependency on continuous infrastructure availability.

The results reinforce that the integration of fog and cloud computing, combined with the MQTT protocol, is an effective solution for real-time remote monitoring. This architecture not only meets the rigorous QoS requirements but also adapts to unstable network contexts, as observed in the evaluated scenarios. By reducing latency, improving reliability, and ensuring data integrity, the FCI architecture enables significant advancements in clinical practices and solutions for remote patient monitoring.

## 9 Conclusion

The proposed architecture stands out from related works by performing data collection, storage, and processing directly on the smartphone, rather than centralizing processing in the cloud. This approach reduces the dependency on constant connectivity with remote servers and enables faster responses to critical events. The results demonstrated lower latencies and higher data integrity, even in mobile networks with intermittent connectivity and instability.

The analysis presented in this paper enriches the literature on distributed architectures for healthcare by showcasing how the integration of non-relational databases and the Transactional Outbox design pattern in fog computing environments can significantly improve the responsiveness and reliability of remote monitoring systems.

This approach not only increases the success rate of packet delivery across networks but also marks a significant advancement in resilience and responsiveness for real-time monitoring systems. The findings validate the feasibility and impact of the proposed solution, positioning it as an efficient and

reliable alternative for health monitoring in outdoor environments.

Future work will focus on exploring advanced techniques to reduce electromagnetic interference and mitigate network overloads, with the goal of enhancing reliability, efficiency, and traffic management for real-time monitoring applications. This includes developing algorithms for dynamic adaptation to varying mobile network conditions, analyzing patterns of interference and congestion, and designing strategies for optimizing packet routing and prioritization

# Declarations

## Authors' Contributions

All authors contributed to the writing, review, and validation of the results presented in this manuscript. Additionally, all authors have read and approved the final version of the manuscript.

## Competing interests

The authors declare that they have no competing interests.

## Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001.

## Availability of data and materials

Data can be made available upon request.

# References

Ahmadi, Z., Haghi Kashani, M., Nikravan, M., and Mahdipour, E. (2021). Fog-based healthcare systems: A systematic review. *Multimedia Tools and Applications*, pages 1–40.

Ahmed Kamal, M., Ismail, Z., Shehata, I. M., Djirar, S., Talbot, N. C., Ahmadzadeh, S., Shekoohi, S., Cornett, E. M., Fox, C. J., and Kaye, A. D. (2023). Telemedicine, e-health, and multi-agent systems for chronic pain management. *Clinics and Practice*, 13(2):470–482. DOI: 10.3390/clinpract13020042.

Al-Joboury, I. and Al-Hemiary, E. (2018). Performance analysis of internet of things protocols based fog/cloud over high traffic. *Journal of Fundamental and Applied Sciences*, 10:176–181. DOI: 10.4314/jfas.v10i6s.113.

Al-Sakran, A., Qattous, H., and Hijjawi, M. (2018). A proposed performance evaluation of nosql databases in the field of iot. In *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, pages 32–37. DOI: 10.1109/CSIT.2018.8486199.

Alshammari, H. H. (2023). The internet of things healthcare monitoring system based on mqtt protocol. *Alexandria Engineering Journal*, 69:275–287. DOI: https://doi.org/10.1016/j.aej.2023.01.065.

Angel, N. A., Ravindran, D., Vincent, P. M. D. R., Srinivasan, K., and Hu, Y.-C. (2022). Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies. *Sensors*, 22(1). DOI: 10.3390/s22010196.

Asri, H. and Jarir, Z. (2023). Toward a smart health: big data analytics and iot for real-time miscarriage prediction. *Journal of Big Data*, 10(1):34. Received: 04 May 2022; Accepted: 23 February 2023; Published: 14 March 2023. DOI: 10.1186/s40537-023-00704-9.

Bansal, S. and Aggarwal, H. (2022). Priority-based cloud-fog architecture for smart healthcare systems. pages 1–7.

Beall, J. (2017). What I learned from predatory publishers. *Biochemia medica*, 27(2):273–278. DOI: 10.11613/BM.2017.029.

da Silva, J. V., Baranauskas, M. C. C., Gonçalves, D. A., and dos Santos, A. C. (2022). Building a space for the human in iot: Contributions of a design process. *Journal of the Brazilian Computer Society*, 28(1):80–95. DOI: 10.5753/jbcs.2022.2958.

Dar, B. K., Ali Shah, M., Shahid, H., and Naseem, A. (2018). Fog computing based automated accident detection and emergency response system using android smartphone. In *2018 14th International Conference on Emerging Technologies (ICET)*, pages 1–6. DOI: 10.1109/ICET.2018.8603557.

Do Nascimento, M. G., Iorio, G., Thomé, T. G., Medeiros, A. A., Mendonça, F. M., Campos, F. A., David, J. M., Ströele, V., and Dantas, M. A. (2020). Covid-19: A digital transformation approach to a public primary healthcare environment. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.

Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., and Mankodiya, K. (2018). Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare. *Future Generation Computer Systems*, 78:659–676.

Fernando, N., Loke, S. W., and Rahayu, W. (2013). Mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(1):84–106. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures. DOI: https://doi.org/10.1016/j.future.2012.05.023.

Garson, G. D. (2001). *Guide to writing empirical papers, theses, and dissertations*. CRC Press.

Goldbort, R. (2006). *Writing for science*. Yale University Press.

Gomes, E., Zanatta, R., Plentz, P., Rolt, C. D., and Dantas, M. (2020). An approach of time constraint of data intensive scalable in e-health environment. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 158–169. Springer.

Haertel, F., Camargo, L., Lopes, J., Pernas, A., Mota, F., Barbosa, J., and Yamin, A. (2022). Helix project: Exploring the social internet of things (siot) in care of blind people. *Journal of the Brazilian Computer Society*, 28(1):26–37. DOI: 10.5753/jbcs.2022.2210.

Hiraman, B. R., Viresh M., C., and Abhijeet C., K. (2018). A study of apache kafka in big data stream processing. In *2018 International Conference on Information , Communi-*

*cation, Engineering and Technology (ICICET)*, pages 1–3. DOI: 10.1109/ICICET.2018.8533771.

Ilyas, A., Alatawi, M., Hamid, Y., Mahfooz, S., Zada, I., Gohar, N., and Shah, M. A. (2022). Software architecture for pervasive critical health monitoring system using fog computing. *Journal of Cloud Computing*, 2022:84. DOI: 10.1186/s13677-022-00371-w.

Kashani, M. H., Madanipour, M., Nikravan, M., Asghari, P., and Mahdipour, E. (2021). A systematic review of iot in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications*, 192:103164.

Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., and Aharon, D. (2015). Unlocking the potential of the internet of things. *McKinsey Global Institute*, 1.

Mendez, D., Graziotin, D., Wagner, S., and Seibold, H. (2020). Open science in software engineering. In *Contemporary empirical methods in software engineering*, pages 477–501. Springer. DOI: 10.1007/978-3-030-32489-6_17.

Mendonça, F. M., Dantas, M. A. R., Fortunato, W. T., Oliveira, J. F. S., Souza, B. C., and Filgueiras, M. Q. (2022). Wearable devices in healthcare: Challenges, current trends and a proposition of affordable low cost and scalable computational environment of internet of things. In Bastos-Filho, T. F., de Oliveira Caldeira, E. M., and Frizera-Neto, A., editors, *XXVII Brazilian Congress on Biomedical Engineering*, pages 1301–1308, Cham. Springer International Publishing.

Mukhopadhyay, A. (2017). Qos based telemedicine technologies for rural healthcare emergencies. In *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 1–7. DOI: 10.1109/GHTC.2017.8239296.

Oliveira, J., Vidal, P., Salles, R., and Filgueiras, M. (2024). Arquitetura multicamadas para coleta e análise de dados de saúde em tempo real em ambientes externos, integrando fog computing e cloud computing. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web*, pages 63–71, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/webmedia.2024.243220.

Paulo C. S. Vidal, Ronaldo M. Salles, M. Q. F. J. F. S. O. (2024). Desenvolvimento e avaliação de uma arquitetura para monitoramento remoto em saúde utilizando fog e cloud computing. *Artigo aceito no XX Congresso Brasileiro de Informática em Saúde (CBIS)*.

Peralta-Ochoa, A. M., Chaca-Asmal, P. A., Guerrero-Vásquez, L. F., Ordoñez-Ordoñez, J. O., and Coronel-González, E. J. (2023). Smart healthcare applications over 5g networks: A systematic review. *Applied Sciences*. DOI: 10.3390/app13031469.

Piwowar, H., Priem, J., Larivière, V., Alperin, J. P., Matthias, L., Norlander, B., Farley, A., West, J., and Haustein, S. (2018). The state of OA: A large-scale analysis of the prevalence and impact of open access articles. *PeerJ*, 6:e4375. DOI: 10.7717/peerj.4375.

Rodrigues, V. F., Righi, R. R., Costa, C. A., Antunes, R. S., Bazo, R., Reis, E. S., Seewald, L. A., Junior, L. G. S., and Eskofier, B. M. (2022). Healthstack: Providing an iot middleware for malleable qos service stacking for hospital 4.0 operating rooms. *IEEE Internet of Things Journal*, 9(19):18406–18430. DOI: 10.1109/JIOT.2022.3160633.

Sasaki, Y. and Yokotani, T. (2019). Performance evaluation of mqtt as a communication protocol for iot and prototyping. *Advances in Technology Innovation*, 4(1):21–29.

Shaji, S., Sankaran, R., Guntha, R., and Pathinarupothi, R. K. (2023). A real-time iomt enabled remote cardiac rehabilitation framework. In *2023 15th International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 153–158. DOI: 10.1109/COMSNETS56262.2023.10041272.

Sharma, P. and Gupta, P. K. (2023). Optimization of iot-fog network path and fault tolerance in fog computing based environment. *Procedia Computer Science*, 218:2494–2503. International Conference on Machine Learning and Data Engineering. DOI: https://doi.org/10.1016/j.procs.2023.01.224.

Sodhro, A. H., Luo, Z., Sangaiah, A. K., and Baik, S. (2019). Mobile edge computing based qos optimization in medical healthcare applications. *Int. J. Inf. Manag.*, 45:308–318. DOI: 10.1016/j.ijinfomgt.2018.08.004.

Stavrinides, G. L. and Karatza, H. D. (2019). A hybrid approach to scheduling real-time iot workflows in fog and cloud environments. *Multimedia Tools and Applications*, 78(17):24639–24655.

Tardieu, H., Daly, D., Esteban-Lauzán, J., Hall, J., and Miller, G. (2020). Case study 2: the digital transformation of health care. In *Deliberately Digital*, pages 237–244. Springer.

Tina Victoria, A. and Kowsigan, M. (2022). Secure management of healthcare data in fog and iot networks: A short survey on existing security protocols. In *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, pages 512–518. DOI: 10.1109/ICOSEC54921.2022.9952038.

Vilela, P. H., Rodrigues, J. J., Vilela, L. R., Mahmoud, M. M., and Solic, P. (2018). A critical analysis of healthcare applications over fog computing infrastructures. In *2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–5. IEEE.