Functionalities as a Service - An Approach to Conciliate Interoperability and Data Reduction in E-Health

João Pedro de S. J. da Costa[1] **, Mário A. R. Dantas**[1] **,**
José Maria N. David[1] **, Fernando de Almeida Silva**[1]

[1]Departament of Computer Science, Federal University of Juiz de Fora (UFJF)
Postal Code 20.010 – 36036-900 – Juiz de Fora – MG – Brazil

joao.costa@estudante.ufjf.br, mario.dantas@ice.ufjf.br

jose.david@ufjf.br, fernandoalmeida.silva@ufjf.br

Abstract. *Interoperability and data reduction have been proven beneficial to health and medical applications that deal with large datasets. Still, the conflicts between these qualities turned out to be a problem for their conciliation. This paper presents an edge-fog-cloud architecture that offers functionalities as a service. These functionalities can guarantee certain qualities depending on the necessities of the client, such as, in our case, interoperability and data reduction. With the use of context simulators, we found that it was possible to significantly increase the output of data delivered to the servers, and decrease the size of the data that transitions in the network and is stored in the servers' databases, without interfering with the syntactic interoperability.*

1. Introduction

With the increase in health service costs and the shortage of professionals, new approaches to providing care have been studied. The use of information technologies as support for health services was one of the solutions found. Eysenbach et al. (2001) define Electronic Health Services (e-Health) as the intersection between medical informatics, public health and business. These services encompass activities and health information that are delivered through or enhanced by the Internet and related technologies. Another problem that health services need to deal with is the growth in demand for these services. AbuKhusa, Mohamed and Al-Jaroodi (2012) exemplify cloud computing as one of the technologies that could help e-Health deal with this growth. His reasoning comes from the fact that cloud computing provides a strong infrastructure and facilitates the supply of services over the Internet.

Another possibility, discussed by Scarpato et al. (2017), is the use of the Internet of Things (IoT). IoT devices provide a vast amount of information to be gathered, stored and analyzed for data analysis processes. The reason is that they are able to collect and share large amounts of data directly with other devices through the cloud environment. Interoperability is another way that healthcare applications gain access to large amounts of data. Shull (2019) addresses about the importance of interoperability for e-Health. Unlike the technologies mentioned above, interoperability is a quality that applications can have. Applications that are interoperable can share data and information with each other, increasing the scope of data which the application works with and, consequently, improving the accuracy of the results of its processing.

However, managing and transporting large amounts of data are challenges for cloud computing architectures. This is due to the fact that these activities generate high maintenance costs. The use of data reduction techniques is one of the ways that these architectures have been found to deal with these challenges. Using these techniques, it is possible to reduce the size of the data to be transported and/or stored. However, due to the fact that they modify the original data, the use of data reduction techniques makes interoperability between applications difficult or even impossible. Decreasing the scope of data that the application has access to and, on the other hand, decreasing the precision of the results of its processing.

Previous studies, that discussed interoperability and the use of reduction techniques in health applications, as far as we researched, did not conciliate both these themes. While Rubí and Gondim (2020) and Rinty, Prodhan and Rahman (2022) sought solutions to challenges related to interoperability, Kahdim and Manaa (2022) and Idrees and Idrees (2022) discuss data reduction for health applications.

This study proposed to develop the prototype of an edge-fog-cloud architecture, for healthcare applications. We use data reduction techniques to reduce the amount of data that travels through the network, without affecting the quality of syntactic interoperability of the applications that are using the architecture.

This document is organized into six sections. Works related to the theme are presented in Section 2. The materials and methods related to the research proposal are presented in sections 3 to 4. Section 5 presents the test approaches and their results. Final considerations and future works are presented in Section 6.

2. Related Work

One of our main goals is lossless data reduction. Idrees and Idrees (2022) proposed a data resilient method, a method used to reduce the amount of data sent to the cloud by Electroencephalogram (EEG) IoT devices. In their project, the data to be reduced is produced by a health monitoring system (SMS). These systems use physical devices attached to the patient's body to collect data and send them for processing over the wired or wireless network. The 3 essential parts of the system, used by Idrees and Idrees (2022), are medical sensor nodes, a gateway and back-end servers. Medical sensor nodes consist of wearable sensors or sensors implantable in the human body. These sensors are used to collect various data related to the patient's physical condition. The gateway connecting the medical sensors to the cloud servers is the critical point of the structure. The data generated by the sensors would not be accessible if the gateway stopped working properly. The back-end servers included cloud servers and health servers. The cloud servers were responsible for gathering, processing and transmitting the data to the health servers. Meanwhile, the health servers analyze the medical data and monitor the patient's physical condition.

The reduction technique that was used consisted of two steps: clustering and compression. The chosen algorithms were the hierarchical Clustering Algorithm and the Huffman Coding Algorithm. Periodically, sensor data was collected on the smart fog gateway. The data were grouped into several clusters according to the similarities found by the Hierarchical Clustering Algorithm. The Huffman Coding Algorithm was applied on each cluster to compress the sent data. The work of Idrees and Idrees (2022) was able to contribute with a new lossless treatment procedure to reduce data produced by EEG sensors.

The solution employs compression in the architecture fog gateway. Extensive experiments were performed, using the Python programming language, on real data collected from patients through EEG sensors.

Kahdim and Manaa (2022) designed a different method of data compression for healthcare applications. Data reduction was used to decrease the amount of data sent from the IoT sensor level to the fog level. The physical components used for the tests were: a temperature sensor, a heart rate sensor, an oxygen sensor, a Raspberry Pi and a desktop personal computer acting as the fog server. Raspberry Pi is a powerful, inexpensive and adaptable mini-computer. Due to its previously mentioned characteristics, as well as the fact that it supports a wide variety of input and output ports, it was chosen as the main device of the solution. It is on the Raspberry Pi where data is aggregated and compressed, in addition to controlling and monitoring the sensors.

Like Idrees and Idrees (2022), Python was the programming language chosen by Kahdim and Manaa (2022). The authors explain that the motivation for choosing this language is the fact that it is adaptable, powerful and useful for different types of tasks. The data flow starts with the reading of the data generated by the IoT sensors. After the data has been aggregated on the Raspberry Pi, the Zstandard algorithm is applied. This compressed data is sent to a fog server. At the fog level, data is decompressed, analyzed, and ultimately stored. As a conclusion, Kahdim and Manaa (2022) state that, with the use of Zstandard compression, the proposed system could significantly reduce the amount of medical data sent by the Raspberry Pi to the fog server.

Interoperability between applications is another critical aspect of our project. Thinking about communication between machines and legacy healthcare applications, Rubí and Gondim (2020) developed a platform that allows these applications to interoperate. Considered as an Internet of Medical Things platform, it made use of OpenEHR and semantic sensor network semantics. OpenEHR is a standard for modeling electronic health records. With the use of OWL ontologies based on the OpenEHR standard, it is possible to standardize the semantics of data shared on the platform.

The solution consists of a storage server and a storage controller mediating communications between legacy applications and new applications. Legacy applications are directly linked to the storage server, this server has a controller that communicates with the network service provider. The storage server translates the data produced by legacy applications to the OpenEHR standard. Doing so, enables that data to be used by new applications that are based on this standard. The opposite is also done, the data in the OpenEHR standard, sent by new applications, are translated to the semantics of the legacy applications. The controller is also responsible for updating the terms on the storage server. Rubí and Gondim (2020) presented as their main contribution an extended OpenEHR ontology, which aligns the domain of standardized applications in the OpenEHR model with legacy applications.

As explained earlier, interoperability makes it possible for applications to share data with each other, however, it is a quality that faces challenges to be implemented and used. Rinty, Prodhan and Rahman (2022) discuss the fact that developing countries have difficulty in applying interoperability in e-Health. Aiming to solve this problem, they proposed an improved e-Health framework for developing countries. The proposal uses distributed memory and loosely coupled heterogeneous data manipulation mechanisms to achieve its objective. The standard for data structure proposed by Health Level Seven International (HL7) was chosen for the framework. The HL7 is a set of international norms, standards and definitions used in transfer and communication between healthcare practitioners. It was chosen because it is one of the most important design models among health data standards.

The proposal consists of two modules controlling data access and transition, the Health System Administrator and the Local Administrator. The module called Health System Administrator communicates directly with the end user. It receives and distributes requests made by end users among the different instances of the Local Administrator, in addition, the Health System Administrator controls the limit of concurrent searches to prevent overloading. Each Electronic Health Record (EHR) server has its own instance of Local Administrator. The Local Administrator is responsible for parsing requests and translating them into executable statements that the EHR server understands. Rinty, Prodhan and Rahman (2022) point to this configuration as what enables low coupling and interoperability of medical data and the search interface.

It was noted that, up until where we researched, the works, that are related to our proposal, haven't made an effort to conciliate interoperability between applications and data reduction techniques. Idrees and Idrees (2022) address lossless data reduction for E-Health, however, they do not address the implications that such an approach would have on the communication with external applications. The same is observed in the work of Kahdim and Manaa (2022). They presented a data reduction method for healthcare applications that also uses data compression. The main difference between the two works is the compression algorithm and data flow that were applied. As for interoperability, both proposals do not discuss it.

In contrast, Rubí and Gondim (2020) and Rinty, Prodhan and Rahman (2022) focus on the issues of interoperability. Rubí and Gondim (2020) discussed and presented a solution for the lack of interoperability with legacy health applications. Rinty, Prodhan and Rahman's (2022) solution addresses the interoperability difficulties of health applications in developing countries. Notwithstanding, both proposals do not take into account cases in which applications use data reduction techniques.

3. Research Question

Taking into account the information in Section 2, it is clear that interoperability and data reduction techniques are advantageous for the functioning and efficiency of health applications. Interoperability gives applications the possibility to access a much larger amount of data, compared to the data they can capture on their own. Whereas, Data reduction techniques significantly reduce traffic and data storage costs. Although, the use of data reduction techniques can generate conflicts with the application's ability to interoperate. Depending on the type of data reduction technique applied, as shown in Figure 1 and Figure 2, different obstacles can inhibit the ability of applications to share significant data with each other.

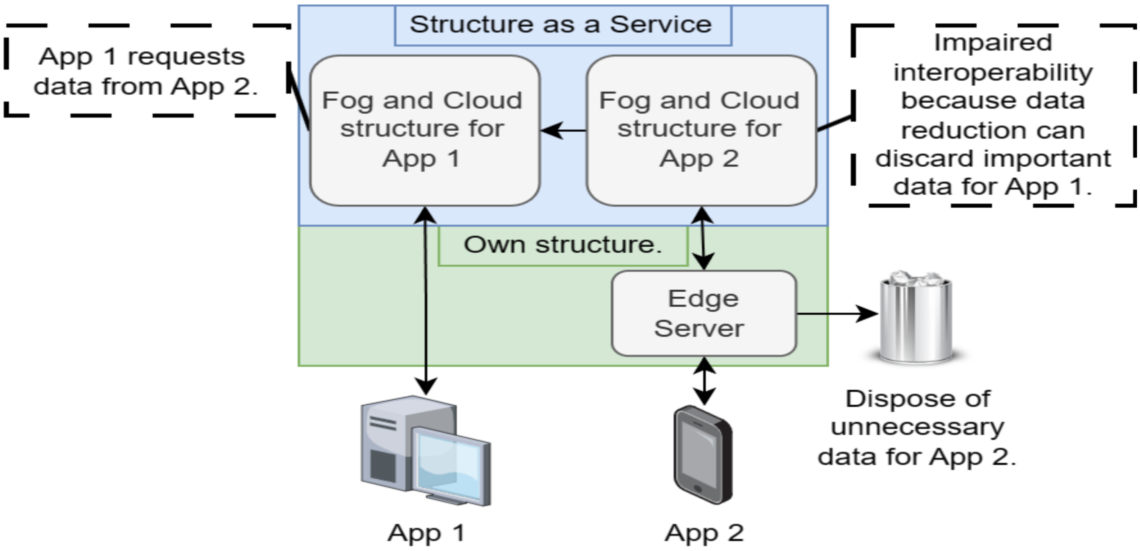

Figure 1. Diagram exemplifying how discarding data can hurt interoperability between applications.

When the application discards any data considered insignificant for its purposes, it's feasible to say that important data for other applications could be discarded. It would only be possible to prevent this data from being discarded, if the application that captured it, and wants to reduce it, knew what data other applications need. However, the greater the diversity of applications with which it communicates, the less data will be discarded. Thus, decreasing the impact of the reduction and increasing the amount of unnecessary information, for its main functionality, that the application would need to retain. Illustrated in Figure 2, another possibility is the use of techniques that modify the format of the data.

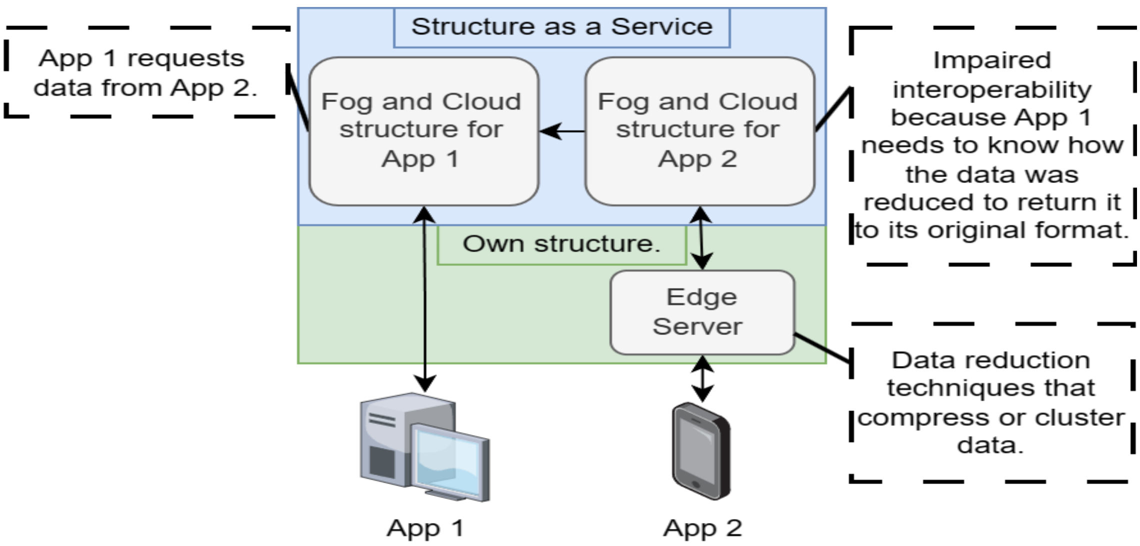

Figure 2. Example of how reducing data, by modifying it, can impede interoperability.

If the shared data is delivered in its modified format, the receiving application needs to know how to undo the modification. Thus, in the same way as discarding data, the greater the diversity of applications which it receives data from, the more manners of undoing modifications it will need to know, adding code that is useless to its original purpose.

4. Proposal

Focusing on those conflicts, we developed an edge-fog-cloud architecture. The architecture uses data reduction techniques without affecting the application's ability to share and receive data at the syntactic level. The solution consists of modules deployed on servers at each layer of an edge-fog-cloud architecture. The idea of the proposal is that the functionalities of these modules, including data reduction itself, are abstract to application developers. As a result, the responsibility for deploying the modules and reducing data would fall to the providers of the edge-fog-cloud services.

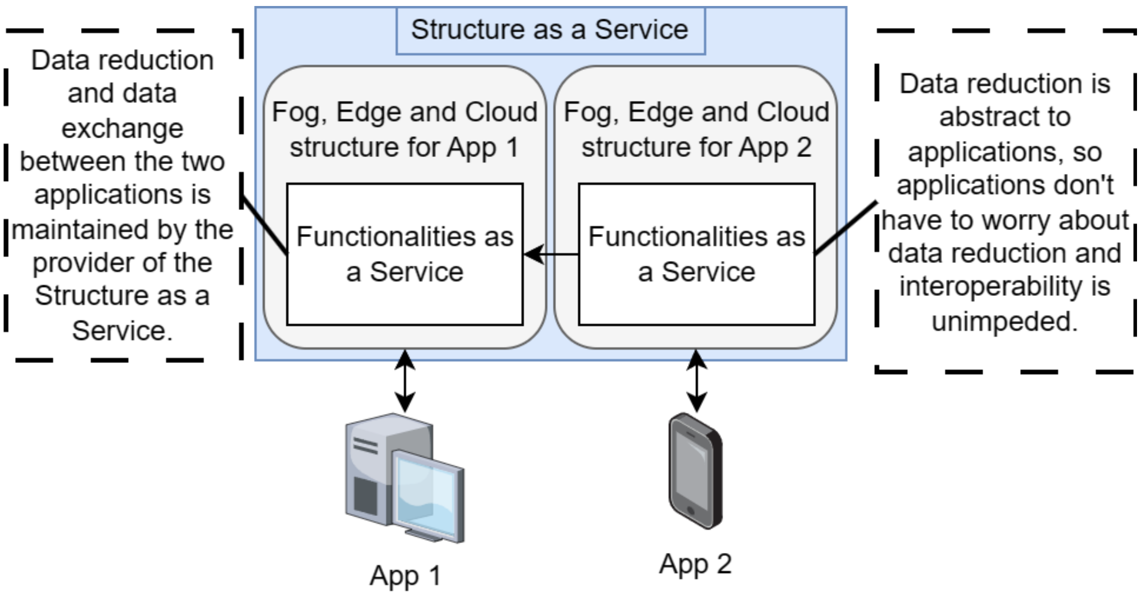

Figure 3. App 2 sharing its data with App 1 through the functionalities offered by the service provider.

The optimal use case of the proposal depends on the existence of Edge servers. In the occasion where edge servers do not exist, the next option would be fog servers. There are two possibilities in which the proposal is able to reduce the costs of data traffic to the cloud server. In the first, the application sends the data to the Edge server and, in the second, if there is no available Edge server, the data is sent to the Fog server. After receiving the data, the server performs data reduction and sends it to the server on the next layer. We believe that, in the event that there are no Edge servers available, the decrease in transport costs would be smaller, as the data would have to travel a longer route in its original format. However, the decrease in storage costs will remain the same, since the data would already be reduced when it reaches the cloud storage server.

With the proposed data reduction approach, the application does not know that the data it sent was reduced. Because of the abstraction, and the fact that data is not discarded, applications are able to communicate at a syntactic level, without the hindrances caused by the reduction of their data. Therefore, since reducing data is the responsibility of the edge-fog-cloud architecture provider, application developers do not need to deal with this responsibility. Figure 4 presents a situation in which two applications share data within the proposed edge-fog-cloud architecture.

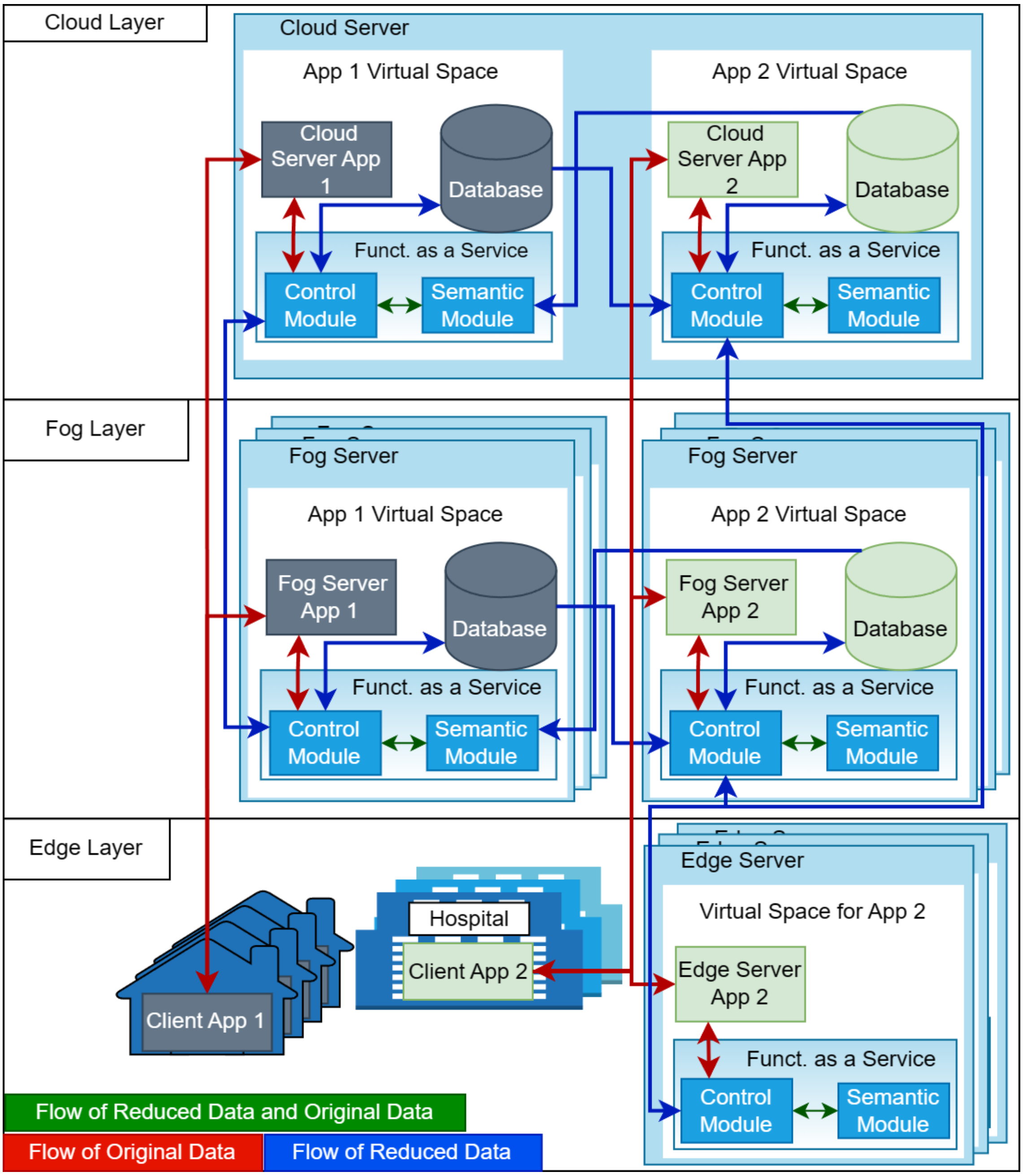

Figure 4. Data flow where two applications use the architecture to interoperate.

The applications illustrated by Figure 4 use the client-server model and are related to the healthcare area. Application 1 was designed to assist patients in their homes, it collects data about the environment around the patient and about their physical condition, in addition to working as a means of remote healthcare. Application 2 supports healthcare professionals in caring for their patients; It is used to manage appointments, patient data and it generates statistics for decision-making.

Application 1 didn't rent edge servers, while Application 2 uses the complete structure. The edge servers are the starting point for data processing, where data is reduced and application-specific pre-processing is performed. The fog server also performs data processing and reduction, in addition, it serves as a partial database, that prioritizes storing the most searched data in the region in which it is located. The cloud server is at the last level of the architecture; different from the other levels, it's a single server. It keeps a copy of all the important data for the application and performs processing that requires much more computational power than the fog and edge servers could offer.

The control module is responsible for the flow of data within the architecture. Offered as a service by the architecture provider, the control module works as the middleman for data exchange between applications. The Semantic Module houses an ontology that is the structure used to reduce and expand the data. An ontology that follows the standards regulated by the Brazilian federal government was chosen, as per the Catálogo de Padrões de Interoperabilidade de Informações de Sistemas de Saúde(CPIISS)[1].

4.1. Data Reduction

By following the CPIISS, we found out that OpenEHR is the semantic standard recommended by the Brazilian government for interoperability between healthcare applications. The official OpenEHR standard wiki[2], recommends the Open Biological and Biomedical Ontology Foundry website[3] as a relevant resource as it pertains to OpenEHR. To find the ontology we used in our project, we studied the ontologies present on this site. Among them, we chose the one developed by Schriml et al. (2019), called Human Disease Ontology, which has a size of 30 megabytes. Its main function is to classify rare and common diseases that can be contracted by modern humans. Furthermore, its scope encompasses the symptoms that can be caused by these diseases, phenotypes, disease-causing elements, susceptibilities and other terms related to contracting diseases.

Every ontological class is assigned a value at the time of its creation, known as id. By default, the ID is a unique value for each class. Like in databases, the ID is a field used to identify something which it is associated with. In the Human Disease Ontology, the ID value is a string composed of two values, a set of characters related to the root class and a natural number, separated by ':' (colon). In our project, we extract the ID value to use it as the reduced version of the received data. When the edge/fog server receives the original data, it associates it with its proper classes. After this process is properly completed, the control module extracts the ID of each class from the ontology. Because the ID itself is also a string, we replace the character set with a natural number and remove the colon symbol. Thus, we obtain a natural number that can be treated as an unsigned integer. Finally, these unsigned integers are sent to the server of the next layer of the architecture, until they reach the cloud server.

By making use of the interface of the Protégé tool[4] to facilitate the visualization

[1] Available at: https://www.conass.org.br/biblioteca/wp-content/uploads/2011/02/NT-37-Padrões-de-Interoperabilidade-versão-2011.pdf

[2] Available at: https://openehr.atlassian.net/wiki/spaces/ontol/overview

[3] Available at: http://obofoundry.org/

[4] Available for download at: https://protege.stanford.edu/

of the process, Figure 5 illustrates the process done to reduce data.

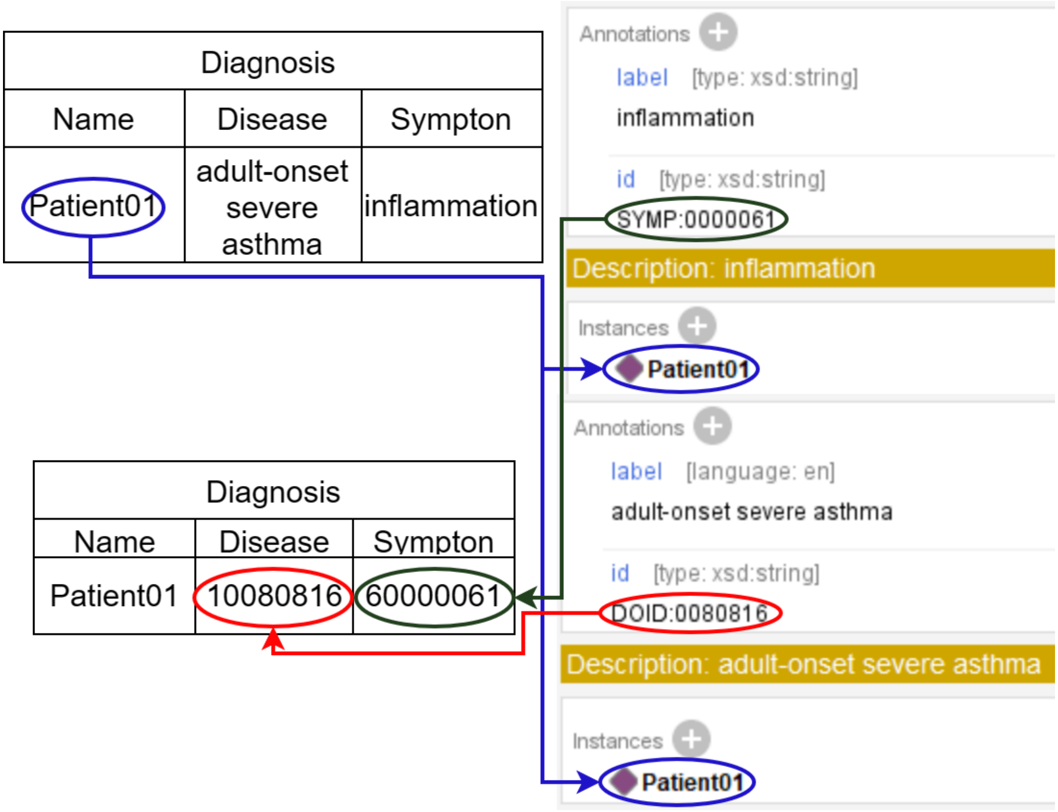

Figure 5. Abstract diagram of the data reduction process.

In SQL, the size of a VARCHAR is 1 byte for each of its characters plus 2 bytes, in contrast, the size of an unsigned INT is fixed at 4 bytes. With these 4 bytes, an unsigned INT can represent 4,294,967,296 (four billion, two hundred and ninety-four million, nine hundred and sixty-seven thousand, two hundred and ninety-six) different numbers. In our example, "adult-onset severe asthma" has 25 characters, that is, it occupies 27 bytes. Inflammation is 12 characters long and occupies 14 bytes. So, in this case, we have that this process reduced the size of the data from 41 bytes to 8 bytes, that is, it decreased by approximately 80.5% (eighty point five percent). It is worth pointing out that these values consider only the data that went through the reduction process.

Although it was used specifically for health applications, simply changing the ontology would enable the use of this proposal in other contexts. Since an ontology is necessary to verify that the applications comply with the chosen standard, embedding the data reduction to that verification simplifies the process. Nevertheless, applying a different lossless data reduction technique, such as compression, would also suffice our conditions.

5. Experimental Environment and Results

We chose to use context simulators to prove the functionality of the proposed concept. This became indispensable because the structure needed to test its operation would require high expenses, which we would not be able to cover. For familiarity, we chose to use the simulator called Siafu[5]. In Costa, Dantas and David (2021), we used Siafu to develop a simulation that would produce data for tests with the proposed application. Where it proved capable of meeting our needs.

5.1. Simulated Scenario

We built our scenario based on Figure 4. This figure was used because we intended to simulate the functioning of the proposed architecture. Another important aspect, chosen during simulation planning, was the background image. The background image does not affect execution, but it facilitates to recognize the context that the simulation is representing. As illustrated in Figure 6, we chose the map of Brazil for our context.

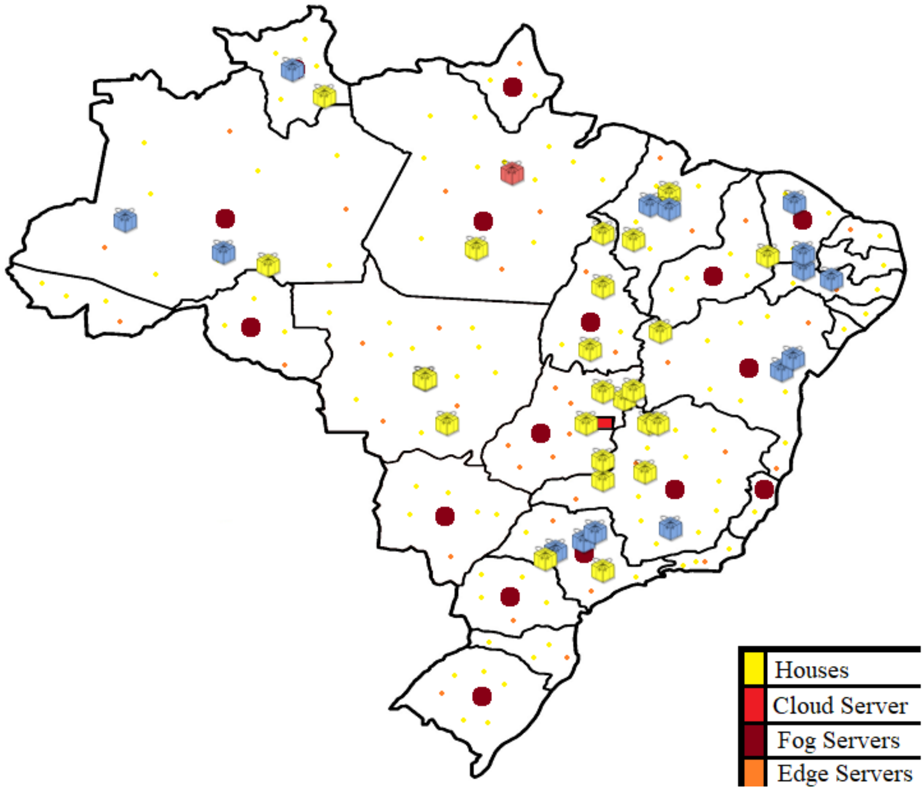

Figure 6. Simulation demonstrating data traveling through the architecture.

Several circles of different colors and sizes and a rectangle were scattered across the map. The yellow circles are the patients' homes, these patients use the application called Application 1. The orange circles are hospitals that have edge servers at their location. Hospitals use Application 2 and edge servers to receive, send, reduce and expand data. The larger, dark red, circles stand for the fog servers. The fog servers receive, send, store, reduce and expand data from both applications, sent by edge servers and houses that are around their mediation. The light red rectangle indicates the location of the cloud server. The placement of servers, homes and hospitals does not represent any real-world infrastructure, any similarity is purely coincidental.

Packets are the data that travels through the architecture's network. Packets that carry reduced data are colored yellow. Blue packets represent data in its original format.

[5]Disponible for download at: https://github.com/tfeijo/Siafu

Finally, the red packet corresponds to data sharing, through syntactic interoperability, between the two applications. To simulate the duration of data transmission over the network, the packet speed will depend on the size of the data it carries. The larger the data size, the slower the packet are, and the longer the transmission will take.

5.2. Results

With the syntactic interoperability guaranteed by the red color package, the next step was to illustrate that the proposed data reduction approach impacts the performance of the architecture. For this, we run our simulation in two ways, in the first way we do not use any type of data reduction and in the second we use the proposed approach. At the end of each simulation, we collect the number of packets and the amount of data, in bytes, that the cloud server received from Application 1 and Application 2. We limited the number of packets transitioning simultaneously, that is, if the number of packets transiting in the simulation is equal to the maximum allowed quantity, a new packet cannot be sent until another packet arrives at its final destination. By Limiting the maximum number of packets, we ensure that, in order to increase the total number of packets delivered, it will be necessary to increase the packet transmission speed. Consequently, if data reduction effectively decreases the data size, the transmission time will be shorter and more packets will be delivered when compared to the version without data reduction.

We performed eight simulations, lasting 42 (forty-two) hours each. Four simulations were done using the standard data reduction technique, while the remaining four did not use any kind of reduction. Recalling the Application 1 structure in Figure 4, it does not use edge servers. In the simulation with data reduction, the data still needs to travel the distance from the edge layer to the fog server in its non-reduced format. Meanwhile, Application 2 makes use of edge servers.Using these servers minimizes the extent to which packets need to travel with their unreduced data. Nonetheless, it persisted that, in all iterations of the simulation with reduction, about four times more packets of both applications were delivered to the cloud server. The medium amount of reduced packets delivered was 21450 (twenty-one thousand, four hundred and fifty), summing to 343200 (three hundred and forty-three thousand and two hundred) bytes. The amount for unreduced packets was 5758 (five thousand, seven hundred and fifty-eight), sizing to 537350 (five hundred and thirty-seven thousand, three hundred and fifty) bytes. The size, in bytes, of the total data that traveled over the network, had an average reduction of 35.8% (thirty five point eight percent), in the simulations that made use of the data reduction.

With the reduction in use, fewer data traveled over the network and stored on the servers. Even so, it delivered more information. By basing ourselves on the relation between the cost of storing and transmitting, and data size. We could assume it lowered the costs of those operations. Implying the effectiveness of the proposal within the simulated environment.

6. Conclusions and Future Works

This paper proposed an architecture edge-fog-cloud where functionalities are offered as a service for health and medical applications. The objective was to tackle the conciliation of syntactic interoperability and data reduction, qualities that enter in conflict when tackled with common approaches. The conciliation of these two qualities was seen as necessary for the fact that both of them are quite beneficial to health and medical applications.

To achieve the proposed objective, we researched about e-health, interoperability, data reduction techniques, cloud computing, fog computing, edge computing, hybrid approaches to edge-fog-cloud computing and Web Ontology Language. With the knowledge base necessary complete, we made use of Siafu, a context simulator, to simulate the solution working in a practical context. The data extracted from the solution was used to verify if the solution was effectively reducing the data, while the applications in the simulation shared data with each other. Which was shown that it was capable of doing so.

Interoperability contains many levels. Each level demands more effort from the applications to understand new information from the data that is being shared. As syntactic is one of the lowest levels, it would be interesting to apply this solution to higher levels of interoperability. Such endeavors could open a new range of information to applications related to healthcare, whilst taking advantage of data reduction benefits.

Acknowledgment This work was realized with the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Financing Code 001 and CNPq.

References

AbuKhousa, E., Mohamed, N., and Al-Jaroodi, J. (2012). e-health cloud: opportunities and challenges. *Future internet*, 4(3):621–645.

Costa, J. P., Dantas, M. A., and David, J. M. (2021). Uma abordagem computacional em ambientes assistidos baseada em atividades físicas e biometeorologia. In *Anais Estendidos do XXI Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 121–126, Porto Alegre, RS, Brasil. SBC.

Idrees, S. K. and Idrees, A. K. (2022). New fog computing enabled lossless eeg data compression scheme in iot networks. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–14.

Kahdim, A. N. and Manaa, M. E. (2022). Design an efficient internet of things data compression for healthcare applications. *Bulletin of Electrical Engineering and Informatics*, 11(3):1678–1686.

Rinty, M. R., Prodhan, U. K., and Rahman, M. M. (2022). A prospective interoperable distributed e-health system with loose coupling in improving healthcare services for developing countries. *Array*, 13:100114.

Rubí, J. N. S. and Gondim, P. R. d. L. (2020). Interoperable internet of medical things platform for e-health applications. *International Journal of Distributed Sensor Networks*, 16(1):1550147719889591.

Scarpato, N., Pieroni, A., Di Nunzio, L., and Fallucchi, F. (2017). E-health-iot universe: A review. *management*, 21(44):46.

Schriml, L. M., Mitraka, E., Munro, J., Tauber, B., Schor, M., Nickle, L., Felix, V., Jeng, L., Bearer, C., Lichenstein, R., et al. (2019). Human disease ontology 2018 update: classification, content and workflow expansion. *Nucleic acids research*, 47(D1):D955–D962.

Shull, J. G. (2019). Digital health and the state of interoperable electronic health records. *JMIR medical informatics*, 7(4):e12712.