SPaMeR: Securing Patient Medical Records in the Cloud - A Microservice and Brokerless Architecture Approach

T. B. Nam[1(✉)], H. G. Khiem[1], M. N. Triet[1], K. V. Hong[1], T. D. Khoa[1], Q. T. Bao[1], N. T. Phuc[1], M. D. Hieu[1], V. C. P. Loc[1], T. L. Quy[1], N. T. Anh[1], Q. N. Hien[1], L. K. Bang[1], D. P. N. Trong[1], N. T. K. Ngan[2], H. Son[3], and H. H. Luong[1(✉)]

[1] FPT University, Can Tho city, Vietnam
namtbce161036@fpt.edu.vn, huonghoangluong@gmail.com
[2] FPT Polytechnic, Can Tho city, Vietnam
[3] RMIT University, Ho Chi Minh city, Vietnam

Abstract. The expansion of Internet of Things (IoT) technologies has revolutionized various sectors, one of the most critical being healthcare. The effective management of Patient Medical Records (PMRs) is an area where IoT plays a significant role, and its integration with Cloud Computing offers an enormous opportunity to enhance data accessibility, efficiency, and cost-effectiveness. However, the challenge of securing PMRs in the cloud remains a key concern. This paper introduces SPaMeR, an innovative IoT platform based on microservice and brokerless architecture, tailored to address this challenge and the specific requirements of healthcare environments. SPaMeR platform incorporates and extends the core functionalities of the IoT platform designed in our previous work - data collection, device and user management, and remote device control - while specifically addressing six critical issues for healthcare data: a) secure and reliable transmission of medical data, b) energy efficiency for healthcare devices, c) high-speed and accurate data collection from medical devices, d) robust security mechanisms to protect sensitive patient information, e) scalability to accommodate the ever-growing number of patients and medical devices, and f) compliance with healthcare data regulations and standards. To demonstrate the effectiveness and feasibility of SPaMeR, we provide a comprehensive evaluation with two distinct healthcare scenarios. Our results indicate significant improvements in the areas of data security, energy efficiency, and system scalability compared to traditional healthcare platforms.

Keywords: Medical record · Internet of Things · microservice · gRPC · Single Sign-On · brokerless · Kafka · micro-service · RBAC

1 Introduction

The Internet of Things (IoT) has been steadily proliferating across various sectors including smart cities, healthcare, supply chains, industry, and agriculture.

Y. Zhang and L.-J. Zhang (Eds.): ICWS 2023, LNCS 14209, pp. 32–46, 2023.
https://doi.org/10.1007/978-3-031-44836-2_3

By 2025, an estimated 75.44 billion IoT devices are projected to be interconnected [3]. Specifically, IoT has revolutionized many sectors, and healthcare stands as a significant beneficiary. The transformative potential of IoT in healthcare extends to patient care, data collection, and healthcare management, shaping a new era of medical practice [26]. As the linchpin of IoT applications, the IoT platform is pivotal in coordinating data collection, managing devices and users, and enabling remote device control.

Recently, there has been an intensified focus on the optimal design of these platforms, particularly in critical sectors such as healthcare. In these sectors, the availability and accuracy of data can substantially influence patient outcomes. Nonetheless, conventional medical IoT systems often prioritize big data or participant access control aspects, thereby downplaying the importance of accurate, rapid, and efficient data collection, power redundancy, and system expansion [26].

To address this, various architectural models have been proposed, including the 5-layer architecture: Things, Connect, Collect, Learn, and Do, introduced by [5]. This model clarifies distinct roles within an IoT Platform, mapping them to different layers for efficient implementation. While the applicability of IoT spans diverse areas, finding a universally fitting architecture remains challenging. Regardless, three features are universally indispensable for an IoT system: i) data collection; ii) device and user management; and iii) remote device control. These features align with the Things, Connect, and Collect layers in the proposed model [5]. Each layer, however, presents unique challenges that can impede the efficiency and security of the IoT system. Things, the physical devices that collect data or perform actions, often grapple with power, processing, and bandwidth limitations [15]. The Connect layer, comprising various IoT protocols like HTTP, CoAP, XMPP, AMQP, and MQTT, must sync with the hardware and network processing capabilities of Things. Despite the pros and cons of each [3,30], MQTT is frequently preferred for constrained network communications owing to its superior transmission speed and energy efficiency [14]. However, MQTT has known security limitations, with threats to Integrity, Availability, and Authentication and Authorization mechanisms [2,19].

The Collect layer is a software ensemble that accumulates data from Things via the Connect layer. The architecture of this layer is determined by the protocol used in the Connect layer. This paper scrutinizes a Collect layer architecture that uses MQTT, a popular protocol in IoT frameworks employed by prominent companies such as IBM, Amazon, and Microsoft [9]. MQTT's dependence on a pub/sub architecture and a centralized broker, however, could lead to single point failures [20], with no guarantees for message storage or the order of delivery [13]. Given the sensitive nature of medical data, robust security mechanisms and stringent privacy protections are vital to thwart unauthorized access and malicious exploits [25].

To surmount these challenges, this paper introduces SPaMeR, a groundbreaking platform that utilizes a brokerless architecture at the Collect layer and a microservice architecture. The platform employs the gRPC protocol to directly collect medical data from IoT devices (e.g., wearable/smart devices, sensors),

thereby eliminating the need for a central broker to coordinate topic-based messages. The amassed medical data is then relayed for storage, distribution, and further processing. Coupled with a microservice architecture, SPaMeR guarantees system robustness, scalability, availability, and load-bearing [1]. Additionally, SPaMeR integrates an RBAC (role-based access control) and hierarchical user management model (tree structure) to enhance authorization, thus bolstering the security of users, devices, and communication channels. Through this framework, SPaMeR aims to secure patient medical records in cloud environments, thereby strengthening the security and reliability of healthcare IoT systems. SPaMeR not only tackles conventional healthcare IoT platform challenges but also introduces advanced features like real-time alerts for medical teams and optimized access control for patient records [25].

The remainder of this paper is organized as follows. Related work is discussed in Sect. 2. SPaMeR platform and the corresponding proof-of-concept are described in Sects. 3 and 4, respectively. Sections 5 and 6 present the test results and the discussion & future work. The paper's conclusion with a summary in Sect. 7.

2 Related Work

In this section, we discuss prior research on brokerless architectures in IoT, IoT platforms based on microservices, and OAuth for IoT. These three categories form the foundation of our proposed SPaMeR architecture. Each of these domains has been studied independently in previous works, and our study aims to unite these areas into a singular, cohesive system, strengthening the security of patient medical records in the cloud.

2.1 Brokerless Architecture in IoT

Alif Akbar Pranata et al. [21] built a water quality monitoring system following a brokerless pub/sub architecture. However, their work does not discuss system scalability and security measures. Similarly, Battery Lv et al. [20] proposed a brokerless IoT system to mitigate common security issues and single-point failure problems. Their approach was well-suited for banking systems but may be limited in scenarios requiring high-speed data transmission. Ryo Kawaguchi et al. [16] explored the potential of a distributed MQTT broker-based system to avoid single-point failures. Yet, the necessity of operating numerous physical servers and sharing user location information pose cost and privacy challenges.

2.2 IoT Platform Based on Microservice

The concept of microservices in IoT has been explored by several researchers. Sergio Trilles et al. [29] built a functional microservice architecture for IoT applicable to Smart Farming, still dependent on a broker architecture for data collection. Luca Bixio et al. [4] extended the Senseioty platform with a streaming data model for real-time data collection. However, their architecture lacks comprehensive security measures and device control mechanisms.

2.3 OAuth and the Internet of Things

In their work, Paul Fremantle et al. [8] successfully demonstrated the feasibility of using OAuth, a commonly utilized open standard for access delegation, within the Internet of Things (IoT) context. Specifically, their implementation focused on enabling access control via the MQTT (Message Queuing Telemetry Transport) protocol, a lightweight messaging protocol designed for constrained devices and high-latency networks, frequently used in IoT applications.

The paper's experimental results convincingly proved that an IoT client could utilize the OAuth token to authenticate with an MQTT broker effectively. This important finding posits OAuth not merely as a suitable authorization mechanism for traditional web applications but also as a practical choice for low-capability hardware devices found in the IoT space. To achieve this, the researchers implemented the Web Authorization Protocol to generate access tokens, which were then integrated into the MQTT client. This demonstrated a powerful synergy between the OAuth standard and the MQTT protocol, expanding the horizons of possible applications in the IoT sphere. Moreover, the paper outlines the detailed procedure of a combined OAuth and MQTT implementation, particularly focusing on the internal communication process between the MQTT broker and the MQTT client.

However, an area of concern in their implementation is the reliance on RESTful services over the HTTP/1.1 protocol for packet transfer. This approach can potentially be inefficient in the IoT context, given that it might consume a substantial amount of bandwidth and energy - a limitation discussed earlier in the Introduction section. To address this, our work aims to incorporate OAuth with Single Sign-On (SSO) into a service specifically designed for device authentication. The goal is to leverage the security benefits of OAuth and the user convenience provided by SSO in an IoT setting. Importantly, we propose the usage of gRPC (Google's high-performance, open-source universal RPC framework) for device communication with this service, given its efficiency and benefits over traditional HTTP/1.1. This approach is geared towards optimizing bandwidth usage and energy consumption, thus mitigating the limitations of the implementation discussed in Fremantle et al.'s work.

3 SPaMeR Architecture

The SPaMeR architecture adopts a microservice design to foster improved scalability and robustness. This approach involves breaking down the architecture into small, loosely coupled services, which can independently develop, deploy, and scale. Each service is designed to fulfill a specific function and can communicate with others to complete more complex tasks. This architecture is divided into three core layers: the Device layer, the Server layer, and the Patient layer (Fig. 1).

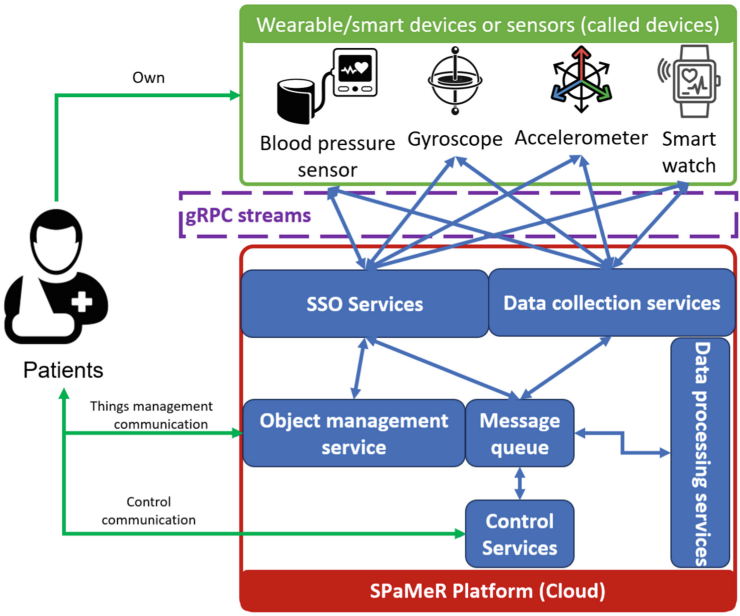

Fig. 1. SPaMeR Platform as Microservice Architecture

Device Layer. The Device layer is the frontline of the architecture, encompassing physical IoT devices such as smartwatches, blood pressure monitors, gyroscopes, accelerometers, and other devices capable of gathering medical data from patients. Each of these devices features an embedded system, which collects patient medical data and transmits it to the Server layer via the gRPC protocol, an efficient, high-performance framework for inter-service communication.

In addition to data transmission, this layer is also responsible for device control. Devices receive control commands either from the Patient layer or from the medical data processing service within the Server layer. These commands adjust the behavior of devices based on the data they collect, thus providing a dynamic, responsive system.

Server Layer. At the heart of the architecture is the Server layer. This layer consists of several microservices, each designed to perform specific tasks, demonstrating the advantages of the microservice architecture in managing complexity by breaking down system functions into manageable, independent services. These microservices include:

– **Data Collection Service:** This service interfaces with the Device layer, collecting medical data from authenticated devices. It also conveys control commands from the Patient layer or medical data processing service to the devices.

- **Single Sign-On (SSO) Service:** Authentication of devices and patients is handled by this service, which follows the OAuth protocol to provide secure, token-based authentication.
- **Object Management Service:** This service is responsible for managing system objects such as patients and devices, ensuring accurate tracking and control over these elements.
- **Control Service:** This service provides an interface for patients to remotely control their devices, forwarding the commands to the Device layer.
- **Medical Data Processing Service:** This service performs the critical task of analyzing collected medical data. It also stores system logs and issues control commands to devices based on predefined triggers.
- **Message Queue:** As the messaging backbone of the Server layer, the Message Queue handles inter-service communication, transporting messages between different services.

Patient Layer. The Patient layer forms the user-facing part of the architecture. Patients interact with the system via the Internet of Devices service. Authentication by the SSO service is required for patients to access and interact with the services in the Server layer. This secure access control mechanism helps maintain patient privacy and data security.

4 Implementation

To better comprehend the SPaMeR platform's design, it's vital to elaborate on the components in the system and their interactions. This section provides a detailed description of the software architecture and the communication workflows that underpin the system's operation.

4.1 Patients

Patients are the end-users of the IoT services offered by our platform. To accommodate various user scenarios, a hierarchical patient model is established. This model is akin to a tree structure, with parent patients being able to create and manage their child patients. This setup offers flexibility and scalability, especially useful for large-scale deployments involving multiple levels of patients, such as in a healthcare institution.

Each patient has a unique patient_id value conforming to the UUID standard, managed by the Object Management Service. This unique identifier allows for efficient and error-free patient data management. Patients use their patient_id and password to request an access token from the Single Sign On service, which is subsequently used to authorize their interactions with the system.

4.2 Devices

Devices represent the various physical devices or applications owned by the patients. These could range from wearable health monitors to home-based healthcare equipment. To register a device, the patient owning the device must provide a valid OAuth token. Once the device is registered, the patient embeds the access token into their devices. Only devices bearing a valid access token can communicate with the Medical Data Collection Service. This token-based authentication mechanism enhances security by ensuring that only legitimate devices can transmit data to the Server layer, thereby protecting the system from potential data breaches or denial-of-service attacks.

4.3 Communication Workflow

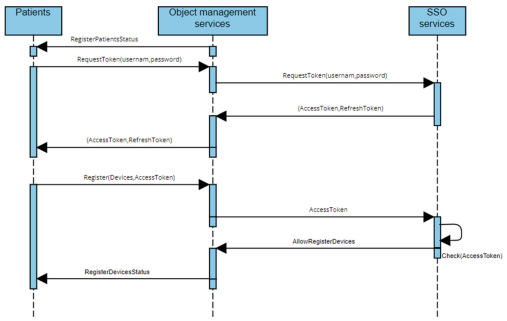

Fig. 2. The patients and their devices initialization processes

Initialization Workflow. The system initialization begins with the registration of a patient. Once registered, the patient requests an access token using their patient name (i.e., account) and password. The system returns an access token and refresh token as per the OAuth standard. With the access token, the patient can then register their devices with the system, enabling the creation of device information in the Object Management Service (Fig. 2).

Medical Data Collection Workflow. Data collection commences with devices sending their access tokens to the SSO Service for authentication. Once authenticated, the devices are permitted to stream medical data to the Medical Data Collection Service. This service then funnels the collected data to the Message Queue. As a robust inter-service communication mechanism, the Message Queue provides other services, such as the Data Processing Service, access to the medical data stream. This multi-step process ensures that data collection is secure, efficient, and capable of handling large volumes of data from numerous devices (Fig. 3).

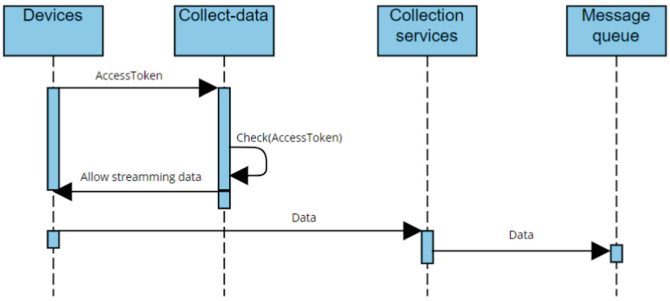

Fig. 3. Medical data collection process

Control Workflow. The control workflow allows patients to remotely control their devices. Patients initiate this process by sending their access token and the desired command to the Control Service. The SSO Service validates the access token. If the token is valid, the Control Service forwards the command to the Message Queue. The Message Queue, in turn, delivers the command to the Medical Data Collection Service, which finally transmits the command to the targeted devices. This cascading workflow ensures that device control commands are securely and efficiently transmitted, allowing for real-time device control.

The SPaMeR platform utilizes a robust and scalable microservice architecture to securely manage patient medical records in the cloud. This system design allows for secure data collection, processing, and remote device control, ensuring that patient health data can be accurately and securely managed. It forms an integral part of the wider Internet of Medical Things ecosystem, allowing for continuous health monitoring and informed medical decisions, thereby significantly enhancing healthcare outcomes (Fig. 4).

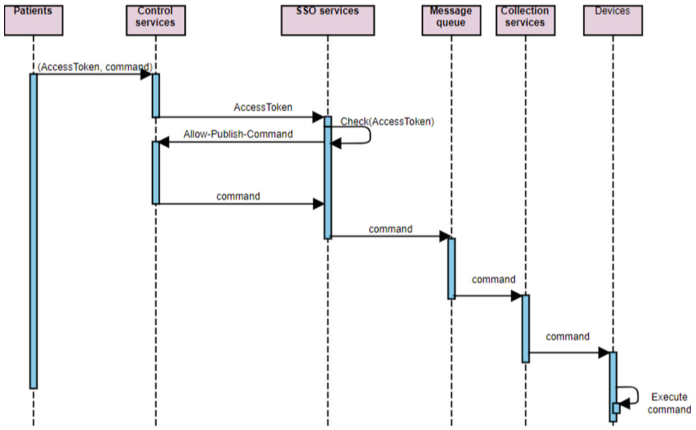

Fig. 4. Control Workflow in the SPaMeR System

5 Evaluation Scenarios

5.1 Environment Setup

The evaluation process of our proposed SPaMeR platform began with the implementation of the Medical Data Collection service. To validate the performance of our platform, particularly in terms of data transmission speed and scalability, we established a specific testing environment and created multiple scripts to emulate real-life scenarios involving the transmission of medical data over a cloud server.

Two core scenarios were formulated to evaluate the efficacy of a broker-less architecture employing the gRPC protocol versus a brokered architecture using the MQTT protocol. Both of these protocols are well-known for their efficiency and reliability in IoT communications. However, their performance characteristics differ under different circumstances, making this evaluation integral to optimizing our platform. The source codes for these scenarios can be found in our Github repository in the previous version of this paper [27], namely the Medical Data Collection service[1] and the MQTT streaming[2]. A visual representation of these scenarios is provided in Fig. 5.

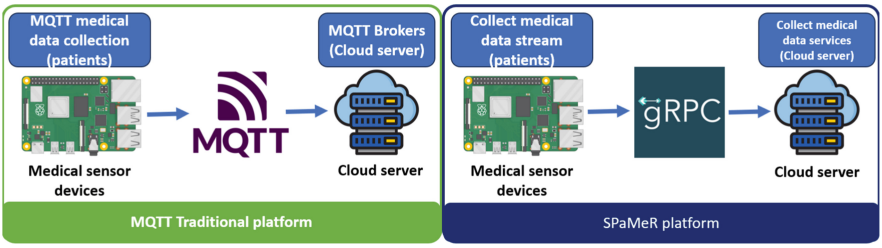

Fig. 5. Comparison scenarios of communication speed between gRPC and MQTT

The test environment comprised two testbeds, each incorporating a medical sensor devices (Raspberry Pi) module and an Amazon EC2 Server configured similarly to mirror the common resources of a typical IoT cloud server environment. This configuration enabled a fair and accurate comparison of the two protocols. Conversely, the first testbed used medical sensor devices and cloud server to deploy an MQTT client and broker, leveraging the MQTT protocol for packet transmission. In the second testbed, the and implemented the Medical Data Collection service, relying on the gRPC protocol for data transmission. This setup represented our platform's broker-less architecture, where communication is direct and efficient, albeit requiring more processing power. This setup highlighted the benefits and drawbacks of a brokered architecture where

[1] https://github.com/thanhlam2110/iot-platform-collect-data-service.
[2] https://github.com/thanhlam2110/mqtt-streaming.

an intermediary manages communication, providing better scalability but potentially limiting data transmission speed. These environmental settings provided the basis for a comprehensive evaluation of our SPaMeR platform's performance and ability to efficiently manage and transmit medical data in a cloud server setup.

5.2 Message Delivery Speed Test Case

One of the critical factors when considering the transmission of medical records in a cloud environment is the speed of message delivery. The reason for emphasizing this aspect is twofold. Firstly, healthcare applications often require real-time or near-real-time data transmission to provide accurate and timely diagnoses or recommendations. Secondly, medical emergencies demand immediate data access, and any delay in data delivery could have severe consequences.

Given this background, our first test case is designed to evaluate the message delivery speed, specifically focusing on comparing the performances of the gRPC and MQTT protocols. We perform this comparison across three runs, and the results are presented in Table 1. In particular, the performance comparison between gRPC and MQTT under different Quality of Service (QoS) settings, i.e., QoS-0 and QoS-2, was the first test case in our evaluation process. It is significant to understand that MQTT provides three levels of QoS, including:

- **QoS-0 (At most once):** The message is sent only once and not acknowledged, making it the fastest but least reliable level.
- **QoS-1 (At least once):** The message is retransmitted until it is acknowledged, providing assurance of message delivery but no guarantee of duplication.
- **QoS-2 (Exactly once):** The message is assured to be delivered exactly once by using a four-step handshake process, offering the highest level of message assurance but also being the slowest.

Table 1. Performance comparison between two scenarios (i.e., gRPC vs MQTT (QoS-0; QoS-2)) in terms of delivery speed (in seconds).

Sending time	gRPC	MQTT (QoS-0)	MQTT (QoS-2)
1st	54 s	269 s	can't complete test
2nd	50 s	294 s	can't complete test
3rd	52 s	295 s	can't complete test
Average	52 s	286 s	can't complete test

From Table 1, we can see that the gRPC protocol significantly outperforms MQTT in terms of message delivery speed. On average, gRPC took just 52 s to complete the transmission, whereas MQTT (Quality of Service level 0) took

around 286 s. Interestingly, MQTT (Quality of Service level 2) was unable to complete the test, suggesting potential issues with handling the large amount of medical data in this scenario.

The results highlight the advantage of gRPC in the case of transmitting patient medical records, reinforcing our decision to utilize this protocol within the SPaMeR platform for efficient and rapid data transmission. The platform's ability to swiftly deliver medical data could prove instrumental in time-sensitive healthcare scenarios, contributing to more efficient medical responses and potentially improving patient outcomes.

5.3 Disrupted Connection Test Cases

The resilience of a platform when handling disruptions in the connection, especially when dealing with sensitive data such as medical records, is of utmost importance. As such, we conducted a series of tests to assess the impact of connection disruptions between data publishers and subscribers in our proposed SPaMeR platform. The test model is depicted in Fig. 6. These tests compared the number of received messages in scenarios both with and without the implementation of the SPaMeR platform when a disrupted connection occurred. In the scenario without the SPaMeR platform, the subscriber could only receive a single message - the most recent message sent by the publisher when the disruption occurred. This outcome is tied to the 'retain' functionality of the MQTT protocol, which, when activated, allows the MQTT broker to store only the most recent message published by the publisher. This message is then received by the subscriber once it reestablishes its connection to the MQTT broker[3]. In stark contrast, while utilizing the SPaMeR platform, the subscriber could receive all the messages published by the publisher, even in the event of a disrupted connection. This capability is attributed to the Kafka message queue embedded within the platform, ensuring data consistency and mitigating data loss. This result

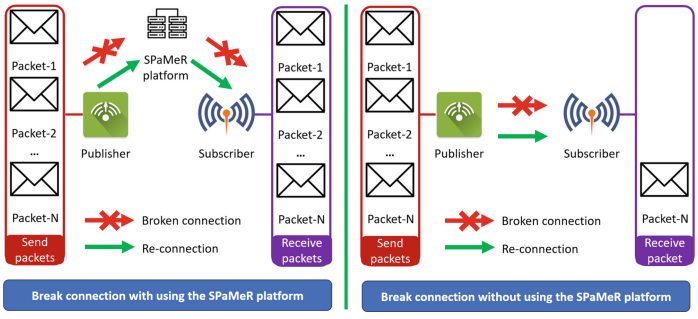

Fig. 6. The simulation of broken connection issue in the received messages when system in the two approaches

[3] http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.

not only illustrates the robustness of the SPaMeR platform when handling connection disruptions but also underscores its suitability for transmitting sensitive patient medical records reliably, even in challenging network conditions.

6 Discussion

6.1 Our Observation

Our work sought to present an improved method for ensuring the secure and reliable transmission of patient medical records within an IoT context. Based on the results of our evaluations, several key discussion points have emerged.

Firstly, the brokerless architecture combined with the gRPC protocol, as implemented in the SPaMeR platform, offers a significant performance advantage in terms of message delivery speed over the MQTT protocol, traditionally used in IoT systems. This was particularly noticeable in our Message Delivery Speed Test Case, where the gRPC implementation demonstrated a markedly faster average delivery speed. This advantage can be crucial in real-world healthcare applications where timely access to data can directly impact patient care. Secondly, the evaluation demonstrated the resilience of our platform under disrupted connection scenarios. It is inevitable that connection disruptions will occur in real-world deployments due to various factors. In such scenarios, the ability of the SPaMeR platform to ensure data integrity and continuity, thanks to the Kafka message queue, sets it apart from traditional MQTT based approaches. This resilience and reliable data delivery become even more important when the data in question are sensitive patient medical records. Finally, the use of a microservice architecture lends a high degree of modularity and scalability to the platform. This structure allows the system to be easily expanded or modified, facilitating a more agile and adaptable approach to system design and development. The microservice architecture combined with the brokerless system and the use of gRPC protocol allows our platform to handle large-scale IoT deployments effectively and efficiently, making it suitable for enterprise-level healthcare IoT applications.

However, while the initial evaluations are promising, further testing and refinement will be necessary to ensure that the SPaMeR platform can meet the rigorous demands of real-world deployments fully. Key areas for future work could include detailed security assessments, comprehensive load testing, and in-depth analysis of platform performance under various network conditions. Additionally, assessing the platform's ability to handle various types of medical data and integrate with different IoT devices will be important for broad adoption in the healthcare industry. In summary, our work presents a significant step towards the reliable and secure management of patient medical records in the IoT domain. It paves the way for more advanced, efficient, and reliable healthcare IoT systems, thereby aiding the broader mission of improving healthcare delivery and outcomes.

6.2 Prospective Developments

Considering the need for scalability to accommodate an expanding number of devices and users demanding swift authorization, addressing security concerns such as object security, privacy, and availability remain central challenges for future investigation.

Concerning security, further research will aim to introduce our solutions in various real-life scenarios. Particular attention will be devoted to the healthcare domain, given its vital societal role and the sensitive nature of data involved [6, 7, 24]. This could include applications such as patient health record management, telemedicine, and hospital information systems.

Regarding privacy, we are considering the application of attribute-based access control (ABAC) [12, 23] to manage the authorization processes of the SPaMeR Platform. ABAC's capacity to use dynamic policies could provide enhanced flexibility in controlling access based on various factors, including the user's role, the requested resource, and the context of the request [22, 28, 31].

Lastly, to enhance system availability and robustness against single points of failure, we plan to investigate the benefits of incorporating blockchain technology into the platform's infrastructure [10, 11, 25]. By decentralizing data storage and management, blockchain could significantly increase the system's resilience and guarantee data integrity [17, 18], making it an attractive solution for the future development of the SPaMeR Platform.

7 Conclusion

Our work proposed the SPaMeR platform's design and architecture, leveraging microservices and a brokerless approach, offering a robust and scalable solution for securing patient medical records in a cloud environment. By utilizing the gRPC protocol, we've presented a substantial enhancement in message delivery speed over traditional MQTT-based approaches, a critical factor in healthcare applications where real-time data accessibility can directly affect patient care. Additionally, the robustness of SPaMeR in handling connection disruptions ensures the integrity and continuity of data transmission, an indispensable requirement when dealing with sensitive patient medical records. The system's resilience stems from the implementation of the Kafka message queue, thereby ensuring no data loss occurs, a significant improvement over MQTT's handling of connection disruptions. The use of a microservice architecture enhances the scalability and modularity of the platform, enabling efficient and effective management of large-scale IoT deployments, positioning SPaMeR as a suitable choice for enterprise-level healthcare IoT applications.

Despite these promising results, we recognize the need for additional testing and refinement to fully prepare SPaMeR for the demanding real-world deployment conditions. Future work should focus on rigorous security assessments, comprehensive load testing, detailed analysis of performance under varying network conditions, and the capability to handle diverse medical data types and integrate with a wide range of IoT devices.

Acknowledgement. We would like to extend our deepest gratitude to Engineer Le Thanh Tuan and Mr. Lam Nguyen Tran Thanh for their invaluable contribution and insight throughout the conceptualization, execution, and assessment of this project.

References

1. Ali, M., Ali, S., Jilani, A.: Architecture for microservice based system. A report (2020)
2. Anthraper, J.J., Kotak, J.: Security, privacy and forensic concern of MQTT protocol. In: Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM). Amity University Rajasthan, Jaipur (2019)
3. Bansal, M., et al.: Application layer protocols for internet of healthcare things (IoHT). In: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), pp. 369–376. IEEE (2020)
4. Bixio, L., Delzanno, G., Rebora, S., Rulli, M.: A flexible IoT stream processing architecture based on microservices. Information **11**(12), 565 (2020)
5. Chou, T.: Precision-Principles, Practices and Solutions for the Internet of Things. McGraw-Hill Education, New York (2017)
6. Duong-Trung, N., et al.: On components of a patient-centered healthcare system using smart contract. In: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, pp. 31–35 (2020)
7. Duong-Trung, N., et al.: Smart care: integrating blockchain technology into the design of patient-centered healthcare systems. In: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, pp. 105–109 (2020)
8. Fremantle, P., Aziz, B., Kopeckỳ, J., Scott, P.: Federated identity and access management for the internet of things. In: 2014 International Workshop on Secure Internet of Things, pp. 10–17. IEEE (2014)
9. Fuentes Carranza, J.C., Fong, P.W.: Brokering policies and execution monitors for IoT middleware. In: Proceedings of the 24th ACM Symposium on Access Control Models and Technologies, pp. 49–60 (2019)
10. Ha, X.S., Le, H.T., Metoui, N., Duong-Trung, N.: DeM-CoD: novel access-control-based cash on delivery mechanism for decentralized marketplace. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 71–78. IEEE (2020)
11. Ha, X.S., Le, T.H., Phan, T.T., Nguyen, H.H.D., Vo, H.K., Duong-Trung, N.: Scrutinizing trust and transparency in cash on delivery systems. In: Wang, G., Chen, B., Li, W., Di Pietro, R., Yan, X., Han, H. (eds.) SpaCCS 2020. LNCS, vol. 12382, pp. 214–227. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68851-6_15
12. Hoang, N.M., Son, H.X.: A dynamic solution for fine-grained policy conflict resolution. In: Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, pp. 116–120 (2019)
13. Hwang, H.C., Park, J., Shon, J.G.: Design and implementation of a reliable message transmission system based on MQTT protocol in IoT. Wireless Pers. Commun. **91**(4), 1765–1777 (2016)
14. Jaikar, S.P., Iyer, K.R.: A survey of messaging protocols for IoT systems. Int. J. Adv. Manage. Technol. Eng. Sci. **8**(II), 510–514 (2018)

15. Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., Alonso-Zarate, J.: A survey on application layer protocols for the internet of things. Trans. IoT Cloud Comput. **3**(1), 11–17 (2015)
16. Kawaguchi, R., Bandai, M.: Edge based mqtt broker architecture for geographical IoT applications. In: 2020 International Conference on Information Networking (ICOIN), pp. 232–235. IEEE (2020)
17. Le, H.T., et al.: Introducing multi shippers mechanism for decentralized cash on delivery system. Int. J. Adv. Comput. Sci. Appl. **10**(6) (2019)
18. Le, N.T.T., et al.: Assuring non-fraudulent transactions in cash on delivery by introducing double smart contracts. Int. J. Adv. Comput. Sci. Appl. **10**(5), 677–684 (2019)
19. Lee, S., Kim, H., Hong, D.K., Ju, H.: Correlation analysis of MQTT loss and delay according to QoS level. In: The International Conference on Information Networking 2013 (ICOIN), pp. 714–717. IEEE (2013)
20. Lv, P., Wang, L., Zhu, H., Deng, W., Gu, L.: An IoT-oriented privacy-preserving publish/subscribe model over blockchains. IEEE Access **7**, 41309–41314 (2019)
21. Pranata, A.A., et al.: Towards an IoT-based water quality monitoring system with brokerless pub/sub architecture. In: 2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp. 1–6. IEEE (2017)
22. Son, H.X., Dang, T.K., Massacci, F.: REW-SMT: a new approach for rewriting XACML request with dynamic big data security policies. In: Wang, G., Atiquzzaman, M., Yan, Z., Choo, K.-K.R. (eds.) SpaCCS 2017. LNCS, vol. 10656, pp. 501–515. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72389-1_40
23. Son, H.X., Hoang, N.M.: A novel attribute-based access control system for fine-grained privacy protection. In: Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, pp. 76–80 (2019)
24. Son, H.X., et al.: Towards a mechanism for protecting seller's interest of cash on delivery by using smart contract in hyperledger. Int. J. Adv. Comput. Sci. Appl. **10**(4) (2019)
25. Son, H.X., Nguyen, M.H., Vo, H.K., Nguyen, T.P.: Toward an privacy protection based on access control model in hybrid cloud for healthcare systems. In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J.A., Quintián, H., Corchado, E. (eds.) CISIS/ICEUTE -2019. AISC, vol. 951, pp. 77–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20005-3_8
26. Thanh, L.N.T., et al.: IoHT-MBA: an internet of healthcare things (IoHT) platform based on microservice and brokerless architecture. Int. J. Adv. Comput. Sci. Appl. **12**(7) (2021)
27. Thanh, L.N.T., et al.: Sip-MBA: a secure IoT platform with brokerless and microservice architecture. Int. J. Adv. Comput. Sci. Appl. **12**(7) (2021)
28. Thi, Q.N.T., Dang, T.K., Van, H.L., Son, H.X.: Using JSON to specify privacy preserving-enabled attribute-based access control policies. In: Wang, G., Atiquzzaman, M., Yan, Z., Choo, K.-K.R. (eds.) SpaCCS 2017. LNCS, vol. 10656, pp. 561–570. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72389-1_44
29. Trilles, S., González-Pérez, A., Huerta, J.: An IoT platform based on microservices and serverless paradigms for smart farming purposes. Sensors **20**(8), 2418 (2020)
30. Verma, S., Rastogi, M.A.: IoT application layer protocols: a survey. J. Xi'an Univ. Archit. Technol. VII **57** (2020)
31. Xuan, S.H., et al.: Rew-XAC: an approach to rewriting request for elastic ABAC enforcement with dynamic policies. In: 2016 International Conference on Advanced Computing and Applications (ACOMP), pp. 25–31. IEEE (2016)