IADIS International Journal on WWW/Internet
Vol. 16, No. 2, pp. 66-79
ISSN: 1645-7641

A MODEL FOR BLOCKCHAIN-BASED DISTRIBUTED ELECTRONIC HEALTH RECORDS

Tiago Quaini, Alex Roehrs, Cristiano André da Costa, and Rodrigo da Rosa Righi
Software Innovation Laboratory - SOFTWARELAB, Applied Computing Graduate Program,
Universidade do Vale do Rio dos Sinos - UNISINOS, Av.Unisinos 950, São Leopoldo, Brazil, 93022-750

ABSTRACT

Electronic health records (EHR) are usually maintained in a centralized way by health organizations, leaving aside an integration between different health organizations to view the complete health history of a given patient. This lack of EHR integration prevents patients and physicians to have a unified view of medical records, which are often stored in different health organizations. Recent studies have proposed using Blockchain technology in distributed architectures due to its security and integration properties. In this context, this article proposes an application architecture using Blockchain technology for distributed EHR integration, which is called UniRec (Unified Medical Records). In the proposed model we seek to understand its effectivity, performance and applicability, using a case study methodology. We developed a prototype, which is evaluated using a test scenario. The prototype proved to be effective for distributed EHR integration, allowing one healthcare institution to view an EHR previously added by different providers, after being granted permission. During the test scenario execution, we obtained an average response time of 2.77 seconds, 961.6 MiB of memory consumption, 21.5% CPU usage and a total disk usage of 34.6 MB. The results reinforce the potential and feasibility of employing Blockchain for managing and storing medical data.

KEYWORDS

Electronic Health Record, Blockchain, Ethereum, Peer-to-Peer, Health Informatics

1. INTRODUCTION

Electronic Health Record (EHR) refers to a digital format structure of a patient's health data, which is maintained throughout its lifetime and is stored in a secure repository (Costa et al., 2018; Roehrs et al, 2017). Currently, health data is mainly centralized stored, which hinders the information exchange and collaboration between healthcare institutions (Simić et al, 2017). Healthcare institutions are often reluctant to share their data because of concerns regarding patients' privacy and data security, since failing to secure patients' EHR can lead to legal and financial consequences for the institutions (Peterson et al, 2016). This means that the patient has its EHR stored locally in each healthcare institution through which it has passed throughout its life, leading to problems for physicians and patients such as loss of information and diagnoses based on incomplete data (Lo Duca et al, 2017). Originally build as a record distribution protocol for the Bitcoin electronic currency system (Nakamoto, 2008); the Blockchain concept has been applied in different areas recently, including healthcare (Pilkington, 2015). Using cryptography techniques for users and systems identification and authentication (XIA et al, 2017), Blockchain provides distributed systems capable of interacting with its nodes in an auditable and reliable way (Simić et al, 2017). These Blockchain features fit into the needs faced by the EHR area, such as security, auditing and availability (Fernández-Alemán et al, 2013).

Recently, several articles have presented architecture models for EHR using Blockchain concepts (Roehrs et al, 2017a; Ekblaw, 2017; Ichikawa et al, 2017; Peterson et al, 2016; Xia et al, 2017). However, few papers implement and evaluate the models using the proposed architectures in real scenarios through prototypes, limiting their studies to theories, hypothesis or simulations. The main objective of this paper is to propose an architecture model for distributed EHR integration using Blockchain and its later application through a prototype in order to verify its effectiveness and seeking to understand the results that can be obtained with the use of the Blockchain platform.

The integration of distributed EHR among healthcare institutions can support physicians for a more accurate diagnosis, since more information about the patients would be available. In addition, the use of a fully distributed architecture combined with EHR management by patients' guarantees higher privacy to the records, since only authorized parties are able to visualize its contents, storing and accessing them without the need for an intermediate part (Roehrs et al. 2018). Finally, the integration of distributed EHR can benefit the scientific research area as a whole, since researchers with the appropriate authorizations could access the complete history of certain patients using a layer that anonymizes this data.

In this sense, the main contribution of this article is to propose a model that contemplates integration, traceability and security of distributed EHR. The model is named UniRec (Unified Medical Records) and consists of a private peer-to-peer network (P2P) shared between different healthcare organizations, including hospitals, clinics and laboratories, where medical data are stored and maintained by each institution. To evaluate UniRec feasibility we assess its effectivity, performance and applicability by developing a prototype, employing a case study methodology.

The remaining of the article is organized as follows. Section 2 presents some background concepts. The method employed to evaluate the proposed model is presented in Section 3. The next section details the proposed model, named UNIREC. Section 5 focuses on evaluation, presenting the main results. In Section 6, we discuss related works, comparing UNIREC with the state-of-the-art in the field. Finally, in Section 7 we draw some conclusions.

2. BACKGROUND

According to the International Organization for Standardization (ISO), EHR can be defined as a "repository of information regarding the health status of a subject of care, in computer processable form" (ISO/TR 20514, 2005), which is maintained throughout the patient's life and is kept in a secure repository. However, there are some limitations in EHR, since its records are based entirely in data provided by healthcare organizations (Roehrs et al, 2017). To address this issue, the concept of Personal Health Record (PHR) was proposed, which extends the concept of EHR providing the patients control over their health data, including the possibility of creating records and granting access for third parties to its records (ISO/TR 14292, 2012).

Originally proposed by Satoshi Nakamoto as a record distribution protocol for the Bitcoin electronic currency system, Blockchain is a decentralized record system. It is made up by a set of blocks, where each block represents one transaction and points directly to the previous block, creating an immutable chain. Instead of relying on a centralized authority for ensuring the network integrity, each node is responsible of validating the current block of the chain, in a validation process called mining (Nakamoto, 2008). In its original proposal, Blockchain served only as an open ledger for the cryptocurrency Bitcoin. However, after its release, different platforms were developed to extend Blockchain's technology beyond cryptocurrency, such as the Ethereum platform. Ethereum has the goal of being a generic Blockchain platform in order to leverage the development of applications for different areas (Buterin, 2014). It provides a programming language called Solidity, which allows the development of smart contracts and decentralized applications. Smart contracts are code blocks that directly control the exchange or distribution of digital assets, such as crypto-coins, according to pre-established rules between the parties involved (Zhang et al, 2017).

The InterPlanetary File System (IPFS) is a P2P distributed file system where all nodes in the network store its data locally and connect to each other in order to transfer such data. IPFS uses a block storage model, creating a Merkle DAG structure where the link between objects are cryptographed hashes. This approach makes all content stored in IPFS uniquely identifiable through its multihash verification sum and provides the following properties to IPFS: content addressing, where every object in IPFS is reachable through its unique identifier; tamper resistance, where IPFS is able to detect data adulteration or corruption; and deduplication, where all objects that have the same content are equal, and therefore stored only once (Benet, 2015).

OpenPGP is a non-proprietary protocol for encryption, combining symmetric key cryptography and public key cryptography to provide confidentiality. Although it is most commonly used to encrypt e-mail communication, PGP can be used to encrypt any data or files (Wilson and Ateniese, 2015). Each symmetric key is used only once and for a single object, generating a session key. Since the session key is used only once, it is linked to the object and transmitted next to it. To ensure its protection, the key is encrypted using the public key of each recipient for the object (Shaw and Thayer, 2007).

3. METHOD

This article applies a case study as method, which is a research model broadly used in biomedical sciences (Gil, 2007). The methodological steps consist in proposing an architecture model and applying it through a prototype development. Afterwards, a test scenario is executed in the prototype, where technical data is collected for later usage during the model evaluation. The work evaluation is done in a quantitative approach using the technical data collected from UniRec's prototype during the test scenario execution. This data refers to the (a) performance, (b) integrity and (c) scalability of the proposed model, as follows: (a) performance data consists in the response time for adding and querying records from the prototype; (b) integrity data consists in execution of attempts to violate the access permissions of the records stored in the prototype; and (c) scalability data consists in disk, CPU and memory usage by the prototype during the execution, as well as CPU data from the miner node for the Blockchain network.

The test scenario consists in the addition of an EHR by one healthcare organization and the later integration of the same EHR with another organization. Its steps are described below, and can be visualized in Figure 1:

1. Healthcare organization A adds patient X to UniRec. Goal: check if an organization is able to add a patient to the system.
2. Healthcare organization A adds an EHR for patient X. Goal: check if an organization is able to add an HER to a given patient.
3. Healthcare organization B queries UniRec for the metadata of patient X's EHRs. Goal: check if an organization is able to view a patient's EHRs metadata for a given patient.
4. Healthcare organization B attempts to view the content of patient X's EHR. Goal: check if the Blockchain will restrict the EHR access only to authorized parties.
5. Healthcare organization B requests access to patient X's EHR. Goal: check if the organization without access to the EHRs are able to request access for it.
6. Healthcare organization A grants healthcare organization B access to patient X's EHR content. Goal: check if the EHR owner organization is able to grant other organizations access for the EHR content.
7. Healthcare organization B views the content of patient X's EHR. Goal: check if, with the proper authorizations, the organization is able to view the EHR content.

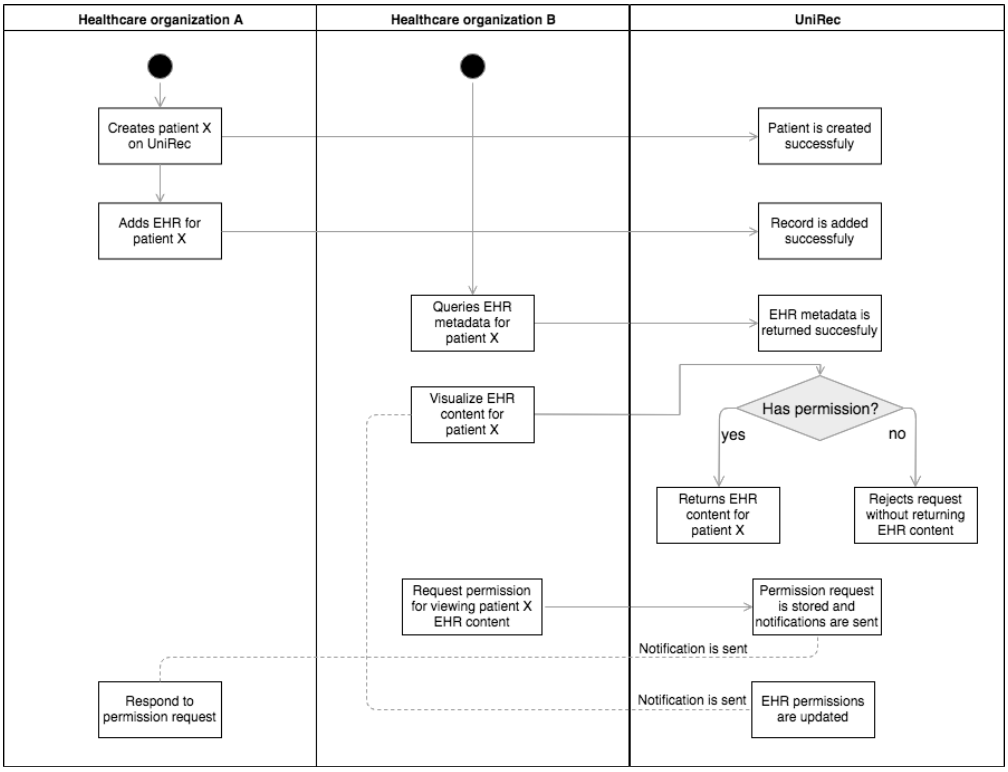
Healthcare organization A
Healthcare organization B
UniRec
Creates patient X on UniRec
Patient is created successfuly
Adds EHR for patient X
Record is added successfuly
Queries EHR metadata for patient X
EHR metadata is returned succesfuly
Visualize EHR content for patient X
Has permission?
yes
no
Returns EHR content for patient X
Rejects request without returning EHR content
Request permission for viewing patient X EHR content
Permission request is stored and notifications are sent
Notification is sent
Respond to permission request
Notification is sent
EHR permissions are updated

Figure 1. Flowchart of UniRec Model's Test Scenario

4. UNIREC MODEL

This section presents the UniRec (Unified Medical Records) model, which seeks to contemplate integration, traceability and security of distributed EHR. UniRec consists of a private P2P shared between different healthcare organizations, such as hospitals, clinics and laboratories, where medical data are stored and maintained by each institution. In this P2P network is contained a Blockchain Ethereum, which maintains a shared an immutable history of each EHR. Medical data is shared between institutions in the private P2P network through IPFS, where a reference to the file is added to the Blockchain through an IPFS URI, making it available for discovery from other institutions. This model is illustrated in Figure 2.

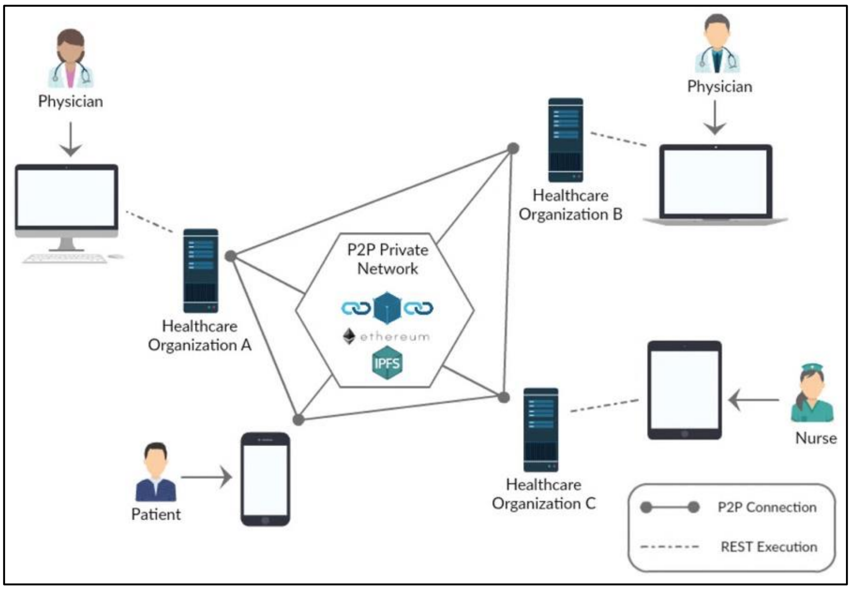

Figure 2. Overview of the UniRec Model

Access authorizations are maintained individually for each EHR by the healthcare organization that adds it to the network, and can be changed at any time by the patient who owns the record through a mobile app. In order to participate of the network, the patient must register with a healthcare organization in order to authorize the integration of its EHR and also to be part of the private network. UniRec has an event-based architecture and is built using the Node.js platform, which is responsible for managing the Blockchain, adding and querying EHRs in the P2P network, cryptographing the EHRs' content and exposing the application functionalities through REST services. The Node.js application is structured in contracts, models and controls, communicating with the P2P network for accessing data and the Blockchain. The contract layer contains the smart contracts written for the Ethereum Blockchain using Solidity. These contracts are compiled and created by the application using Ethereum's JavaScript API, which is called web3.js. This API is used exclusively by the repositories in the model's layer, which encapsulates all the communications between the application and the Blockchain. In addition, the models' layer also uses the IPFS JavaScript API for adding, removing and querying EHRs in the P2P network. Before being added to the P2P network, EHR content is cryptographed using OpenPGP API. Finally, the application controllers have the responsibility of communicating with the repositories and exposing the application's functionalities through REST services, which are consumed by healthcare organizations.

Before being added to the P2P network, each EHR is cryptographed using OpenPGP. The EHR's owner public key is used, and thus only this party – healthcare organization or patient – is able to access the contents of the EHR using its private key. In order to view the content of the EHR, other healthcare organizations have to request access to its contents. This request is sent to the Blockchain and must be approved either by the patient or by the owner healthcare organization. Although the owner healthcare organization has autonomy to grant access to other institutions, the patient is able to revoke access to its EHR whenever needed. Whenever an access request is approved, the owner healthcare organization generates a new cryptographed version of the EHR using the requesting organization's public key and adds this new version of the EHR to the P2P network afterwards, allowing the requesting organization to view the EHR content. In addition, IPFS is able to detect if the content of a given EHR has been changed due to its tamper resistance, which is able to detect changes or corruptions, and would make the file's unique identifier invalid.

5. RESULTS AND DISCUSSION

Two virtual machines were used during the test scenario execution, each representing one healthcare organization running UniRec. The virtual machines were created using Virtual Box, and were executing Ubuntu 16.04 OS, having 10 GB of disk space and 2 GB or RAM memory each. The host computer was executing Windows 10 OS, having an Intel Core i7-6500U processor and 8 GB of RAM memory. In addition, the private Blockchain was mined through a MacBook Pro with an Intel Core i7-4770HQ processor and 16 GB of RAM memory, connecting to the Blockchain through the host computer IP address. The connection between nodes for the test scenario execution is represented in Figure 3.

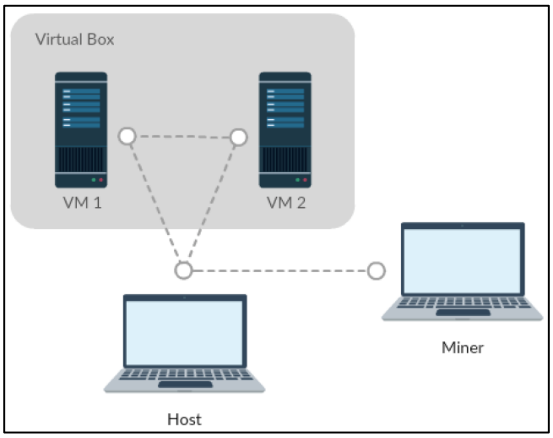

Figure 3. Connection between Nodes for Test Scenario

UniRec model was implemented on top of the Node.js platform using an event-based architecture. The Node.js application manages the Blockchain and the P2P network, adds and retrieves the EHRs from the P2P network, cryptographs the EHR content and exposes its functionalities through REST services, as seen on the component diagram illustrated in Figure 4. It is structured in contracts, models and controllers: the contract layer contains smart contracts written using Solidity for the Ethereum Blockchain, which are compiled by the application using the Ethereum's web3.js JavaScript API; the model layer contains the repositories, which uses both web3.js and IPFS APIs to communicate with the Blockchain and the P2P network respectively; lastly, the controllers are responsible for interacting with the repositories and exposing the application functionalities through REST services, which are consumed by the healthcare organizations.

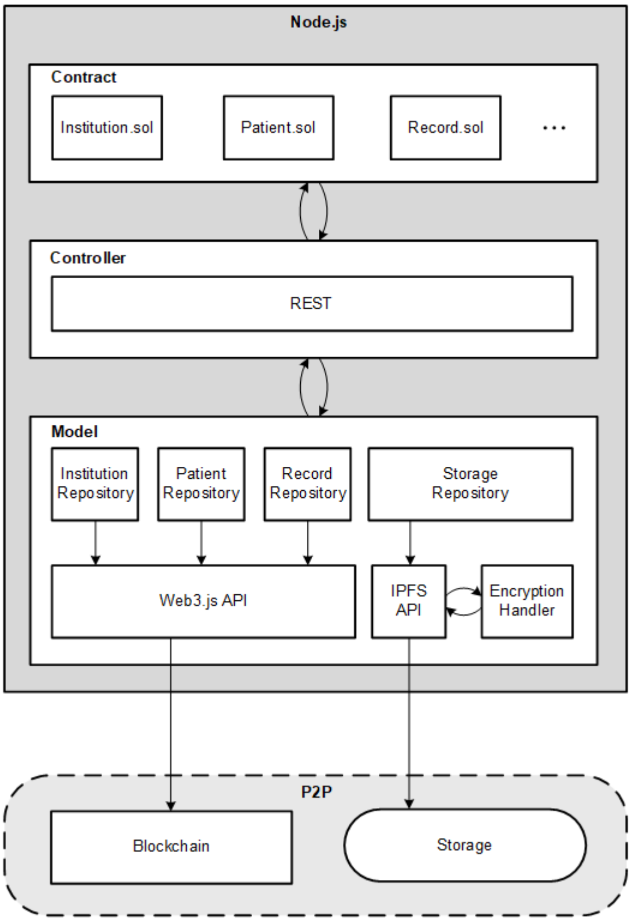

Figure 4. UniRec Model Architecture

The prototype covered all the steps of the test scenario successfully, returning the expected result for each step of the scenario. Technical data related to memory, CPU and disk consumption were collected for the three different processes required for UniRec's execution: UniRec Node.js process, Ethereum and IPFS. These data were collected using Ubuntu's activity monitor. The response time was collected directly from Postman software, which was used for consuming UniRec's APIs. In average, the API executions during the test scenario took 2.77 seconds to be completed. Furthermore, executions that performed changes on the Blockchain contracts, such as creating or modifying an existent contract, took 3.75 times longer to be completed compared to executions that only queried the data in the Blockchain, which took an average of 0.78 seconds to be completed. The response time identified in each step from the test scenario execution is illustrated in Figure 5.

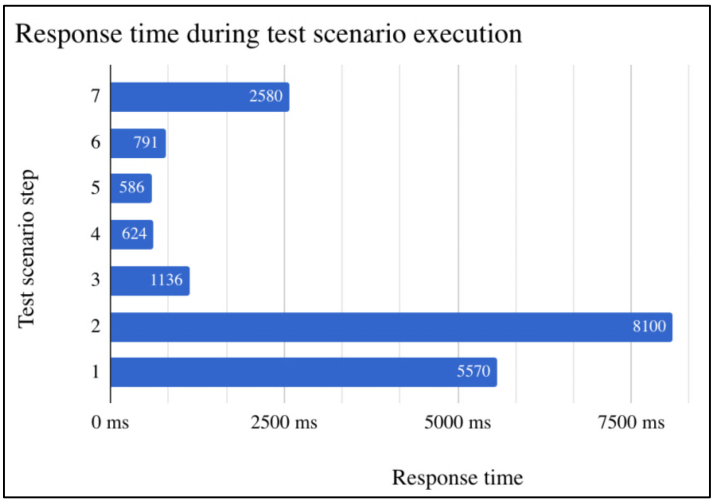

Figure 5. Response Time during Test Scenario Execution

Regarding the memory allocation, both UniRec and Ethereum components presented a stable and linear allocation, even though Ethereum's allocation presented a notable difference in node 1 and 2. The memory consumption in nodes 1 and 2 is demonstrated in Figure 6. Ethereum needed an average memory allocation of 1116.7 MiB on node 1, while it needed an average of 804.4 MiB on node 2. UniRec presented an average memory allocation of 106.9 MiB on node 1 and 114 MiB on node 2. On the other hand, IPFS memory allocation presented the highest variation in the measured processes, varying from 51.1 MiB to 79.8 MiB on node 1 and from 43.9 MiB to 85.1 MiB on node 2. On average, IPFS memory allocation was 75 MiB on node 1 and 69.2 MiB on node 2. Considering the prototype's disk usage, Ethereum and IPFS were analyzed. Ethereum disk usage in node 1 varied from 34.3 MB at the beginning of the execution to 35.1 MB at the end, and in node 2, it varied from 33.3 MB to 33.6 MB. On the other hand, IPFS varied from 212 kB to 264 kB in node 1, and from 208 kB to 260 kB on node 2.

In the matter of CPU consumption, it was the analyzed metric that presented the highest variations during the execution of the tests. UniRec's CPU consumption has significant increase whenever a new execution is started, but not affecting the other node's consumption, and reaching peaks of about 40% in both nodes. IPFS CPU consumption was stable, presenting its peaks of 9% in node 1 and 31% in node 2 during steps 2 and 7, respectively, which are the steps where files are added and queried from the P2P network. Lastly, Ethereum's CPU consumption was stable, staying below 10% during the whole execution in both nodes. In addition, the miner node CPU consumption presented huge variations, having an average CPU consumption of 40.63% and presenting 6 peaks of more than 75% CPU consumption.

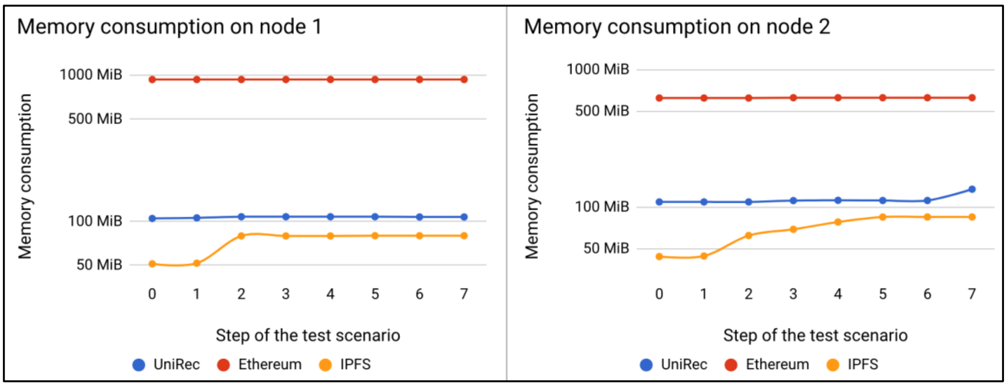

Figure 6. Memory Consumption in the Nodes during Test Execution

According to Nielsen (2014), 10 seconds is the time limit for keeping a user's attention while a computer system performs a task. With that in mind, it is possible to say that UniRec prototype presents an acceptable performance regarding its response time, since its executions during the test scenario took an average of 2.77 seconds to be completed. In addition, the executions that did not involve creating patients or adding records to UniRec took an average of 0.78 seconds to be completed, which according to Nielsen (2014) fit into the time interval where users' flow of thought remain uninterrupted during the software execution. Considering specifically UniRec Node.js component, it was responsible for the usage of about 9.57% from the total memory allocation used by the prototype in node 1, and about 14.17% in node 2. In addition, the only significant change in memory allocation by UniRec was identified at node 2 during the execution of step 7, where node 2 has to search the P2P network through IPFS and decrypt it using OpenPGP. Regarding CPU consumption, the steps that handle encryption of EHR content and managing it on the P2P network through IPFS are the steps that use more CPU in both nodes, reaching up to about 40% CPU consumption.

Regarding IPFS component, it presented low CPU and memory consumption, as well as low disk usage. A positive factor of IPFS is its disk usage, since an EHR added by one node does not occupy any disk space on any other nodes until they have permission and request for its content. The only downside noticed for IPFS during the test scenario execution was its CPU consumption, which presented irregularities during node 2's execution, presenting oscillations and high CPU usage even when IPFS was not directly triggered by any UniRec executions. In node 1, this behavior was not observed. However, some negative points were identified in Ethereum component. Ethereum used about 83.71% of the entire memory allocation for UniRec prototype in node 1. In node 2, Ethereum uses 77.23% of the total memory allocation. The CPU usage was stable in both nodes, with a peak of 20% in node 2 during step 7 execution. Nevertheless, the miner node presented a high CPU consumption, with an average consumption of 40.63% and presenting 6 peaks of more than 75% CPU consumption, which can be assigned to the creation of smart contracts during the test scenario, specifically in steps 1, 2, 5, 6 and 7.

Regarding Ethereum's disk usage, before the test execution start the node 1 was already using 34.36 MB and node 2 was using 33.31 MB, which refer to the directed acyclic graph (DAG) which was mined by the private network prior to the test execution. At the end of the test scenario execution, node 1's disk usage had increased 760 kB, while node 2's disk usage had increased 296 kB. Anyway, the disk usage in Ethereum was higher than in IPFS, where the actual EHR content is stored, and could be a potential issue for the model's performance and scalability, since a huge amount of disk space would be needed to support more healthcare organizations and patients in UniRec. Considering the development of the prototype, some lessons learned can be highlighted. The APIs provided by both Ethereum and IPFS were functional, meeting expectations and needs during prototype development. The major challenges faced are related to Ethereum's private network setup, which requires the configuration of an initial block in the network, called genesis, which must be the same for each node in the Blockchain network. This block has a unique identifier, called chainID, whose singularity makes the Blockchain into a private network. In addition, a configuration for nodes discovery is required, which is done using a configuration file containing each Blockchain node's identifier, called *enode*, together with the IP of the given node. Together with the configuration file is created a specific node to enable discovery between nodes in the Blockchain, called bootnode. This means that the private Blockchain requires the use of static IP addresses in Blockchain nodes and on the bootnode, since changing them would result in the need of adjusting the discovery configuration of Blockchain node.

Another topic observed during the prototype development phase was the transaction costs in the Blockchain, which in Ethereum is performed using the cryptocurrency *eth*. This cost considers the difficulty of mining the blocks and is used for rewarding the miners that maintain the Blockchain's integrity. In order to execute the test scenario, a virtual wallet was created for each Blockchain node with the necessary number of *eths*. However, for applying UniRec in a real scenario, it would be necessary to elaborate a reward system, in order to provide an incentive to healthcare organizations that add an EHR to the private network and also to those that dedicate their computational resources to mine the network.

6. RELATED WORK

This section presents related works that propose architectures based on Blockchain applied for healthcare. The research sought to consider works that addressed the EHR integration or interoperability subject and presented models using Blockchain for this topic. Although Blockchain was presented in 2008, works suggesting Blockchain implementations for EHRs were only found in the last five years. Therefore, the selected related work present similar concepts to those proposed by this work, seeking to compare its models with the model proposed in this article.

MedRec is a decentralized system for EHR management, which uses Blockchain (Ekblaw, 2017). Its architecture distributes the medical records in different databases, storing references to these records on the Blockchain. In addition, the Blockchain also stores information related to each record's permissions, which are provided and managed by the patients themselves. MedRec model was applied into a small-scale prototype in a healthcare institution in Israel, using Ethereum as its Blockchain platform.

Decentralized Application for Smart Health (DASH) is a Blockchain application focused on the healthcare sector (Zhang et al, 2017). It provides a web portal where patients can access its EHRs and also submit prescription requests, in addition to managing the permissions to its EHRs. This work proposes the application of software design patterns into the DASH model, in order to overcome the issues and challenges faced during the development of DASH prototype.

Ichikawa et al (2017) presented the application of Blockchain technology as a storage platform for a mobile health (mHealth) application focused on cognitive behavioral therapy for insomnia (Ichikawa et al, 2017). In this architecture, the patients upload their data daily through their smartphone and receive feedback for the uploaded data, while the data is stored in the Blockchain structure. During the prototype evaluation, a system failure was simulated, where one Blockchain node was disconnected and later successfully reconnected.

Peterson et al (2016) presented a proposal for sharing EHR in an efficient and safe manner (Peterson et al, 2016), suggesting the structure for a health data block, which rather than containing the contents of the EHR only stores a reference to the record through an URL. However, this work does not present a defined architecture, but some conceptual proposals for an interoperable EHR system.

The Table 1 presents a comparison between UniRec's functionalities and characteristics with the related works. The analyzed points were the model's technical characteristics, its functionalities and also the methodology applied in the related works. Table 1's items were selected according to the information gathered in the bibliographic research step, determining the needs of applications built for EHR integration.

Table 2. Comparison between UniRec and its Related Works

	MedRec (2016)	DASH (2017)	mHealth (2017)	PETERSON et al (2016)	UniRec (2018)
1	FHIR	FHIR	JSON	FHIR	Unrestricted (consensual)
2	E	E	H	E	E
3	Yes	Yes	No	Yes	Yes
4	Yes	Yes	No	Yes	Yes
5	Yes	No	Yes	Yes	Yes
6	Yes	No	Yes	No	Yes
7	Prototype tested with 200 records in a health organization	Applied software design patterns into its prototype	Prototype resisted tampering attempts	Proposed an architecture using Blockchain	Prototype had an average response time of 2.77 seconds

Legend: 1 – Data format; 2 – Blockchain platform; 3 – Patients control EHR access; 4 – Supports different types of users; 5 – Detailed architecture; 6 – Use scenario for testing; 7 – Results summary | E – Ethereum; H – Hyperledeger;

It was decided that the model should use a health interoperability standard as a data format, such as *open*EHR or FHIR. In UniRec model, there is no dependency on the usage of a specific standard, as long as there is a consensus among network participants to use the same standard and data format. Regarding the Blockchain platform to be used in the model, a platform that supports smart contracts development should be used, such as Ethereum or Hyperledger. UniRec model uses Ethereum platform due to its strong integration with IPFS and also due to its extensive documentation and community support. Another important point for choosing Ethereum as the Blockchain platform is its usability, since it offers several methods of communicating with the Blockchain for different programming languages and provides its own programming language for creating smart contracts, called Solidity. In addition, most of the related work models' use Ethereum as its Blockchain platform. Another relevant point to the model is supporting different types of users, such as healthcare organizations and patients, providing patients the possibility of controlling the access to their data. Thus, it was found that most of the related works were applying PHR concepts on its models. UniRec applies the concept of EHR, with patient defined record privacy from the PHR definition added on top of it. Finally, the works methodology for presenting and evaluating its models was compared. In this work, both the model and its evaluation are presented in detail, so that it is possible to clearly visualize the behavior and the potential efficiency of Blockchain applied to the integration of EHRs.

7. CONCLUSION

Blockchain is a very broad subject area and is in highlight currently, with several papers proposing models applying it in different areas, including health care and EHR integration. Among these papers, however, few implement and expose the proposed models to tests with real medical data, limiting the Blockchain proposal on its papers to hypotheses based on its characteristics. Given this open point on researches applying Blockchain for EHR integration, this paper applied its proposed model through a prototype and validated it using real medical data, seeking to understand the model's and also Blockchain's effectiveness in this area. UniRec model's conception was based on the literature review of the EHR area and also on the models proposed by the related works, seeking to fill the gaps and opportunities found by these works. Its main goal is to enable distributed EHR integration and management, allowing healthcare organizations to have a unified view of their patients EHRs and patients to manage their EHRs' permissions individually. Applying the UniRec model through a prototype made it possible to confirm the hypothesis that it is possible to achieve distributed EHR integration and management through Blockchain in a completely distributed way between healthcare organizations and patients, meeting security, auditability, confidentiality and availability requirements without the need of any centralized authority in the process. The prototype presented an average response time of 2.77 seconds, which is a promising performance according to the time guidelines proposed by Nielsen (2014). However, it would be beneficial to evaluate the UniRec model in a future work using a larger scale, with more patients and healthcare organizations being executed on different servers, unlike virtual machines.

In addition, the prototype development provided the opportunity to analyze the compatibility of Ethereum as UniRec's model Blockchain platform. It was verified that the Ethereum was the prototypes' component that presented higher memory and disk consumption, and it was also possible to identify an open point for the application of UniRec in real scale. In Ethereum, transactions have a cost that is paid by *eth* cryptocurrency. However, the UniRec model does not present a strategy to cover the miners' rewards, where healthcare organizations would somehow need to receive *eths* through certain actions, such as adding an EHR to the network, and spending their *eths* to access EHR contents added by other healthcare organizations, when authorized. This question goes beyond the scope of this paper and should be addressed in a future work. Another possibility of future work would be the application of the UniRec model using another Blockchain platform, such as Hyperledger, whose use was suggested in models from related works (Ichikawa et al, 2017).

ACKNOWLEDGEMENT

The authors would like to thank the Brazilian National Council for Scientific and Technological Development - CNPq (Grant Numbers 303640/2017-0 and 405354/2016-9) for supporting this work.

REFERENCES

Benet, J., 2015. IPFS - Content Addressed, Versioned, P2P File System. [online] Arxiv.org. Available at: https://arxiv.org/abs/1407.3561 [Accessed 18 Oct. 2017].

Buterin, V., 2014. A next-generation smart contract and decentralized application platform. [online] Ethereum, pp 1-36. Available at: http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf [Accessed 22 Sep. 2017].

Costa, C. A., Pasulosta, C. F., Eskofier, B., Silva, D. B., Righi, R. R., 2018. Internet of Health Things: Toward Intelligent Vital Signs Monitoring in Hospital Wards. *Artificial Intelligence in Medicine*, Vol. 89, p. 61-69.

Ekblaw, A., 2017. MedRec: Blockchain for medical data access, permission management and trend analysis. [online] Dspace.mit.edu. Available at: https://dspace.mit.edu/handle/1721.1/109658 [Accessed 9 Oct. 2017].

Fernández-Alemán, J., Señor, I., Lozoya, P. and Toval, A., 2013. Security and privacy in electronic health records: A systematic literature review. *Journal of Biomedical Informatics*, Vol. 46, p. 541-562.

Gil, A., 2007. *Como elaborar projetos de pesquisa*. Atlas, São Paulo.

Ichikawa, D., Kashiyama, M., and Ueno, T., 2017. Tamper-Resistant Mobile Health Using Blockchain Technology. *JMIR Mhealth Uhealth*, Vol. 5, p. e111.

ISO, 2012. *ISO/TR 14292:2012 - Health informatics – Personal health records – Definition, scope and context*. [online] Iso.org. Available at: https://www.iso.org/standard/54568.html [Accessed 5 Oct. 2017].

ISO, 2005. ISO/TR 20514:2005: *Health informatics – Electronic health record – Definition, scope and context*. [online] Iso.org. Available at: https://www.iso.org/standard/39525.html [Accessed 4 Oct. 2017].

Lo Duca, A. Bacciu, C., and Marchetti, A., 2017. How distributed ledgers can transform healthcare applications. *ERCIM NEWS*, Vol. 110, p. 25-26.

Nakamoto, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. [online]. Available at: https://bitcoin.org/bitcoin.pdf [Accessed 13 Aug. 2017].

Nielsen, J., 2014. Response Times: The 3 Important Limits. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/response-times-3-important-limits [Accessed 2 May 2018].

Peterson, K. Deeduvanu, R. Kanjamala, P., and Boles, K., 2016. A Blockchain-Based Approach to Health Information Exchange Networks. [online]. Available at: https://www.healthit.gov/sites/default/files/12-55-Blockchain-based-approach-final.pdf [Accessed 7 Aug. 2017].

Pilkington, M., 2015. Blockchain Technology: Principles and Applications. [online] Research Handbook on Digital Transformations, pp 1-39. Available at: http://papers.ssrn.com/abstract=2662660 [Accessed 16 Aug. 2017].

Roehrs, A. Da Costa, C. A., Righi, R. R., and De Oliveira, K. S. F., 2017. Personal health records: A systematic literature review. *Journal of Medical Internet Research*, Vol. 19, p. e13-21.

Roehrs, A. Da Costa, C. A., and Righi, R. R., 2017a. OmniPHR: A Distributed Architecture Model to Integrate Personal Health Records. *Journal of Biomedical Informatics,* Vol. 71, p. 70-81.

Roehrs, A. Da Costa, C. A., and Righi, R. R., Rigo. S., Wichman, M. 2018. Toward a Model for Personal Health Records Interoperability. *IEEE Journal of Biomedical and Health Informatics*, Vol. 22, p 1-7.

Simić, M. Sladić, and G. Milosavljević, B., 2017. A Case Study IoT and Blockchain powered Healthcare. *The 8th PSU-UNS International Conference of Engineering and Technology*.

Wilson, D. and Ateniese, G. 2015. From pretty good to great: Enhancing PGP using bitcoin and the Blockchain. *International Conference on Network and System Security*, p. 368-375.

Xia, Q., Sifah, E., Smahi, A., Amofa, S. and Zhang, X. 2017. BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments. *Information*, Vol. 8, p. 44.

Zhang, P., White, J., Schmidt, D., and Lenz, G., 2017. Applying Software Patterns to Address Interoperability in Blockchain-based Healthcare Apps. [online] Arxiv.org. Available at: https://arxiv.org/abs/1706.03700 [Accessed 25 Aug. 2017].