MiSIS: An HL7 FHIR Middleware for Healthcare Information Systems

Ramon Santos Malaquias(✉), Itamir de Morais Barroca Filho, Jean Mário Moreira de Lima, André Morais Gurgel, and Bruna Alice Oliveira de Brito

Digital Metropolis Institute, Federal University of Rio Grande do Norte, Natal, Brazil
{malaquias,itamir.filho,jean.lima}@imd.ufrn.br, andre.gurgel@ufrn.br

Abstract. Several e-Health and m-Health technological solutions are increasingly supporting Healthcare services. In this context, Electronic Health Records (EHR) deal with data of different types, which, if integrated, can be of great value for the patient's health care. However, data heterogeneity is a factor that makes this integration difficult, and middleware appears as an option as a solution to this problem. The basic definition of middleware is a software layer that abstracts development details and encourages integration between solutions. This work describes the requirements and implements a middleware for healthcare systems based on HL7 FHIR, an open development standard that enables interoperability between healthcare solutions. As a result, we present a proof of concept that integrates data from 5 different health systems and makes the integrated data available using the HL7 FHIR standard.

Keywords: healthcare · middleware · interoperability · e-Health · m-Health

1 Introduction

With the advancement of technological applications aimed at the healthcare context, there is a need to store, process, and extract information through data used in healthcare to support decisions and optimize processes to improve the service provided. In this sense, Health Information Systems (SIS) emerge, which, according to Fatima et al. [10], can be defined as "a set of interrelated components that collect, process, store and distribute information to support the decision-making process and assist in the control of healthcare organizations."

The use of SIS enables the creation of Electronic Health Records (EHR), defined by Iakovids [15] as digitally stored health information about an individual's life to support continuity of care, education, and research, ensuring confidentiality at all times. In this way, EHR systems allow the clinical team to access patient information [35] easily. Therefore, using EHR-based health systems makes it possible to maintain health data records, integrate services, provide information to different sectors, reduce health treatment costs,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14814, pp. 243–260, 2024.
https://doi.org/10.1007/978-3-031-64608-9_16

avoid loss of information, and contribute to better healthcare provision. Health service [5].

Solutions based on SIS and EHR are examples of *eHealth* solutions. *EHealth* is the field that emerges as an intersection between medical informatics, public health, and business, referring to improved health services through the use of technologies and web technologies [9]. Furthermore, the advancement of mobile communications, which currently support 3G, 4G, and 5G mobile networks for data transport, means that mobile computing promotes countless possibilities for creating solutions for the healthcare sector, bringing to light the concept de *mHealth* [35] [28]. *MHealth* is the area that consists of the use of mobile computing and communication technologies in health [13]. Therefore, systems based on *eHealth* and *mHealth* can support clinical and administrative data management, predictive systems, monitoring vital signs and sending alerts, medication management, and exam and results management.

In this sense, an ecosystem of digital solutions for the health sector can be composed, for example, of solutions aimed at sensing and monitoring patients' vital signs, including solutions for managing exams and results, management data from hospitals and clinics, epidemiological data, personal and sociodemographic data of patients, and etcetera. If integrated, this data could not only promote better support for clinical decision-making and the provision of health services but also help significantly reduce costs in this process. However, the heterogeneity of healthcare data is one of the main challenges in developing integrated solutions for this area [16], along with interoperability, data quality, and security and privacy [17,21,34,41].

Interoperability consists of the ability of a system to transfer data and the ability to interpret the data transferred to other systems [6]. Therefore, nowadays, one of the main requirements for improving the quality of care and services provided in the healthcare domain is to promote interoperability between the applications involved in this context. Therefore, Middleware Platforms aimed at the healthcare context emerge as options to meet the interoperability requirement since they are solutions for distributed applications that make network details transparent and protect functionality implementation details, providing interfaces for customer integration [23].

However, studies indicate that the development of *middleware* platforms for the healthcare area still has limitations, such as the non-use of open standards such as $HL7\ FHIR$. In the literature review carried out by Malaquias et al., through a Systematic Mapping [20], 14 *middleware* platforms for the healthcare area were analyzed and none of them presented a solution entirely based on an open communication standard and prepared for data exchange between services *web*, such as $HL7\ FHIR$, to promote [27] interoperability.

Based on this context, this paper aims to describe the development and validation process of a Middleware platform called MiSIS based on the $HL7FHIR$ Standard for Health Information Systems. The remainder of this work presents the following organization: Sect. 2 deals with related work and middleware solutions aimed at the healthcare area; Sect. 3 details MiSIS from the point of view

of its requirements and the software architecture designed to meet the solution's requirements; Sect. 3 deals with the current state of development of MiSIS; Sect. 4 describes the methodology used to validate MiSIS, as well as the results obtained from carrying out proofs of concept and carrying out solution availability tests; and, finally, Sect. 5 presents the conclusions obtained, as well as predictions for future work based on the development of MiSIS.

2 Related Works

To fulfill this work, the authors read and analyzed thirty-four articles to understand how each responds to the questions defined in the research protocol of the review. In this sense, solutions based on the $HL7$ [12] standard are cited as options for the development of *middleware* platforms for [22] healthcare systems. Furthermore, in Zhang et al. [40] some solutions referred to as "standards" by the authors are mentioned, such as: "*Medical Telemonitoring System*" which consists of a method of sending physiological and medical information remotely through the internet for an analysis and diagnostic center; in addition to *middleware* described, "*Mobile Context and Ontology-based Reasoning/Feedback* ($MCOM$)", a system that monitors patients' health status using *Smartphone*.

In the work of Alonso et al. [2], authors carried out on several *middleware* platforms, including Kura, a *middleware* proposed and designed by the Eclipse consortium to be part of *gateways*; AllJoyn, proposed by the Linux Foundation, which provides a version for restricted resources and also provides different APIs for developers; Macchina.io, which according to the author, is very modular and extensible, implementing a messaging protocol based on the publish/subscribe model. In the line of applications that use this model, Almadani et al. [1] proposes a *Middleware Real-Time Publish Subscribe* ($RTPS$), and Singh et al. [37] that present a *middleware* model for controlling information sharing in a publish/subscribe-based environment.

Madureira et al. [26] describes $My - AHA$, a multiplatform *middleware* designed to perfectly integrate different health and active aging solutions, aiming for well-being. According to the authors, the proposed architecture for $My - AHA$ is a multi-module system architecture, fully scalable and easily implementable, focusing on helping caregivers and seniors improve their current condition, considering changes in cognitive, physical, and physical parameters, social and psychological.

A framework for interoperability between health systems is proposed by Ryan et al. [33]. This framework is *Health Service Bus* (HSB), a solution based on the *software* architecture of *middleware Enterprise Service Bus* (ESB), which provides a weakly coupled and highly distributed for enterprise systems integration. Another example of *middleware* is $SALSA$, proposed by Rodriguez et al. [32] and which allows developers to create autonomous agents that react to contextual elements of the medical environment and communicate with other agents, users, and services available in the environment.

$POStCODE$ (*POstmarket SurveillanCe Of DEvices*) is a *middleware* proposed by Chaudhry et al. [8] which provides the operational details, taking care

to delete private patient data, from devices directly to manufacturers. Thus, it enables manufacturers to monitor and maintain medical devices. In terms of synchronizing medical data, we also have the solution proposed by Lomotey et al. [24], a *middleware* for *mHealth (mobile health)* that facilitates the efficient process of synchronizing medical data, and with minimal latency.

SBUS is *middleware* presented by Singh et al. [36]. *SBUS* is a solution that allows multiple forms of communication that are dynamically reconfigurable. It ensures that the message types of the terminals match and that each transmission (message) conforms to the type scheme. Additionally, *SBUS* supports multiple communication methods, including client-server (request-response and RPC) and *streaming* messages *push/pull*.

Some initiatives to promote interoperability between health systems at the continent or country level have also been identified in the literature. For example, Ferrara [11] proposes *DHE* (*Distributed Healthcare Environment*), which is a *middleware* following the proposed European Standard Health Information System Architecture, implementing a distributed repository for all clinical, organizational and managerial information in the healthcare structure, to make it available when and where needed for all applications in the healthcare information system. AlZghoul et al. [3] describes a *middleware* architecture to help healthcare providers in Jordan access electronic health data in a national healthcare database.

Healthcare applications deal with a series of multimedia files, mainly image format exams. In this sense, Kallepalli et al. [19] describes a security *middleware* for *DICOM* images that provide fine-grained access control, policy management, demographic filtering, and log keeping restricted to *Canadian-Manitoban* PHIA and the DICOM standard. Another *middleware* solution proposed for use in health data environments is *CORBA* (*Common Object Request Broker Architecture*), applied in the work of Murshed et al. [29], Waluyo et al. [39] and Blobel and Holena [7].

Hsieh et al. [14] uses a series of technologies, systems, and standards to characterize a *middleware* solution, namely: a *Single Sign-On Server* (SSOS) in a Health Information Enterprise Portal (HEIP) and a health information system (HIS). *National Taiwan University Hospital* (NTUH) received this above-cited set of solutions.

Finally, a series of *middleware* platforms are in the analyzed works, including *MiThrilNGN*, *MyHearth* and *X73uHealth* [17], *ANGELAH* (AssistiNG ELders At Home) - proposed in Taleb et al. [38] and *HYDRA*, a solution that allows the integration of different devices into applications through a simple services interface.

The analysis of these works showed us that there are already *middleware* platforms proposed or developed for the healthcare area. However, these solutions are not entirely flexible and applicable in different contexts and, therefore, do not meet all the essential requirements for *middleware* platforms for the healthcare area.

Given this, the opportunity to create *middleware* for health data that uses open standards and facilitates integration between *web* systems, such as $HL7\ FHIR$, for integration and interoperability between different solutions health center is still open. Therefore, we propose and develop MiSIS: A Middleware based on the HL7 FHIR Standard for Health Information Systems, described in Sect. 3.

3 MiSIS: A Middleware Based on the HL7 FHIR Standard for Healthcare Information Systems

In this section, MiSIS, a Middleware Platform based on the HL7 FHIR Standard for Health Information Systems, will be described and its implementation and evaluation process. MiSIS aims to promote interoperability between different applications and health information systems, making details of their implementations invisible to developers, focusing only on the communication layer, and enabling integration between heterogeneous solutions.

3.1 Solution Requirements

The main challenges for developing middleware for health information systems consist of the following: Provide solutions that deal with the heterogeneity of data present in the context of health and its applications [11,21,31]; Support devices, systems, and sensors that communicate through different means [17,25]; securely provides reliable data, especially considering the degree of sensitivity of health data [4,31].

In this sense, based on these challenges and the state-of-the-art analysis carried out by Malaquias et al. [27] that, below, are described the main requirements that MiSIS must cover:

- **Promote transparent integration of heterogeneous data:** MiSIS should provide a means of treating heterogeneous data with high abstraction, flexibility, and dynamicity in different contexts and environments. Enabling different types of health data to be treated and integrated transparently and with a high level of abstraction makes it possible to centralize health data, facilitating access and cross-referencing between this information.
- **Enable scalability:** MiSIS must provide means by which applications can scale in terms of connected devices and systems, as well as the volume of data transmitted, but without compromising quality requirements, such as reliability, fault tolerance, real-time synchronization, and response;
- **Provide security and privacy:** MiSIS must guarantee the protection and privacy of data used by doctors, patients, and other users using user access control strategies.

Software Architecture strategies were defined and described in the 3.2 subsection to address these requirements and deal with the above challenges.

3.2 Solution Architecture

The *Middleware* proposed in this work supports the integration between different health systems and applications regardless of the type of data managed by each of them. Figure 1 illustrates the role of the proposed solution, where *software* component interconnects different digital solutions aimed at the health area. That software component is responsible for integrating health data from these solutions and making them available to other client applications that use the data, which are, in turn, integrated.

As illustrated, the solutions in Fig. 1 are varied, being systems that deal with patient data, monitoring vital signs, exam management, hospitalization, and bed management. These solutions are examples of health information systems focused on solving specific problems in the health context. However, they initially do not interoperate among them, and there is no integration between their data. Therefore, to enable transparent integration between these systems, an architecture was defined for the proposed solution based mainly on the following items:

- *middleware*, which is responsible for promoting data integration in a transparent, secure, and easy-to-implement manner;
- And the clients, which consist of applications that communicate with the *middleware* component through $HTTP$ requests, sending data to be integrated or receiving data already integrated in a structured way.

Figure 2 illustrates the view of MiSIS modules, describing the modules that organize each component of the *solution*, as well as the format and standard ($HL7\ FHIR$) that must be followed in the process of integration and communication between health applications.

As a solution based on *WEB* systems, MiSIS must be made available through an application server. Figure 3 illustrates the implementation diagram for MiSIS Fig. 3, and, following this structure, authors carried out two proofs of the concept of the solution.

3.3 Solution Development

MiSIS was developed using the Java Platform *Enterprise Edition* (Java EE), together with the *Spring Framework* [18] tools for developing solutions based on *Web*. Data from client applications integrated by *Middleware* is stored in a $MongoDB$, non-relational and open source database [30]. The open source library $HAPI\ FHIR$ carries out the conversion and interpretation of data to $HL7\ FHIR$, and the data is made available through RESTful API services in $JSON$ and XML formats.

The proposal integrates data from five consolidated but different Health Information Systems used daily by healthcare professionals to carry out the Proofs of Concept and performance tests. These are the considered information systems: the Clinical Data Platform - SigSaúde; the Remote Assistance Platform - PAR; the Beds - Information system based on *web* for managing ICU beds; the SUVEPI System; and the COVID-19 Examination System.

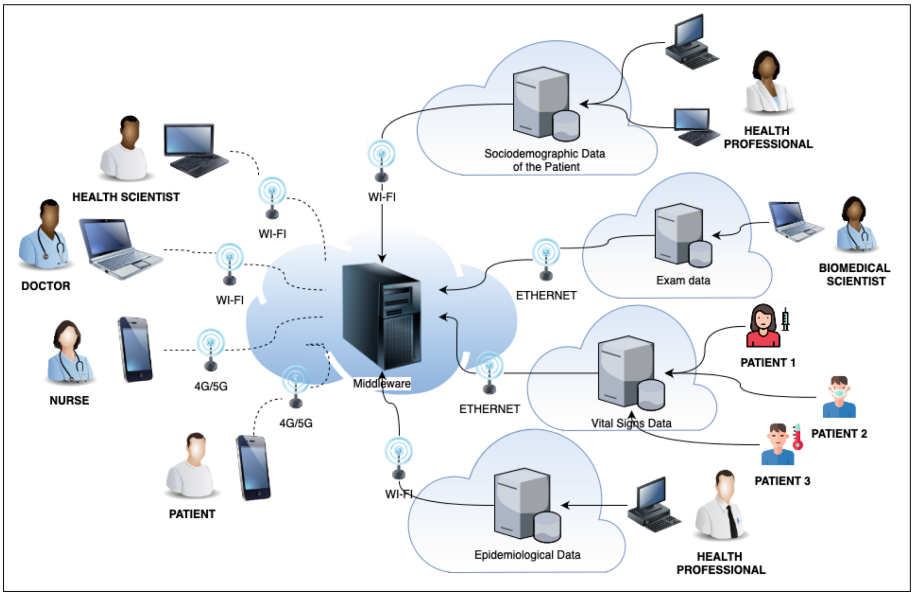

Fig. 1. Conceptual representation of MiSIS: A Middleware Based on the *HL7 FHIR* Standard for Health Information Systems.

4 Studies on the Use and Evaluation of MiSIS

According to the state-of-the-art review carried out in the work of Malaquias et al., [27], *Middleware* Platforms for healthcare systems are evaluated mainly from the point of view of performance, reliability, and direct and indirect costs. The proposal uses two ways to evaluate MiSIS, namely:

- By carrying out two proofs of concept, which consisted of the development of two new applications that consume integrated data and aimed to evaluate cost characteristics of implementing new solutions with the data made available from *middleware*;
- From the execution of stress tests using Apache JMeter, a solution designed to measure the performance of solutions based on the execution of functional behaviors at scale enabled data collection related to the performance and reliability of *middleware*.

Therefore, the details of these evaluations, the methodology used, and the following sections discuss obtained results.

4.1 Analysis of Proofs of Concept

In the context of this work, two new Health Information Systems were developed, one of them being *eHealth* and the other *mHealth*, being *Integrated Health*

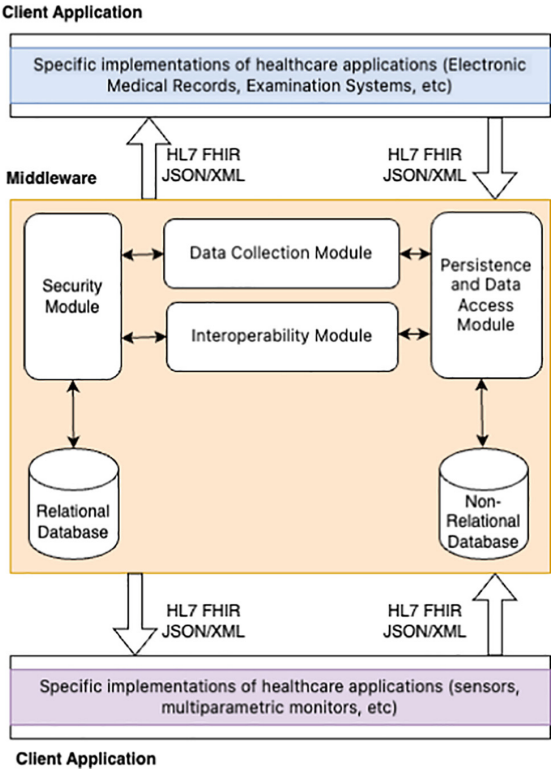

Fig. 2. View of solution modules and the way they communicate.

Application (IHA) and the application "My Medical record", respectively. From the development of these applications, it was possible to analyze whether MiSIS meets the requirements of promoting transparent integration of heterogeneous data and providing security and privacy.

Methodology. Initially, the IHA application was developed, a *eHealth* application based on *web* that aims to present a patient's health data in a unified interface and, for this purpose, performs queries on the RESTful API provided by MiSIS. This application was developed by a single developer, using the Java EE Platform combined with *Spring Framework* solutions. In the experience of developing the application, we noticed the productivity gain when consulting the already integrated data since this application displays four different types of health data (personal data, hospitalization data, exam data, and health data). Completely independent systems collected vital signs, and if they were not connected and integrated from MiSIS, the only way to obtain their data would be if each of them had communication interfaces available, requiring a more significant effort of time and costs for the development of integration.

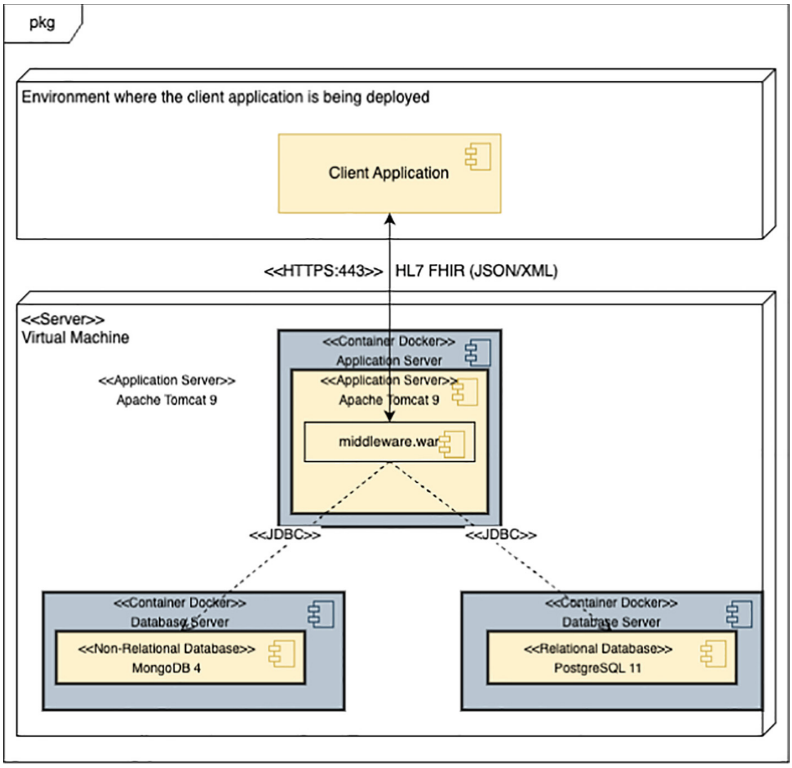

Fig. 3. MiSIS Deployment Diagram.

Secondly, the "My Medical Record" application was developed, a *mHealth* application that aims to facilitate access to a patient's historical health data based on unique identifiers such as CPF and CNS. A team of 3 developers developed this application using the Flutter framework with Dart programming language. To evaluate the experience that developers had in the process of developing a new health application using data integrated from *Middleware* based on the *HL7 FHIR* Standard for Health Information Systems, a *online* questionnaire, built on the *Google Forms* Platform, and can be accessed through the following address: https://forms.gle/rF2BPjUdzynLqXXa6. The questionnaire made available consisted of the following questions:

- **How do you assess your degree of seniority in programming?** This question was multiple choice and aimed to identify the degree of programming experience of each developer based on three options: Junior, Full, and Senior;
- **How many hours did you spend developing the application?** This question aimed to identify the total number of hours spent developing the application;

- **How would you effectively divide the hours allocated to solving technical problems (technology research) and the amount allocated to working on integration activities?**
  In this question, the objective was to understand the number of hours that each developer needed to work on *middleware* integration and data consumption activities effectively;
- **Do you believe using the *JSON* format to exchange data influenced the development process? As?**
  The objective of this question is to investigate whether the use of an already disseminated standard for exchanging information between *web* systems influenced the development process;
- **How do you evaluate, pointing out positive and negative points, the experience in using *Middleware* in developing an application that involves integration between health systems?**
  This question aims to identify the positives and negatives of the programming experience consuming *middleware* data from the developers' point of view.

Results Obtained. At the end of the application's development, the questionnaire was available so that team members who worked on the development of the "My Medical Record" mobile application could answer it. It is essential to highlight that the developer who worked on implementing the "IHA" application did not answer the questionnaire since he was the author of this work and developer of MiSIS. The responses of each of the developers to the questionnaire, which are also reported in general below:

- **How do you assess your degree of seniority in programming?**
  When answering this question, the three developers classified themselves as "Junior Developers", pointing out their low level of experience and indicating the need for more time required to implement the application;
- **How many hours did you spend developing the application?**
  Table 1 shows that each of the three developers reported having spent 10, 7, and 6 h, respectively, totaling 23 h to develop the application;

Table 1. Number of hours worked by each developer on the development of the 'My Medical Record' app.

Developer	Work hours
A	10
B	7
C	6
Total	23

– **How would you effectively divide the hours allocated to solving technical problems (technology research) and the amount allocated to working on integration activities?**
Table 2 shows how developers divided their working hours according to demand needs. Each of them stated that they spent only 3 h of work on integration activities and the remaining hours researching technical questions more related to the Flutter programming language, pointing out that the difficulties encountered because they had little experience in programming were the factor that made the development process more complex compared to the difficulties of the health data integration process, as they spent fewer hours working on this process;

Table 2. Number of hours worked by each developer on specific issues of the Flutter programming language and integration

Developer	Hours spent researching inquiries regarding Flutter	Hours spent working on integration
A	7	3
B	4	3
C	3	3
Total	14	9

– **Do you believe using the *JSON* format to exchange data influenced the development process? As?**
Unanimously, all developers confirmed that the use of the *JSON* format had a positive influence on the development process, reporting facilities such as the availability of examples of *FHIR online* resources in the *JSON* format and also previous experiences of integrations using this format;
– **How do you evaluate, pointing out positive and negative points, the experience in using *Middleware* in developing an application that involves integration between health systems?**
As positive points, the developers reported the ease of using data from different databases. They integrated from *middleware* using already known and well-documented standards. In addition, if the integration had not been done with *middleware*, the application would necessarily have to connect several different communication services to obtain the same data. As a negative point, the developers noted the need for a specific mapping for the *FHIR* resources since the library that handles *HL7 FHIR* for the Flutter programming language presented errors during the development process.

In general, the developers who worked on the applications resulting from the Proofs of Concepts attested that *middleware* meets the requirement of promoting transparent integration of heterogeneous data since two solutions that use

health data were developed quickly due to the use of *middleware*; it also meets the requirement to provide security and privacy, since to consume health data, applications needed to have a valid access key; and the requirement to provide support for "*plug-and-play*" data collection, considering that to consume the data made available by *middleware*'s RESTful API, applications only had to implement strategies to convert the data received in $HL7\ FHIR$. Furthermore, the developers involved in implementing the new applications pointed out the ease brought by the use of *Middleware* based on the $HL7\ FHIR$ Standard in the process of integration and consumption of health data, showing the possibility of cost reduction development and implementation using *Middleware*.

4.2 Performance Test Analysis

In order to analyze the scalability, availability, and reliability requirements of MiSIS, the proposal carried out stress tests after the development and deployment of the solution on a server available in a private cloud. To carry out the tests, Apache JMeter, a solution that aims to measure the performance of systems based on the execution of functional behaviors at scale, was used to collect data related to the performance and reliability of the tested solution. The test environment configured to carry out the tests, as well as the methodology used to carry out the tests and the results obtained, will be discussed in the Subsects. 4.2, 4.2 and 4.2, respectively.

Test Environment Specification: The environment used to carry out the tests consisted of a virtual machine in a private cloud, which deployed the MiSIS application, and a personal computer with Apache JMeter to simulate client applications that consume integrated data from MiSIS. Below is a description of the configurations of the virtual machine and computer with Apache.

- **Virtual Machine for deploying MiSIS**:
  - **Operating System**: Ubuntu 20.04.1 LTS
  - **RAM memory**: 8Gb
  - **Number of vCPUs**: 8
  - **Disk space**: 20Gb
- **Personal computer to run the client healthcare application simulator**:
  - **Operating System**: macOS Big Sur 11.2.2
  - **RAM memory**: 8Gb
  - **Number of vCPUs**: 8
  - **Disk space**: 512Gb SSD

Figure 4 shows the deployment diagram of the environment used to carry out the performance tests, illustrating the environments with the installed application and the form of communication between the client simulation application and MiSIS that is carried out through $HTTP$ requests to the RESTful API, which returns data to clients in the $HL7\ FHIR$ standard and $JSON$ and XML formats.

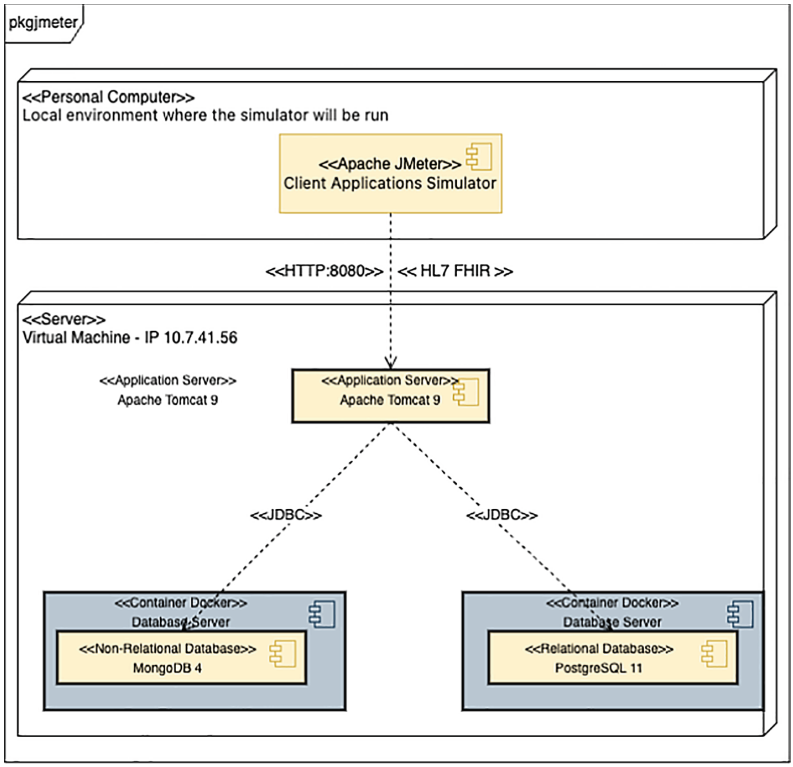

Fig. 4. Diagram of deploy environment performance testing

Methodology: After implementing MiSIS for the virtual machine, the proposal proceeds with data integration from 5 consolidated health systems that are in a productive environment, namely the Clinical Data Platform (SigSaúde), the Remote Assistance Platform (PAR), the COVID-19 Exam Management System, the Bed Management System; and the System for Management of COVID-19 Cases in the State of Rio Grande do Norte (SUVEPI). After processing and integrating data from these systems, the *middleware* database totaled more than 417 thousand health data from 3 different $HL7\ FHIR$ resources, namely Patients, Observations, and Encounters.

The experiments consisted of 5 sets of tests to evaluate the performance of the integrated data search for each type of $FHIR$ resource available. Therefore, tests were carried out based on the number of client applications that simultaneously accessed *middleware* in order to evaluate latency, reliability (error rate), and *throughput* characteristics in data transmission. The number of simulated client applications was 5, 25, 50, 75, and 100 for each of the test batteries and data on request processing time (average, minimum, and maximum latency), standard deviation, error rate, and *throughput* (i.e., the number of requests made every second).

Results Obtained: The results related to the processing time of each request from each of the test batteries are present in Table 3. The minimum latency of requests varies from 196 to 215 milliseconds, showing low variation according to the number of clients that perform queries to the *middleware* API. Authors considered these values low, which indicates that, under ideal infrastructure conditions, request processing can be quite fast. Such an archive is significant as it demonstrates the efficiency of MiSIS.

The average and maximum latency values increased as the number of simulated clients increased. It shows that, quite possibly, the configurations of the environment used for deploying the *middleware* were not ideal for its best functioning, and this is even clearer in the test batteries that simulated more than 50 clients, which showed response from 1745 ms. Furthermore, the high value of the standard deviation in all batteries indicates that there is a difference in the processing time between different requests, and this was due to variations in load on the server during the test batteries (an expected behavior, considering that test batteries queried different types of data).

Table 3. Minimum, mean, maximum, and standard deviation values for the response time (latency) to requests made to the Middleware in each test batch.

Clients	Min (mili)	Mean (mili)	Max (mili)	Standard deviation
5	196	538	1408	213.82
25	202	973	2232	435.94
50	191	1745	3617	881.4
75	215	2654	5395	1267.56
100	215	3487	7014	1870.02

Table 4 shows that in all tests, the percentage of error (that is, requests that did not return *STATUS* 200 of the *HTTP* protocol) was below 2%. The errors presented were server errors caused by concurrency problems, pointing out weaknesses in the *hardware* where *middleware* was installed. Finally, the *throughput* rate was 5.6 requests made per second for the scenario with five clients in the test batteries, but from the moment 25 or more clients were simulated, this rate was between 13.4 and 14.7 requests performed every second. This data is also an indicator that, under ideal computational infrastructure conditions, the *middleware* can process and deliver data quickly, as noticed that as more requests come at the same time, the *middleware* textitmiddleware needed more time to process and return information to clients.

Finally, it is essential to clarify that the results obtained in the simulations and the discussions presented based on them are directly related and influenced by the configurations of the testing environment used to deploy the *middleware* and the simulator. However, the data shows us that the MiSIS presented in this work satisfactorily meets the requirement of providing scalability, as long as it

is installed in an environment with better sized *hardware*, being able to process more requests per second and making integrated health data available in an increasingly shorter time.

Table 4. Error rate obtained and the number of requests per second throughput performed in each test batch executed.

Clients	Error Rate	Throughput (req/sec)
5	0%	5.6
25	0.71%	13.4
50	1.03%	14.8
75	2%	14.4
100	1.29%	14.7

5 Conclusions and Future Work

As presented in Sect. 1, there is a strong trend in the development of technological solutions for the healthcare sector. This is primarily aimed at facilitating access to and management of health data by those involved in healthcare service provision (patients, healthcare professionals, managers, or family members), in order to optimize processes, enhance service quality, and reduce costs. Consequently, numerous technological solutions are developed for the healthcare context and often used to address specific problems, resulting in a heterogeneous ecosystem of solutions that are not integrated with each other.

Therefore, Sect. 2 introduced fundamental concepts related to this problem, such as Health Information Systems, *eHealth*, *mHealth*, *Middlewares*, and the $HL7FHIR$ Standards, which were necessary for understanding this work and formed the basis for the creation of MiSIS, a Middleware based on the $HL7FHIR$ Standard for Health Information Systems, detailed in Sect. 3. Section 3 described the requirements of MiSIS, as well as the proposed software architecture to meet these requirements and the current state of the implemented solution.

Furthermore, Sect. 4 outlined the studies conducted to validate MiSIS, which were carried out in two main ways: (i) through Proof of Concepts, involving the implementation of two new Health Systems that consume integrated data from MiSIS, and (ii) through performance tests, aiming to analyze the scalability, availability, and reliability requirements of MiSIS. Overall, the studies indicated that MiSIS effectively meets the requirements related to facilitating integration among heterogeneous healthcare systems, thereby reducing integration implementation costs. On the other hand, performance tests indicated that under ideal conditions for deployment, MiSIS can process and deliver data more quickly.

Finally, future work includes: Conducting further availability tests on infrastructures better than the one where MiSIS was deployed in the context of this work; Deploying MiSIS in a public cloud solution, such as *Amazon Web Services*, to evaluate scalability requirements and real-time data provisioning; Validating MiSIS through case studies in real scenarios involving an ecosystem of healthcare information systems with heterogeneous data, involving healthcare professionals in the validation process (e.g., in a hospital).

Acknowledgement. This study was funded by its authors.

Disclosure of Interests. The authors have no competing interests to declare that they are relevant to the content of this article.

References

1. Almadani, B., Saeed, B., Alroubaiy, A.: Healthcare systems integration using real time publish subscribe (RTPS) middleware. Comput. Electr. Eng. **50**, 67–78 (2016)
2. Alonso, L., Barbarán, J., Chen, J., Díaz, M., Llopis, L., Rubio, B.: Middleware and communication technologies for structural health monitoring of critical infrastructures: a survey. Comput. Stand. Interf. **56**, 83–100 (2018)
3. AlZghoul, M.M., Al-Taee, M.A., Al-Taee, A.M.: Towards nationwide electronic health record system in Jordan. In: 2016 13th International Multi-Conference on Systems, Signals & Devices (SSD). pp. 650–655. IEEE (2016)
4. Arunachalan, B., Light, J.: Middleware architecture for patient care data transmission using wireless networks. In: Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing, pp. 182–185 (2007)
5. Barroca Filho, I., et al.: Development of a health dashboard for an electronic health record system. In: 2020 20th International Conference on Computational Science and Its Applications (ICCSA), pp. 16–22. IEEE (2020)
6. Bass, L., Clements, P., Kazman, R.: Software architecture in practice (2013)
7. Blobel, B., Holena, M.: Comparing middleware concepts for advanced healthcare system architectures. Int. J. Med. Informatics **46**(2), 69–85 (1997)
8. Chaudhry, J., Valli, C., Crowley, M., Haass, J., Roberts, P.: Postcode middleware for post-market surveillance of medical devices for cyber security in medical and healthcare sector in Australia. In: 2018 12th International Symposium on Medical Information and Communication Technology (ISMICT), pp. 1–10. IEEE (2018)
9. Eysenbach, G.: What is e-health? J. Med. Internet Res. **3**(2), e20 (2001). https://doi.org/10.2196/jmir.3.2.e20, http://www.jmir.org/2001/2/e20/
10. de Fátima Marin, H.: Sistemas de informação em saúde: considerações gerais. J. Health Inform. **2**(1) (2010)
11. Ferrara, F.M.: The standard 'healthcare information systems architecture'and the dhe middleware. Int. J. Med. Informatics **52**(1–3), 39–51 (1998)
12. Foundation, H.F.: Hl7 fhir foundation about page (2021). http://fhir.org/about.html
13. Free, C., Phillips, G., Felix, L., Galli, L., Patel, V., Edwards, P.: The effectiveness of m-health technologies for improving health and health services: a systematic review protocol. BMC. Res. Notes **3**(1), 1–7 (2010)

14. Hsieh, S.L., et al.: An integrated healthcare enterprise information portal and healthcare information system framework. In: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4731–4734. IEEE (2006)
15. Iakovidis, I.: Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in Europe. Int. J. Med. Informatics **52**(1–3), 105–115 (1998)
16. Jayaratne, M., Nallaperuma, D., De Silva, D., Alahakoon, D., Devitt, B., Webster, K.E., Chilamkurti, N.: A data integration platform for patient-centered e-healthcare and clinical decision support. Futur. Gener. Comput. Syst. **92**, 996–1008 (2019)
17. Ji, Z., Ganchev, I., O'Droma, M., Zhang, X., Zhang, X.: A cloud-based x73 ubiquitous mobile healthcare system: design and implementation. Sci. World J. **2014** (2014)
18. Johnson, R., et al.: Spring framework documentation (2020)
19. Kallepalli, V.N., Ehikioya, S.A., Camorlinga, S., Rueda, J.A.: Security middleware infrastructure for DICOM images in health information systems. J. Digit. Imaging **16**(4), 356–364 (2003)
20. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering-a systematic literature review. Inf. Softw. Technol. **51**(1), 7–15 (2009)
21. Kliem, A., Boelke, A., Grohnert, A., Traeder, N.: Self-adaptive middleware for ubiquitous medical device integration. In: 2014 IEEE 16th international conference on e-health networking, applications and services (Healthcom), pp. 298–304. IEEE (2014)
22. Liu, L., Huang, Q.: An extensible hl7 middleware for heterogeneous healthcare information exchange. In: 2012 5th International Conference on BioMedical Engineering and Informatics, pp. 1045–1048. IEEE (2012)
23. Liu, X., Ma, L., Liu, Y.: A middleware-based implementation for data integration of remote devices. In: 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 219–224. IEEE (2012)
24. Lomotey, R.K., Nilson, J., Mulder, K., Wittmeier, K., Schachter, C., Deters, R.: Mobile medical data synchronization on cloud-powered middleware platform. IEEE Trans. Serv. Comput. **9**(5), 757–770 (2016)
25. Lomotey, R.K., et al.: Using cloud-based middleware to enable mobile medical data management. In: 2016 IEEE International Conference on Mobile Services (MS), pp. 142–149. IEEE (2016)
26. Madureira, P., Cardoso, N., Sousa, F., Moreira, W.: My-aha: middleware platform to sustain active and healthy ageing. In: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 21–26. IEEE (2019)
27. Malaquias, R.S., Filho, I.M.B.: Middleware for healthcare systems: a systematic mapping. In: Gervasi, O., et al. (eds.) ICCSA 2021. LNCS, vol. 12957, pp. 394–409. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87013-3_30
28. de Mattos, W.D., Gondim, P.R.: M-health solutions using 5g networks and m2m communications. IT Professional **18**(3), 24–29 (2016)
29. Murshed, A.N., Almansoori, W., Xylogiannopoulos, K.F., Elzohbi, M., Alhajj, R., Rokne, J.: Developing an efficient health clinical application: Iiop distributed objects framework. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 759–764. IEEE (2012)

30. Pollack, M., Risberg, T., Gierke, O., Leau, C., Brisbin, J.: Spring data mongodb-reference documentation (2011)
31. Prados-Suarez, B., Molina, C., Peña-Yañez, C.: Providing an integrated access to EHR using electronic health records aggregators. Studies in Health Technol. Informatics **270**, 402–406 (2020)
32. Rodriguez, M.D., Favela, J.: An agent middleware for ubiquitous computing in healthcare. In: Sordo, M., Vaidya, S., Jain, L.C. (eds.) Advanced Computational Intelligence Paradigms in Healthcare-3, pp. 117–149. Springer, Cham (2008). https://doi.org/10.1007/978-3-540-77662-8_6
33. Ryan, A., Eklund, P.W.: The health service bus: an architecture and case study in achieving interoperability in healthcare. Stud Health Technol. Inform. (2010)
34. Shand, B., Rashbass, J.: Security for middleware extensions: event meta-data for enforcing security policy. In: Proceedings of the 2008 Workshop on Middleware Security, pp. 31–33 (2008)
35. Silva, B.M., Rodrigues, J.J., de la Torre Díez, I., López-Coronado, M., Saleem, K.: Mobile-health: a review of current state in 2015. J. Biomed. Inform. **56**, 265–272 (2015)
36. Singh, J., Bacon, J.: Managing health information flows with a reconfigurable component-based middleware. In: 2011 IEEE 12th International Conference on Mobile Data Management, vol. 2, pp. 52–54. IEEE (2011)
37. Singh, J., Vargas, L., Bacon, J., Moody, K.: Policy-based information sharing in publish/subscribe middleware. In: 2008 IEEE Workshop on Policies for Distributed Systems and Networks, pp. 137–144. IEEE (2008)
38. Taleb, T., Bottazzi, D., Guizani, M., Nait-Charif, H.: Angelah: a framework for assisting elders at home. IEEE J. Sel. Areas Commun. **27**(4), 480–494 (2009)
39. Waluyo, A.B., Pek, I., Chen, X., Yeoh, W.S.: Design and evaluation of lightweight middleware for personal wireless body area network. Pers. Ubiquit. Comput. **13**(7), 509–525 (2009)
40. Zhang, W., Thurow, K., Stoll, R.: A knowledge-based telemonitoring platform for application in remote healthcare. Int. J. Comput. Commun. Control **9**(5), 644–654 (2014)
41. Zuehlke, P., Li, J., Talaei-Khoei, A., Ray, P.: A functional specification for mobile ehealth (mhealth) systems. In: 2009 11th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 74–78. IEEE (2009)