Journal of Biomedical Informatics 92 (2019) 103140

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Analyzing the performance of a blockchain-based personal health record implementation

Alex Roehrs[a], Cristiano André da Costa[a,*], Rodrigo da Rosa Righi[a], Valter Ferreira da Silva[b], José Roberto Goldim[b], Douglas C. Schmidt[c]

[a] Software Innovation Laboratory (SOFTWARELAB), Applied Computing Graduate Program, Universidade do Vale do Rio dos Sinos (UNISINOS), Av. Unisinos, 950, 93022-750 São Leopoldo, RS, Brazil
[b] Research Group and Graduate Studies (GPPG), Hospital de Clínicas de Porto Alegre (HCPA), Ramiro Barcelos, 2350, 90035-903 Porto Alegre, RS, Brazil
[c] Institute for Software Integrated Systems (ISIS), Vanderbilt University, 1025, 16th Ave So., Nashville, TN 37212, USA

ARTICLE INFO

Keywords:
Personal Health Record (PHR)
Blockchain
Software architecture
Distributed systems

ABSTRACT

Background: The Personal Health Record (PHR) and Electronic Health Record (EHR) play a key role in more efficient access to health records by health professionals and patients. It is hard, however, to obtain a unified view of health data that is distributed across different health providers. In particular, health records are commonly scattered in multiple places and are not integrated.

Objective: This article presents the implementation and evaluation of a PHR model that integrates distributed health records using blockchain technology and the *open*EHR interoperability standard. We thus follow OmniPHR architecture model, which describes an infrastructure that supports the implementation of a distributed and interoperable PHR.

Methods: Our method involves implementing a prototype and then evaluating the integration and performance of medical records from different production databases. In addition to evaluating the unified view of records, our evaluation criteria also focused on non-functional performance requirements, such as response time, CPU usage, memory occupation, disk, and network usage.

Results: We evaluated our model implementation using the data set of more than 40 thousand adult patients anonymized from two hospital databases. We tested the distribution and reintegration of the data to compose a single view of health records. Moreover, we profiled the model by evaluating a scenario with 10 superpeers and thousands of competing sessions transacting operations on health records simultaneously, resulting in an average response time below 500 ms. The blockchain implemented in our prototype achieved 98% availability.

Conclusion: Our performance results indicated that data distributed via a blockchain could be recovered with low average response time and high availability in the scenarios we tested. Our study also demonstrated how OmniPHR model implementation can integrate distributed data into a unified view of health records.

1. Introduction

The adoption of the Electronic Health Record (EHR) has evolved as a consolidated technology for recording patient health data [1,2]. A key difference between an EHR and a Personal Health Record (PHR) is that a PHR enables patients to access and control their own data [3]. PHR is an emerging trend with growth potential in the health care domain [4]. Improving the management and sharing of health records is a key focus of our work reported in this article.

Although initiatives to adopt PHR have evolved in recent years, they face barriers to adoption [5]. One barrier faced by both EHR and PHR is the distribution and limitations of health record integration. Other barriers relate to security issues, such as confidentiality and privacy of health records [6,7].

Patient health data are conventionally stored in health care provider repositories [8,9]. Often, however, these data are not shared between providers or with patients. Moreover, even where there is an intention to share data, there are barriers to achieving this goal [10], including

(a) Interoperability stemming from the lack of common health data standards [7].
(b) The difficulty of integrating large amounts of data contained in medical records [11].

* Corresponding author.
E-mail address: cac@unisinos.br (C.A. da Costa).

https://doi.org/10.1016/j.jbi.2019.103140
Received 10 October 2018; Received in revised form 24 January 2019; Accepted 22 February 2019
Available online 04 March 2019
1532-0464/

As a consequence, patients must often re-inform their health history, repeat laboratory exams, or even perform unnecessary tests when they are attended by different health providers [12]. Although some countries have initiatives to integrate personal health history, this integration often occurs only at the organizational level, without patients having access to their digital records [13]. In such cases, therefore, only the data reported in the health organizations are integrated, regardless of factors like patient wellness data, nutrition, data collected on wearables, or collected on monitoring equipment at home [14]. Moreover, patient care often comes from health providers who are not part of an integrated network of health organizations, e.g., if patients are treated in a foreign country [9].

Blockchain technologies [15,16] are a promising means to address the barriers with distributed PHRs described above by forming a unified view of PHRs. Blockchain technology has been researched and implemented in various domains, initially in the financial domain with virtual currencies and more recently in the health domain [17,18]. Various approaches to applying blockchain to health data have been proposed, centered largely around composing a distributed ledger of health records [19] and providing useful tools to preserve patient privacy [20].

The performance of distributed PHRs and integration of health data among health organizations are crucial factors to ensuring the adoption of blockchain technologies. In prior work, we have devised an architecture model named OmniPHR [21,22] and characterized its key components and interoperability features. This article extends our prior work using a prototype implementation of OmniPHR model in production scenarios by evaluating health records from two health organizations.

A key aspect of our work involves evaluating a model for distributed PHR integration based on blockchain technology. The research gap that our work addresses involves determining how to develop a distributed and interoperable PHR implementation using blockchain technology to integrate patient health records. In particular, this article

(a) evaluates the distribution and reintegration of health records via blockchain technologies to compose a unified PHR view,
(b) analyzes the assessment of non-functional performance requirements, such as measure response time, CPU usage, memory occupation, disk and network usage of a varied number of superpeers and concurrent sessions transacting different operations on health records simultaneously, and
(c) discusses best practices for deploying blockchain technologies in healthcare.

OmniPHR approach is innovative since it promotes the integration of health data through the use of a distributed, private, and customizable platform, along with interoperable and standards-based protocols. Likewise, we integrate distributed health records in a unified, safe, and interoperable manner for use by health providers and patients. In particular, the key contribution is that OmniPHR promotes the sharing of PHRs among health care providers, with the possibility of knowledge and consent of the patient.

The remainder of the article is organized as follows: Section 2 summarizes the terminology and platforms used in this paper; Section 3 explains the methods used in OmniPHR prototype, evaluation, and results collection; Section 4 describes OmniPHR architecture and the application model, as well as key aspects of OmniPHR implementation and scenarios applied in our evaluation environment; Section 5 analyzes the results obtained from our empirical evaluations and compares our results with related work; and Section 6 presents concluding results and future work.

2. Terminology and platforms

This section summarizes the terminology and platforms used in this article.

Personal Health Record (PHR) can be considered an evolution of an *Electronic Health Record* (EHR). According to ISO/TR 18638:2017 [3], PHR is a "representation of information regarding or relevant to the health, including wellness, development, and welfare of a subject of care, which may be stand-alone or integrating health information from multiple sources, and for which the individual, or their authorized representative, manages and controls the PHR content and grants permissions for access by and/or sharing with other parties."

Blockchain is a linked list of datablocks chained together in a distributed ledger by pointers, represented by a hash code that identifies each block, and where each datablock has, beyond the content, the pointer to the previous datablock in the chain [15,34]. In a blockchain, each node in the peer-to-peer (P2P) network acts as a recorder of datablocks and as an evaluator of appropriate access and permissions of the content. Each node can add new blocks in the list and execute evaluation rules every interaction. These checks are performed in conjunction with the other nodes, forming the consensus protocol [35,36].

Smart contracts are another concept applied in blockchain technology to incorporate business rules or scripts to the processing performed on the platform. According to [37], a smart contract is a "set of promises, specified in digital form, including protocols within which the parties perform on these promises." In many cases, smart contracts are used to verify the validity of contracts between two or more participants in a contract.

One way to make health records interoperable is to use recognized data standards or protocols [38,39]. Several health data standards are defined around the world, with different purposes. Two internationally recognized standards used for electronic medical records are HL7 [40] and *open*EHR/ISO CEN13606 [41]. The *open*EHR standard has the differential to treat health records semantically through ontology [42]. In the *open*EHR standard, instances of datablocks can be serialized in either archetype (RDF/XML or JSON) or ontology (OWL) format, where RDF stands for "Resource Description Framework" and OWL stands for "Web Ontology Language."

3. Methods applied in our study

This section explains the methods used in OmniPHR prototype, evaluation, and results collection. Due to the barriers to adoption of distributed health records across different health providers discussed in Section 1—and in accordance with the background underlying PHR and Blockchain technology discussed in Section 2—we researched the state-of-the-art regarding open issues in this area. Below we explain how we researched and analyzed related work and then outline the steps used to evaluate OmniPHR model. Section 4 then describes OmniPHR architecture in detail and shows how we integrated it with blockchain technologies.

We first reviewed the state-of-the-art by analyzing articles related to OmniPHR, which implements blockchain solutions applied to health records. For this review, we used strings combining the PHR and EHR definitions with blockchain. We then submitted these strings to PubMed, Medline, CiteSeerX, Cochrane, HealthStar, Elsevier and Google Scholar, which are common portals that index scientific studies in the area of Health and Information Technology.

The selected related work studies are listed in Table 1, which lists the model name and reference, year of publication, health data standards, used framework, and if the study meets only organizational (EHR) or personal (PHR) health records. Table 1 underscores the fact that few studies dealt with the implementation of blockchain technology applied to health records. Moreover, even fewer articles presented results with systematic quantitative evaluations.

We analyzed the studies returned from these searches and selected only those studies that demonstrated blockchain implementations involving health records in actual databases. We discarded studies that only conducted simulated evaluations, as well as those that only dealt

Table 1
Related Work - Comparison of work with blockchain-based implementations.

Model & Year[a]	Health Data Std.[b]	FW[c]	EHR	PHR	Results
[23] Invisible Ink, 2015		E		✓	Built Certified Mail service as a sensitive user-data management platform
[24] FairAccess, 2016		E	✓	✓	Established an initial implementation with IoT and local blockchain
[25] Healthbitt, 2016	HL7/FHIR, ISO13606		✓	✓	Stores patient data in a distributed ledger allowing sharing with doctors
[26] HGD, 2016			✓	✓	Potential way to house and share health care data
[27] MyData, 2016			✓	✓	Provides useful information on business models and ecosystems
[28] CBTi, 2017		H	✓	✓	Data update and evaluation process worked normally
[29] D-CAM, 2017					Adds a modest overhead and can be scaled for large networks
[30] MedRec, 2017	HL7/FHIR	E	✓	✓	Describes the technical design and early-stage prototype
[31] MeDShare, 2017					Comparable to solutions for data sharing between cloud services
[32] Patientory, 2017	HL7/FHIR	E	✓	✓	Potential to eliminate friction and the costs of third-party intermediaries
[18] Ancile, 2018	HL7	E	✓	✓	Discusses interactions with patient's needs, providers and third parties
[33] FHIRChain, 2018	HL7/FHIR	E	✓	✓	Demonstrates a case study of a collaborative app for remote cancer care

[a] Models in ascending order by year.
[b] Health data standards.
[c] Platforms used in the solution, where E: Ethereum and H: Hyperledger Fabric.

with surveys or proposed solutions, i.e., without implementations that processed real data. Although the related work we examined was not restricted by date, we found relevant publications only from the year 2015 onwards since blockchain technologies have just recently been explored in the context of healthcare.

In addition to verifying the correct reunification of patients' scattered data, we evaluated non-functional requirements [43,44]. The requirements and statistical formulas used to collect the data are described below.

Initially, we counted the Mean Time Between Failures (*MTBF*):

$$MTBF = \frac{TotalWorkingTime - TotalBreakdownTime}{TotalBreakdownIncidences} \quad (1)$$

and Mean Time To Repair (*MTTR*):

$$MTTR = \frac{TotalBreakdownTime}{TotalBreakdownIncidences} \quad (2)$$

to compose the Availability (*A*):

$$A = \frac{MTBF}{MTBF + MTTR} \quad (3)$$

Finally, we evaluated the Performance (*P*) extraction arithmetic mean:

$$P = \frac{1}{n}\sum_{i=1}^{n} a_i \quad (4)$$

through the accounting of main memory, storage occupation, response time and throughput, where a compose the values and n the total of observations.

4. Blockchain model for OmniPHR

This section describes OmniPHR architecture and our application model. It also discusses key aspects of our OmniPHR implementation and scenarios applied in our evaluation environment. Our prototype follows the definitions proposed in OmniPHR model [21,22] and uses a distributed P2P network architecture with superpeers [45].

Our first article on this subject [21] dealt with OmniPHR model in a broader context [21]. In contrast, the current study expands and introduces improvements on OmniPHR's blockchain-based architecture and implementation, as well as evaluates OmniPHR prototype in three other production health organization scenarios. In particular, this article deals with aspects focused on OmniPHR's blockchain architecture and the impacts arising from the replication of health data.

OmniPHR's blockchain architecture model is comprised of the following two architectural layers:

(a) **Client modules**, which are installed in the health providers and in patient devices;
(b) **Server layer**, which is distributed in superpeers on a platform based on blockchain technology.

This architecture is formed via a private P2P network, where health records are organized into datablocks comprising a linked list and a distributed ledger of health data [46]. Fig. 1 depicts the architecture of OmniPHR prototype. This figure shows how clients communicate with the underlying blockchain platform via pull and push messaging [45]. This format enables all clients connected in the network to update their data proactively, i.e., datablocks can be sent and received automatically.

On the server, the blockchain platform is installed on a set of distributed superpeers. This private network stores datablocks within a KnowledgeBase, which is a non-relational NoSQL database based on a Graph or RDF DBMS. The KnowledgeBase itself is implemented using the *open*EHR ontology to store the data in a non-relational database based on graphs.

OmniPHR prototype also uses a parallel database in an entity-relationship (ER) model to store the datablocks in the format of archetypes, which is a relational DBMS. These archetypes follow the *open*EHR health data standard, which we adopt for communication and data storage in our blockchain network. The compositions of archetypes are the units that comprise the *open*EHR medical record structure [47]. The chained health datablocks in this database are used in forming the PHR smart contract.

Fig. 2 shows how OmniPHR prototype chains health datablocks together. Each datablock consists of (a) content formed by an archetype containing the health record, (b) a field containing the hash code representing the digital signature of the content of the archetype, and (c) a pointer with hash code that set the previous datablock. The first datablock is named the 'genesis block' and the 'previous hash' field points to no other datablock since it is the first node in the linked list.

OmniPHR prototype applies the blockchain smart contract feature [37] to verify and prevent violations of PHR data. Another highlight of OmniPHR prototype involves the role of each node in the blockchain network of health records. In particular, our prototype only allows superpeers located in the private network to evaluate the correctness of datablocks. Client nodes therefore only consume microservices provided by superpeers. Moreover, clients also produce content that is evaluated and distributed on the blockchain by superpeers.

Datablocks in OmniPHR prototype can be stored in the following two ways:

(a) Replicated in all nodes, following the approach adopted by the crypto-currency Bitcoin [48] or
(b) Using a replication algorithm, such as Chord [21], to replicate records only on certain nodes in the private blockchain network.

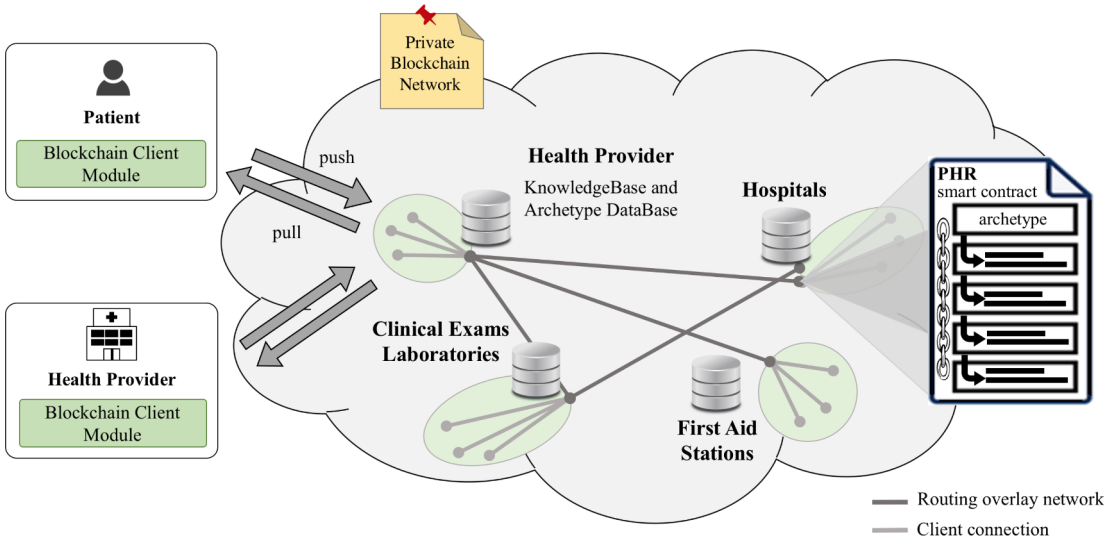

Fig. 1. The architecture of OmniPHR prototype.

OmniPHR model can be configured to support both forms of replication because when using the Chord algorithm we can set up to how many nodes we want to replicate the data blocks. The Chord algorithm was used to make this decision flexible. This flexibility is one of the main characteristics of the model, since it may not be desirable or even performative to replicate health blocks for all nodes in the network.

4.1. The structure and functionality of OmniPHR prototype

A distinguishing characteristic of OmniPHR prototype is its modular and distributed architecture based on components and microservices. We support the use of different components, as shown by the ecosystem in Fig. 3.

This figure should be viewed from the inside ring outwards. The core ring is PHR, which focuses on the integration of patient records. The second ring is based on a private blockchain network and data protocol following the *open*EHR or ISO 13606 standard. The third ring

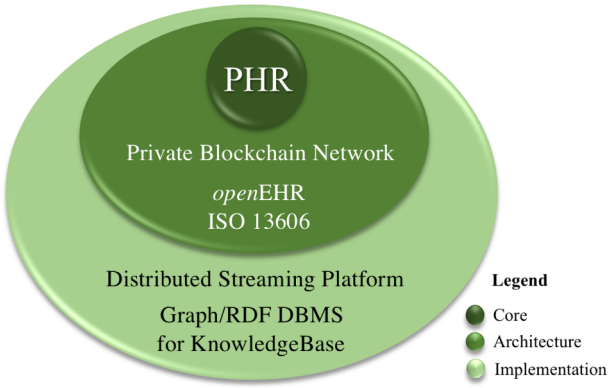

Fig. 3. OmniPHR application ecosystem.

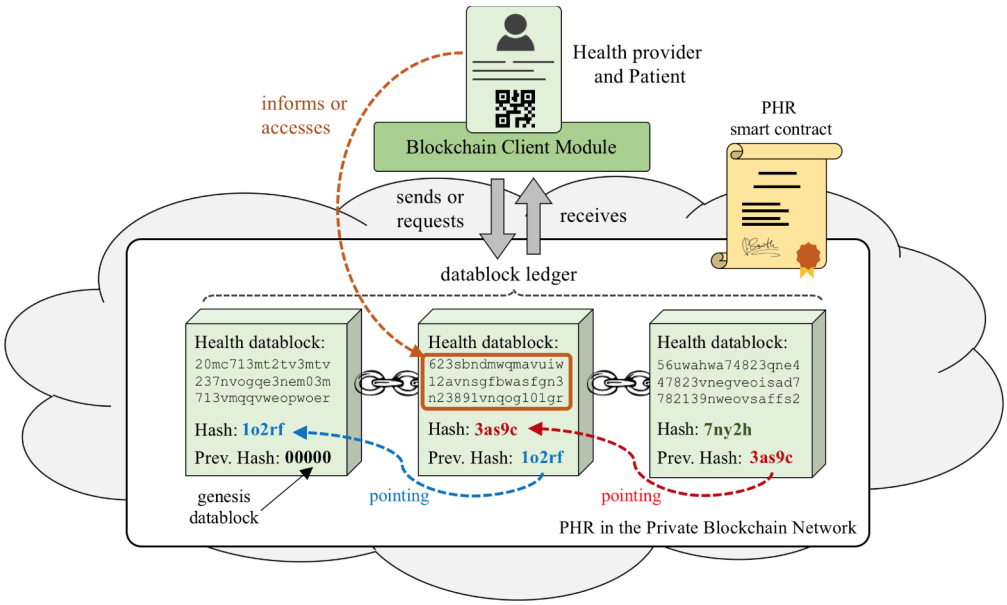

Fig. 2. PHR Blockchain in OmniPHR.

Table 2
Architectural choices.

Option	Potential benefits
Apache Kafka[a]	Distributed platform to store data safely in the distributed, replicated and fault-tolerant network
Apache Zookeeper[b]	Configuration and synchronization services
Apache Storm[c]	Real-time computing for data stream distribution
Apache Spark[d]	Engine for large-scale data processing
OpenLink Virtuoso[e]	Multi-model DB, supporting KB and ER store

[a] Apache Kafka – https://kafka.apache.org/.
[b] Apache Zookeeper – https://zookeeper.apache.org/.
[c] Apache Storm – http://storm.apache.org/.
[d] Apache Spark – https://spark.apache.org/.
[e] OpenLink Virtuoso – http://sourceforge.net/projects/virtuoso/.

used supports and implements the blockchain network via a distributed streaming platform, as well as a graph-based database or RDF. This streaming platform enables the distribution and integration of health records, whereas the database in Graph or RDF format forms the KnowledgeBase ontology.

To support OmniPHR, we evaluated several blockchain platforms that have been applied to support health records, including Hyperledger Fabric (www.hyperledger.org) [28] and Ethereum (www.ethereum.org) [49]. To gain greater control, however, we developed our own blockchain platform based on open APIs. This platform applies a private blockchain format, i.e., a trusted network, where only clients who are authorized to participate in the network can access health datablocks [50].

Table 2 summarizes all the platforms and tools employed in OmniPHR prototype. We use the Apache Kafka platform to distribute the datablocks in the superpeers network [51]. Kafka abstracts application concerns about data replication by extending its producer and consumer classes, which represent client nodes sending and receiving datablocks, respectively.

The Apache Kafka platform also acts as the message broker in OmniPHR architecture, which uses its messaging and queuing features to exchange data between nodes. Its high-performance partitioning and replication capabilities are also used to support real-time processing systems. Apache Storm is a real-time distributed computing system associated with Apache Kafka. In contrast, Apache Spark supports large-scale data processing, making OmniPHR architecture scalable and fault tolerant when distributing messages with health records.

We also use Apache Zookeeper in conjunction with the network resources provided by Apache Kafka. In particular, we use Zookeeper as a microservice interface to perform distributed configuration and synchronization of the messages that circulate in the blockchain network [52]. Apache Storm and Apache Spark services [53,54] are also applied to support scalable and responsive processing needs.

OmniPHR prototype contains classes that serve as an interface to access the blockchain, as well as store and remove content from the ledger. These classes enable the creation and maintenance of the PHR smart contract. Health data is stored in the open-source edition OpenLink Virtuoso database, which can store both relational storage (archetypes) and triple store (ontology) [55].

The Virtuoso database enables data querying via the SQL or SPARQL (RDF) query languages. OmniPHR prototype applies the Docker platform (www.docker.com) as the network container to provide a layer that abstracted and automated the virtualization [56]. To automate the building and deploying of code we use Gradle (gradle.org) [57].

To verify the transactions that circulate in the platform and to check with the content transmitted in the prototype, we exposed some microservices through RESTful web services and we used the HTTP client SoapUI (www.soapui.org) to test the unification of health records. Finally, we used the Apache JMeter tool (jmeter.apache.org) to represent the concurrent load of client nodes by performing insertions of new datablocks in the network or queries of existing blocks on the network.

4.2. Environment for evaluation methodology

To help load the KnowledgeBase of health data, we used the CaboLabs EHRServer [58] platform. This platform implements the *open*EHR standard in a relational database. Using data stored in archetypes—and following the *open*EHR standard—we distributed the records into datablocks in the blockchain.

To evaluate if the datablocks comprised a unified view of the health records, we evaluated the response time, the amount of memory occupied and the CPU usage, in a private blockchain network with 10 superpeers and up to 40,000 concurrent sessions. That is since the used database has data of 40,000 patients, and as a way to perform a stress test on the system, we have tested the blockchain to the limit of having at least one block of data from each patient searched or included concurrently. Each superpeer node consisted of Intel(R) Core(TM) i5, 3.30 GHz CPU, 4 cores, and 8 GB RAM. We also profiled OmniPHR prototype behavior by submitting different types of queries from an increasing series of client nodes.

Our evaluation environment used EHR and PHR for data query and health record manipulation [59]. As a load test scenario, therefore, we shared the use of the network blockchain by having half the client nodes query blocks of registers and the other half insert blocks into the blockchain network. For comparison purposes, we created the following three test scenarios that performed an increasing number of queries and inserts operations:

(a) **Light scenario**, which had datablocks triggered from 50 up to 500 concurrent sessions in the network;
(b) **Medium scenario**, which had datablocks triggered from 1000 up to 10,000 concurrent sessions;
(c) **Heavy scenario**, which had blocks of records transmitted from 13,000 up to 40,000 sessions on the network.

In the test scenarios, the number of users accessing the network was the number of concurrent sessions connected to the network, with the same increasing number of requests to the network [60].

We chose a private blockchain to restrict the management and access of network participants, thereby avoiding unauthorized sharing. This approach used mining resources and data evaluation more effectively by limiting access only to members of the network. In particular, evaluation in our private network was only performed by superpeers rather than burdening client nodes (which only produce and consume datablocks registered in the blockchain).

Two other factors justified our use of a private blockchain network: (a) to facilitate the traceability of updates and (b) to reduce intermediaries in data exchanges since the superpeers concentrate the execution of operations on health records. Moreover, we applied the *open*EHR standard since it stored data in meta-data blocks, which integrates seamlessly into the blockchain model. OmniPHR prototype accepts JSON and XML, though we applied XML predominantly within the blockchain and for the evaluation tests since XSD is useful to evaluate content and typing.

This study just focused on private blockchains instead of public blockchains due to data security and privacy issues, as well as due to the specific domain of healthcare targeted by OmniPHR. We, therefore did not allow access to other nodes since we handled sensitive health data that should only be shared by health providers and patients.

5. Results of performance experiments

This section analyzes the results obtained from our empirical evaluations and compares the results of our performance experiments with related work.

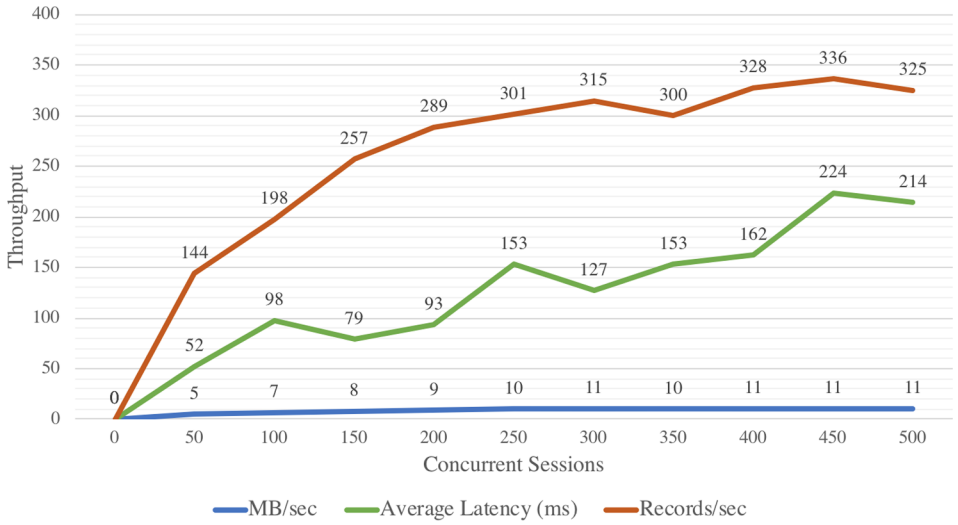

Fig. 4. Light load scenario.

5.1. Summary of our performance experiments

After configuring the settings to start each test scenario, we ran the network for nearly a week. During this period of ~160 h, we performed several load tests to evaluate the Light and Heavy scenarios described in Section 4.2. These load tests obtained the necessary values for the MTBF and MTTR calculations discussed in Section 2, obtaining results of 3.9586 and 0.0414, respectively.

Based on these results we calculated the Availability (A), where we obtain the value of 0.98964. The number of users accessing the network during the execution of the Light scenario was increased gradually, starting from 50 initial concurrent sessions until reaching the number of 500 users, as shown in Fig. 4, which depicts the Light scenario results.

The average load of blocks transmitted in the blockchain during the load test period is represented in Megabytes. The average response time (i.e., the average time of end-to-end latency that a client node requests to query a block or insert a new data in the blockchain and obtain the response) is represented in milliseconds. Fig. 4 shows the number of users accessing simultaneously the network in the Light scenario is increasing, as is the average load of records and the average response rate obtained. In this scenario, the load tests start from 50 concurrent sessions accessing the network, with a load of 5 MB/s of throughput, an average latency rate (end-to-end latency) of 52 ms and 144 records processed per second, reaching 500 users (concurrent sessions), with 11 MB/s of throughput in the network, one average response rate of 214 ms and 325 records/s.

In the second scenario of Fig. 5, we can see a range from 1000 to 10,000 concurrent sessions. Throughput ranges from 30 MB/s to 64 MB/s. We can observe that latency is stable, almost unchanged, going from 447 ms to 449 ms, i.e., less than half a second. And the number of records per second goes from 919 to 1917 records/s.

In the third scenario, represented by Fig. 6, we can see the results from the Heavy scenario. This scenario also shows an increasing number of users, average load of datablocks and response rate. The initial load was 68 MB/s with response time of 432 ms for 13,000 concurrent sessions until 40,000 were reached, with 77 MB/s of throughput, an average response rate of 404 ms and 2298 records/s. We can observe that even by increasing the number of concurrent sessions and throughput, the average response time remained stable.

The Table 3 presents data collected in the load test profiling for other non-functional requirements. The items analyzed were (a) CPU Usage, (b) Memory, (c) Disk throughput, (d) Network throughput (Sender) and (e) Network throughput (Receiver), for each of three scenarios evaluated (Light, Medium and Heavy). The variations of the data obtained in our tests for these requirements did not significantly impact the performance of the superpeers, except in the case of the heavy scenario, where there was a greater use of machine resources.

5.2. Analysis of our results

After we applied the methods presented in Section 3, the results from the MTBF and MTTR calculations comprised and demonstrated a 98% solution availability during load tests. These results were obtained by subjecting the model to three scenarios: one light with until 500 concurrent sessions accessing the network, one medium with up to 10,000 sessions and one heavy with up to 40,000 sessions. The scenarios used the same amount of patient data.

Although there were some periods with communication problems in the network (i.e., some nodes were not accessible), these periods were generally short. Our blockchain solution ensured that superpeers knew about the distribution of other nodes connected to them. In particular, since the Chord algorithm provided access to nodes with replicated content, superpeers could access other nodes with replicated data even though some nodes had communication problems. As a result, the overall operation of our solution was not impeded.

Another aspect is regarding smart contracts used to evaluate the permissions granted on the PHR. The smart contract can specify who can access PHRs and what permissions each client can get on the data. A smart contract on OmniPHR prototype, therefore, maintains the security and privacy of health records.

One difficulty faced in evaluating OmniPHR prototype stemmed from the challenge of submitting data to the model. To test the prototype we had to submit a considerable volume of health records to evaluate its performance. However, the results from the load tests shown in Fig. 6 indicated that in the heavy scenario response times stabilize around 500 ms. In general, OmniPHR prototype demonstrated average responses below one second. Although average response times grew with the load and number of users, response times remained low even as the loads increased. In particular, response times are nearly instantaneous with smaller loads and few simultaneous accesses. The network still responded quickly, however, even with larger simultaneous loads and accesses.

5.3. Limitations with our performance experiments

Our performance experiments did not cover the execution of business rules and inferences about records, such as specific evaluations of the content of patients' health records. Instead, we limited OmniPHR

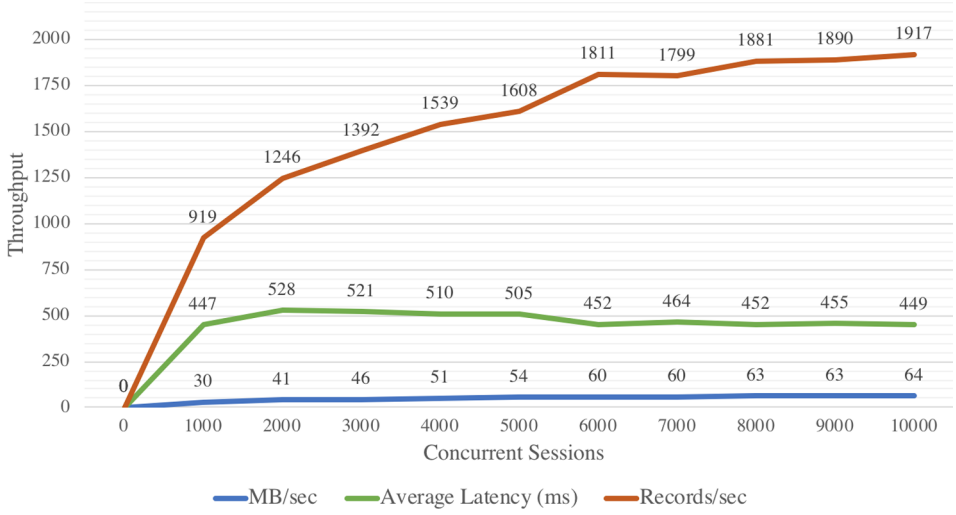

Fig. 5. Medium load scenario.

prototype to join datablocks that formed a unified view of patient data. In particular, our load tests only focused on evaluating the distribution and traffic of the blocks of records based on blockchain technology and the *open*EHR standard. We made this provision to isolate the performance evaluation of the blockchain solution without the interference of the usual business rule validations that health information systems have.

Another limitation is for image files, such as DICOM images. These images can occupy large spaces because of their size in megabytes. Replication of these files in the blockchain is not foreseen, although the location address is provided. In this way, the images are stored off-chain with a content hash code, and only the address where the images are located is replicated to the network.

We created the test scenarios in order to stress the system and verify that it remained stable without generating errors or crashes, such as OOM (Out Of Memory). We went to the limit of having at least one block of data from each registered patient handled concurrently. We tried to verify if the system remained stable of the original form as it was constructed, without using special tunings of optimization.

5.4. Comparison with related work

Table 1 summarizes results obtained by related work. Although these studies espouse the benefits of applying blockchain technologies to the healthcare domain through qualitative evaluations, few studies present empirical results to substantiate their claims. We, therefore, focus on qualitative analyses that evaluate the performance and efficacy of integrating health records via blockchain technologies. Although all projects use some blockchain technology in their implementations, only Healthbitt [25], MedRec [30], Patientory [32] and FHIRChain [33] applied at least one health data standard and focus on providing access to both health providers and patients.

Among the related work efforts presented in Table 1, seven used at least one of the two cross-industry platforms: Ethereum or Hyperledger. Most of these studies used Ethereum [18,23,24,30,32,33] as their blockchain platform and only one used Hyperledger Fabric [28]. The Ethereum platform uses the Ether (ETH) crypto-currency, whereas Hyperledger is not associated with any crypto-currency.

Related work focuses largely on describing how models can utilize blockchain technologies. In contrast, our research presented in this article focuses on demonstrating the viability of blockchain technologies by evaluating the behavior of OmniPHR prototype in production health record scenarios. Moreover, unlike related work that use conventional blockchain platforms like Ethereum or Hyperledger, OmniPHR uses the Chord algorithm, which supports replication.

Conventional blockchain platforms generally follow the original blockchain concept applied to crypto-currencies, which replicate data to all nodes in the network. In contrast, the Chord replication algorithm

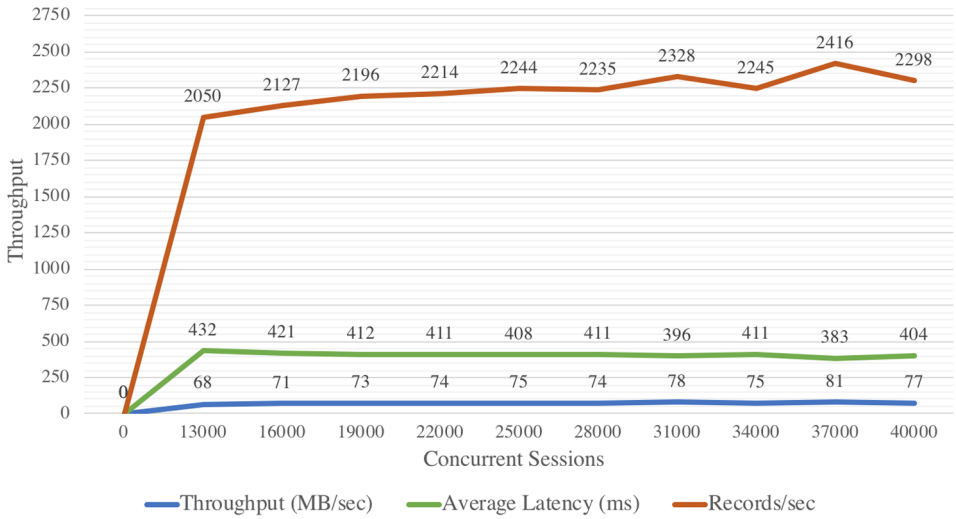

Fig. 6. Heavy load scenario.

Table 3
Performance scenarios (average usage value per node).

Rated item	Light load	Medium load	Heavy load
CPU usage average	0.3 GHz (10%)	0.75 GHz (25%)	1.5 GHz (45%)
Memory	0.8 GB (10%)	2.08 GB (26%)	3.6 GB (45%)
Disk throughput	0.1 MB/s (0.1%)	4 MB/s (4%)	10 MB/s (10%)
Network throughput (Sender)	0.1 MB/s (0.1%)	4 MB/s (4%)	10 MB/s (10%)
Network throughput (Receiver)	0.4 MB/s (1.5%)	2 MB/s (10%)	4.5 MB/s (21%)

enables finer-grained control over how much, how, and where to replicate the data, thereby enabling more granular control of replications. Our results in Section 5. A show that Chord optimizes performance, although data redundancy is reduced. In addition, by storing datablocks in ontology format, i.e., in the Ontology Web Language (OWL), the KnowledgeBase enables the creation of semantic rules that allow inferences about possible patient health problems.

6. Concluding remarks

This article presented the prototype implementation and evaluation of OmniPHR architecture model that integrates distributed health records using blockchain technology and the *open*EHR interoperability standard. OmniPHR prototype comprises a novel blockchain-based design that optimizes health data replication across computing nodes. We evaluated the performance of OmniPHR prototype by subjecting it to loads of thousands of concurrent sessions transmitting datablocks on a network of 10 superpeers. We also evaluated implementation strategies related to the replication of health-oriented blockchain solutions to promote the unification of patient health data.

The following are a summary of the lessons learned from conducting our research on OmniPHR:

(a) Combining the *open*EHR standard with blockchain technologies created a unified and interoperable view of health data. Even with some limitations, such as not executing business rules on the prototype (since it is not a complete system), we observed promising results of the architectural model using our private blockchain platform.
(b) Applying the Chord algorithm for directed and limited data replication is a more scalable alternative than conventional cryptocurrency platform replication models, where all nodes receive all data. Chord's scalability is a critical factor to effectively support health data. In particular, it enables data replication with restricted access, providing control and management by patients and healthcare professionals.
(c) The results of our empirical evaluations showed that OmniPHR blockchain architecture provided adequate network level performance. It, therefore, appears that patient health records can be integrated effectively via a blockchain network using technologies applied to the treatment of large masses of data and an interoperable health data standard.

In future work, we plan to evolve OmniPHR prototype to incorporate additional databases and conduct additional tests to evaluate its performance in even more scalable and realistic production environments. Other evaluations we plan to conduct involve data security and privacy, especially in the case of external access to private blockchain networks.

Competing interests

The authors have no competing interests to declare.

Contributors

All authors contributed to the conception of the work, revising and criticizing the content. All authors approved the manuscript for publication.

Acknowledgments

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

References

[1] E. Jamoom, N. Yang, E. Hing, Adoption of certified electronic health record systems and electronic information sharing in physician offices: United States, 2013 and 2014, NCHS data brief (236) (2016) 1–8.
[2] P. Yadav, M. Steinbach, V. Kumar, G. Simon, Mining electronic health records (ehrs): a survey, ACM Comput. Surv. 50 (6) (2018) 85:1–85:40. Available: http://doi.acm.org/10.1145/3127881.
[3] ISO, Health informatics – guidance on health information privacy education in healthcare organizations, Technical Report, no. ISO/TR 18638:2017, 2017. Available: <https://www.iso.org/obp/ui/#iso:std:iso:tr:18638:ed-1:v1:en:term:3.20>.
[4] S. Wass, V. Vimarlund, Same, same but different: Perceptions of patients online access to electronic health records among healthcare professionals, Health Informatics J. (2018) p. 1460458218779101.
[5] J.P. New, D. Leather, N.D. Bakerly, J. McCrae, J.M. Gibson, Putting patients in control of data from electronic health records, BMJ 360 (2018) j5554.
[6] E.W. Ford, B.W. Hesse, T.R. Huerta, Personal health record use in the united states: forecasting future adoption levels, J. Med. Internet Res. 18 (3) (2016).
[7] M.A. Alyami, Y.-T. Song, Removing barriers in using personal health record systems, 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), IEEE, 2016, pp. 1–8.
[8] T. Heart, O. Ben-Assuli, I. Shabtai, emr and ehr integration: a more personalized healthcare and public health policy, Health Policy Technol. 6 (1) (2017) 20–25.
[9] H.S. Gardiyawasam Pussewalage, V.A. Oleshchuk, A distributed multi-authority attribute based encryption scheme for secure sharing of personal health records, Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, ACM, 2017, pp. 255–262.
[10] C. Showell, Barriers to the use of personal health records by patients: a structured review, PeerJ 5 (2017) e3268.
[11] K. Kaur, R. Rani, Managing data in healthcare information systems: many models, one solution, Computer 48 (3) (2015) 52–59.
[12] M.D. Krasowski, D. Chudzik, A. Dolezal, B. Steussy, M.P. Gailey, B. Koch, S.B. Kilborn, B.W. Darbro, C.D. Rysgaard, J.A. Klesney-Tait, Promoting improved utilization of laboratory testing through changes in an electronic medical record: experience at an academic medical center, BMC Med. Informatics Decis. Making 15 (1) (2015) 11.
[13] L.L. Fragidis, P.D. Chatzoglou, Development of nationwide electronic health record (nehr): an international survey, Health Policy Technol. 6 (2) (2017) 124–133.
[14] E. Chiauzzi, C. Rodarte, P. DasMahapatra, Patient-centered activity monitoring in the self-management of chronic health conditions, BMC Med. 13 (1) (2015) 77.
[15] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008.
[16] P. Zhang, M. Walker, J. White, D.C. Schmidt, G. Lenz, Metrics for assessing blockchain-based healthcare decentralized apps, 2017. Available: <http://www.dre.vanderbilt.edu/schmidt/PDF/IEEE-Healthcom-2017.pdf>.
[17] D. Randall, P. Goel, R. Abujamra, Blockchain applications and use cases in health information technology, J. Health Med. Informat. 8 (276) (2017) 2.
[18] G.G. Dagher, J. Mohler, M. Milojkovic, P.B. Marella, Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology, Sustain. Cities Soc. 39 (2018) 283–297.
[19] T.-T. Kuo, H.-E. Kim, L. Ohno-Machado, Blockchain distributed ledger technologies for biomedical and health care applications, J. Am. Med. Informatics Assoc. 24 (6) (2017) 1211–1220.
[20] J.R. Goldim, S. Gibbon, Between personal and relational privacy: understanding the work of informed consent in cancer genetics in brazil, J. Community Gen. 6 (3) (2015) 287–293.

[21] A. Roehrs, C.A. da Costa, R. da Rosa Righi, Omniphr: A distributed architecture model to integrate personal health records, J. Biomed. Informatics 71 (2017) 70–81.

[22] A. Roehrs, C.A. da Costa, R. da Rosa Righi, S.J. Rigo, M. Wichman, Toward a model for personal health records interoperability, IEEE J. Biomed. Health Informatics (2018).

[23] A. Lazarovich, Invisible ink: blockchain for data privacy (Ph.D. dissertation), Massachusetts Institute of Technology, 2015. Available: <http://hdl.handle.net/1721.1/98626>.

[24] A. Ouaddah, A. Abou Elkalam, A. Ait Ouahman, Fairaccess: a new blockchain-based access control framework for the internet of things, Security Commun. Netw. 9 (18) (2016) 5943–5964.

[25] D.K. Rono, A restful e-health interoperability platform: case of nairobi county health facilities (Ph.D. dissertation), Strathmore University, 2016.

[26] X. Yue, H. Wang, D. Jin, M. Li, W. Jiang, Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control, J. Med. Syst. 40 (10) (2016) 218.

[27] K. Laura, T. Koivumäki, S.D. Saraniemi, Business models for platform operators in mydata based ecosystem–context preventive healthcare, Marketing, 2016. Available: <http://jultika.oulu.fi/files/nbnfioulu-201605121721.pdf>.

[28] D. Ichikawa, M. Kashiyama, T. Ueno, Tamper-resistant mobile health using blockchain technology, JMIR mHealth and uHealth 5 (7) (2017) Available: <http://mhealth.jmir.org/2017/7/e111/>.

[29] M. Henze, B. Wolters, R. Matzutt, T. Zimmermann, K. Wehrle, Distributed configuration, authorization and management in the cloud-based internet of things, 2017. Available: <https://www.comsys.rwth-aachen.de/fileadmin/papers/2017/2017-henze-trustcom-dcam.pdf>.

[30] A.C. Ekblaw, Medrec: blockchain for medical data access, permission management and trend analysis (Ph.D. dissertation), Massachusetts Institute of Technology, 2017. Available: <https://dspace.mit.edu/handle/1721.1/109658>.

[31] Q. Xia, E.B. Sifah, K.O. Asamoah, J. Gao, X. Du, M. Guizani, Medshare: Trust-less medical data sharing among cloud service providers via blockchain, IEEE Access 5 (2017) 14 757–14 767. Available: https://doi.org/10.1109/ACCESS.2017.2730843.

[32] C. McFarlane, M. Beer, J. Brown, N. Prendergast, Patientory: A healthcare peer-to-peer emr storage network v1.0, 2017. Available: <https://www.patientory.com/wp-content/uploads/2017/04/Patientory_Whitepaper-1.pdf>.

[33] P. Zhang, J. White, D.C. Schmidt, G. Lenz, S.T. Rosenbloom, Fhirchain: applying blockchain to securely and scalably share clinical data, Comput. Struct. Biotechnol. J. 16 (2018) 267–278.

[34] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.

[35] D. Kraft, Difficulty control for blockchain-based consensus systems, Peer-to-Peer Netw. Appl. 9 (2) (2016) 397–413.

[36] C. Stagnaro, White paper: Innovative blockchain uses in health care, 2017. Available: <http://www.freedassociates.com/wp-content/uploads/2017/08/Blockchain_White_Paper.pdf>.

[37] N. Szabo, Smart contracts: Building blocks for digital markets, 1996. Available: <http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html>.

[38] J.C. Mandel, D.A. Kreda, K.D. Mandl, I.S. Kohane, R.B. Ramoni, Smart on fhir: a standards-based, interoperable apps platform for electronic health records, J. Am. Med. Informatics Assoc. 23 (5) (2016) 899–908.

[39] S. Sachdeva, S. Batra, S. Bhalla, Evolving large scale healthcare applications using open standards, Health Policy Technol. 6 (4) (2017) 410–425.

[40] Y. Aliakbarpoor, S. Comai, G. Pozzi, Designing a hl7 compatible personal health record for mobile devices, 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI), IEEE, 2017, pp. 1–6.

[41] G.-H. Ulriksen, R. Pedersen, G. Ellingsen, Infrastructuring in healthcare through the openehr architecture, Comput. Supp. Cooper. Work (CSCW) 26 (1–2) (2017) 33–69.

[42] M.d.C. Legaz-García, M. Menárguez-Tortosa, J.T. Fernández-Breis, C.G. Chute, C. Tao, Transformation of standardized clinical models based on owl technologies: from cem to openehr archetypes, J. Am. Med. Informatics Assoc. 22 (3) (2015) 536–544.

[43] M. Galster, E. Bucherer, A taxonomy for identifying and specifying non-functional requirements in service-oriented development, IEEE Congress on Services-Part I, 2008, IEEE, 2008, pp. 345–352.

[44] L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos, Non-functional Requirements in Software Engineering Vol. 5 Springer Science & Business Media, 2012.

[45] G. Coulouris, J. Dollimore, T. Kindenberg, G. Blair, Distributed systems: Concepts and design, 2012.

[46] M. Walport, Distributed ledger technology: Beyond blockchain, UK Government Office for Science, 2016.

[47] B. Li, J. Li, X. Lan, Y. An, W. Gao, Y. Jiang, Experiences of building a medical data acquisition system based on two-level modeling, Int. J. Med. Informatics (2018).

[48] M. Asplund, J. Lovhall, S. Nadjm-Tehrani, In-store payments using bitcoin, 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, 2018, pp. 1–6.

[49] A. Ekblaw, A. Azaria, J.D. Halamka, A. Lippman, A case study for blockchain in healthcare:medrec prototype for electronic health records and medical research data, in: Proceedings of IEEE Open & Big Data Conference, vol. 13, 2016, p. 13.

[50] V. Dhillon, Designing decentralized ledger technology for electronic health records, Telehealth and Medicine Today, vol. 2, no. 6, 2017. Available: <http://www.telhealthandmedtoday.com/designing-decentralized-ledger-technology-for-electronic-health-records/>.

[51] V.-D. Ta, C.-M. Liu, G.W. Nkabinde, Big data stream computing in healthcare real-time analytics, 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), IEEE, 2016, pp. 37–42.

[52] G. Wang, J. Koshy, S. Subramanian, K. Paramasivam, M. Zadeh, N. Narkhede, J. Rao, J. Kreps, J. Stein, Building a replicated logging system with apache kafka, Proc. VLDB Endowment 8 (12) (2015) 1654–1655.

[53] A. Jain, Mastering apache storm: Real-time big data streaming using kafka, hbase and redis, 2017.

[54] M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M.J. Franklin, et al., Apache spark: a unified engine for big data processing, Commun. ACM 59 (11) (2016) 56–65.

[55] D.J. Odgers, M. Dumontier, Mining electronic health records using linked data, AMIA Summits on Translational Science Proceedings 2015 (2015) 217.

[56] T. Adufu, J. Choi, Y. Kim, Is container-based technology a winner for high performance scientific applications? Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific, IEEE, 2015, pp. 507–510.

[57] H.K. Ikkink, Gradle Dependency Management, Packt Publishing Ltd, 2015.

[58] P.P. Gutiérrez, "Cabolabs ehrserver: Service-oriented openehr repository for clinical data with composition commit, query and retrieve capabilities," 2018. Available: <https://github.com/ppazos/cabolabs-ehrserver>.

[59] G.H. Shah, J.P. Leider, B.C. Castrucci, K.S. Williams, H. Luo, Characteristics of local health departments associated with implementation of electronic health records and other informatics systems, Public Health Rep. 131 (2) (2016) 272–282.

[60] R. Morabito, A performance evaluation of container technologies on internet of things devices, 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2016, pp. 999–1000.