A Storage Architecture for Blockchain-Based Healthcare Systems

Diederik Heijbroek[1] , **Chhagan Lal**[1] , **Mauro Conti**[1]
[1]TU Delft

Abstract

Patient monitoring and clinical trial management generate continuous large volumes of healthcare data, causing the healthcare industry to be a data-intensive domain. Existing data storage techniques and access mechanics in the healthcare domain exhibit several challenges related to data security, patient privacy, and interoperability. Blockchain technology, together with the support from smart contracts, is considered a proper facilitator for secure and efficient healthcare data storage and sharing. Blockchain technology has unique features, such as decentralization, trustlessness, immutability, traceability, and transparency.

Ongoing efforts show promising results considering blockchain technology to improve different aspects of healthcare data sharing and data management. However, all of these initiatives are still in the initial stages and lack technical details. This specifically pertains to the lack of investigation and evaluation of existing storage methods and techniques. Therefore, we will provide a comprehensive evaluation of different storage methods and techniques in ongoing efforts. We introduce a storage architecture for a possible blockchain-based healthcare system. The research proposals are evaluated considering the different storage methods and techniques. Additionally, we investigate the presence of six key requirements to provide a good storage solution.

These requirements consist of data location, storage security, access mechanisms, third parties, storage purpose, operations and data integrity. None of the investigated systems meet all six requirements identified in this study. Therefore, we have proposed a storage architecture of our own.

Keywords Blockchain, distributed, centralized, on-chain, off-chain, storage.

1 Introduction

The healthcare industry generates continuous large volumes of healthcare data, such as patient monitoring and clinical trial management, causing the industry to be a data-intensive domain [1]. The data needs to be shared among different medical facilities for various purposes, such as collaborative research and personalized healthcare services [2]. Existing data storage techniques and access mechanics in the healthcare domain exhibit several challenges [3]. The key challenges include (I) storage security [4], (II) privacy [5], (III) interoperability and (IV) accessibility. The patients' data is private and needs to be secured. Thus, the aforementioned challenges must be maintained.

Blockchain (BC) technology could become the solution to the aforementioned challenges by providing transparency, immutability, confidentiality and auditability of managing electronic medical records (EMRs) [6]. Large volumes of healthcare data (HD) have a major impact on the size of the BC. Running a BC costs space and time. When the BC becomes too large, technical and financial impracticalities occur. Due to the rapidly increasing scale of the healthcare domain, we need to investigate alternative storage solutions for the BC system [7].

Ongoing research efforts address challenges related to secure storage of large volumes of HD in BC-based solutions [8][9] [10]. They indicate that the large volumes of HD require alternative solutions where the data is not stored on the BC itself. In these alternative solutions, the HD is stored off-chain and the BC only stores its corresponding metadata tags and hash values. Furthermore, they provide methods for high availability, integrity and confidentiality of the HD. These requirements can for instance be achieved by distributed storage of the HD. MedRec [11] and MedChain [10] particularly focus on the storage of EMRs. Moreover, Liu et al. [12] use BC along with other techniques to support efficient access control to medical records.

For our research, we evaluate different storage techniques and investigate which requirements are needed for secure data storage, collection and accessibility. We then investigate the presence of the requirements in existing literature. During this investigation, also the security and privacy (S&P) parameters that have not been discussed in this literature are addressed. Specifically, in this paper, we aim to address the following research question:

“Given the blockchain-based healthcare system, what is the most optimal architecture considering different storage methods?”

The aforementioned research work shows promising results considering the use of blockchain to improve different aspects of HD sharing and HD management techniques. They provide the general structure of a blockchain-based healthcare system (BBHS). However, all of these initiatives are still at initial stages and lack technical details. Especially storage techniques and storage methods have not been evaluated properly. The proposals make assumptions on what storage methods and techniques fit with their proposed BBHS. A clear explanation and evaluation on why this would be the optimal storage method for the proposed BBHS is not yet available. We aim to find more information about the different storage methods and techniques in existing literature. With that information we can investigate and evaluate the different security and privacy parameters corresponding with the different storage methods and techniques. Finally, we investigate what S&P parameters remain unanswered in literature and come up with a proposal for a BBHS ourselves. A comprehensive comparison of the different techniques does not yet exist and these parameters indicate what future work is required for BBHSs.

This research paper is organized as follows: Section 2 provides some background on BC. Furthermore, we address the advantages and disadvantages of the storage methods and techniques in medical data sharing applications. In Section 3, the key requirements for a BBHS are listed. We investigate the presence of these requirements in existing literature. Moreover, we propose the design of a solution that includes these requirements. Section 4 addresses the remaining challenges of medical data sharing systems and suggests required future work. Furthermore, Section 5 discusses the responsible research of the system. Lastly, Section 6 summarizes the subtasks of this research and the outcomes of these sub-tasks.

2 Background and related work

This section provides background of BC and the use of smart contracts (SCs) in a BC system. Furthermore, we investigate storage techniques in existing literature. Lastly, we will discuss advantages and disadvantages of proposed storage methods in related work.

2.1 Blockchain and Smart Contracts

Blockchain (BC) is a chain of data blocks and functions as a distributed ledger. BC technology provides immutability and integrity of data on the BC. This is because each block contains the hash value of the previous block. BC technology is used for networks where peers do not trust each other. Its functionalities ensure that the data is tamper proof and consensus mechanisms are used to validate new transactions. Current consensus mechanisms are Proof-of-Work, Proof-of-Stake and Practical Byzantine Fault Tolerance (PBFT). Permissionless (public) BC networks, such as Bitcoin [13], allow for anyone to join the network and participate. BC networks can also be private and are called permissioned or consortium BC. In these private networks, a third party keeps track of users and certificates in order to assign roles. Unfortunately, data on the BC is visible to every peer who stores a copy of the ledger. The European GDPR indicates that user privacy must be guaranteed, which can be achieved by the use of several techniques, such as Zero Knowledge Proofs (ZKPs) [14]. ZKPs provide interesting techniques to ensure confidentiality and privacy of data.

BC technology allows for functions to be executed on the ledger. This can be achieved by using smart contracts (SCs). A SC consists of code which can be executed on the ledger once a set of preliminaries hold. In most cases, this set consists of consent by different stakeholders. Due to the immutability of BC technology, it is feasible to allow SCs in BC-based systems. The SCs are deployed and protected by the BC. Unique benefits of SCs are that its code is immutable, consesus nodes execute without mutual trust and the SC enables automation of tasks.

The use of BC and SCs is rapidly increasing in various real-world applications and domains. Therefore, they are a promising target for cybercriminals. Advanced techniques and tools are required to maintain security in the BC application.

2.2 Data storage techniques

There are many ongoing efforts that propose a blockchain-based healthcare system (BBHS). In these proposals an improvement on access control, security or efficiency is given. Almost all of the proposals indicate that off-chain storage is the best method for storing and acccessing HD. For off-chain storage, different storage techniques can be implemented, such as centralized (cloud-based) or distributed (decentralized) storage. For centralized storage, either one or multiple peers can provide HD in a central location. Distributed storage splits the HD into multiple pieces of data. Each piece has multiple copies stored at other HD providers. Therefore, distributed storage techniques provide more integrity of the HD than centralized storage techniques. However, the level of complexity increases rapidly.

Centralized storage techniques

The establishment of a medical data center requires high construction costs and professional technical support. Wang et al. propose a cloud storage technology [15] to solve this. Cloud storage is essentially a cloud computing system with a large storage capacity. Advantages are fast transmission, convenient sharing, storage capacity, low cost, easy access, and dynamic association. Cloud storage can serve as a platform for information sharing between remote hospitals and solves the problem of remote collaborative diagnosis [16]. The medical cloud system not only provides great convenience for doctors and patients, but also helps patients to better control their own condition by providing easy access to their EMRs. However, when users store EMR data on the cloud server, the data suffers a variety of security threats [17] involving the privacy, integrity and authentication of EMR data. Therefore, a lot of complex S&P techniques are required to maintain a private and secure centralized cloud storage solution.

Xia et al. propose a centralized BC-based data sharing framework called Blockchain-Based Data Sharing (BBDS) [18]. BBDS sufficiently addresses the access control challenges associated with sensitive data stored in the cloud. This can be achieved by using immutability and built-in autonomy properties of the BC. BBDS allows users to access EMRs from a shared repository upon successful verification of their identity. They employ the identity-based authentication and key agreement protocol proposed in [19] to provide user membership and authentication. However, their secure sharing of sensitive medical information is limited to invited and verified users. Moreover, their proposal of using asymmetric encryption algorithms to encrypt medical information does not seem to be a beneficial option due to the poor encryption/decryption performance of asymmetric encryption.

Distributed storage techniques

Chen et al. propose HyperBSA [8]. HyperBSA focusses on improving the I/O performance of distributed storage systems. Distributed storage is common in consortium BC systems. There are three main components or modules in HyperBSA: Filelog to deal with continuous data, Multi-level Cache Mechanism with Persistence Policy (MCMPP) to deal with state data and a distributed extension of the architecure. They perform a number of experiments comparing their I/O performance with the performance of existing consortium blockchain systems. The results of these experiments indicate that FileLog is a lot faster in time with read and write operations in comparison to LevelDB (a commonly used database for BC systems). Furthermore, MCMPP improves the efficient storage of state data to a certain extent. However, the experiment was not realistic when considering the amount of users and the large volumes of HD. Therefore, we cannot draw a proper conclusion on the actual improvements on the performance of the proposed system.

Distributed storage solutions deal with the problem of offline data providers, as HD is stored on multiple locations. Storj [20] provides valuable solutions for distributed storage systems. The technique uses distributed hash tables (DHTs) [21] to keep track of the different pieces of HD. Storj is a commercial project, where storage location is made available and capacity can be purchased by consumers. However, the system does not handle data sharing possibilities between consumers. In order to do so, a user needs to pull a local copy of the data and send it to another user. The other user then needs to re-upload the data into the network in order to both have access. This is highly inefficient, especially for large volumes of HD.

Do et al. propose BlockDS [22] to combat this use-case. BlockDS stores a piece of data across the DHT with primary access rights given to the owner of the data to assign access rights to other consumers. The data can be accessed by the consumers via a static set of keywords. These keywords prove that the users are allowed to read the data. Moreover, the keywords are used to decrypt the encrypted [22]. A user could potentially download a local copy of the file and redistribute it, meaning the original owner of the data no longer controls it.

Distributed databases (DBs) are similar to distributed storage techniques using DHTs. The difference with a distributed DB is the addition of a query language on top of the back-end of the storage. The query language allows more operations to be executed by consumers. EthDrive [23] and BigchainDB [24] are existing solutions that use DDBs integrated with BC. A distributed DB functions almost the same as a classic DB because of its query language. In order to solve problems of scalability, additional nodes can be added to a distributed DB in order to further increase storage and processing capacity. However, the large volumes of HD require a lot of nodes, which cause a rapid increase of energy consumption and decreases the overall efficiency of the system.

Another interesting technology for distributing HD is IPFS [25]. IPFS is a distributed Peer-to-Peer (P2P) storage network for storing and accessing files, websites, applications, and data. Content is accessible through peers located anywhere in the world, whom might relay information, store it, or do both. We will not go into depth in the techniques behind IPFS, as these techniques are out of the scope of this project. IPFS is present in many ongoing efforts and seems to be a promising solution. However, there are no existing evaluations of IPFS's performance in a large scale BBHS.

Security techniques

The on-chain metadata and hash link to off-chain HD needs to be stored securely. Kosba et al. propose the Hawk system [26] to ensure secure on-chain data. Hawk is a decentralized smart contract (SC) system that provides transactional privacy from the public's view. A Hawk programmer can write a private SC in an intuitive manner without having to implement cryptography. Their compiler automatically generates an efficient cryptographic protocol where contractual parties interact with the BC. However, Hawk requires a manager to facilitate the execution. The manager has access to the plaintext contract details, which Hawk is supposed to protect. This defeats the purpose of the protocol.

Zyskind et al. propose Enigma [27] as a solution to the aforementioned problem. Enigma is a P2P network, which allows multiple parties to share data and code securely. Additionally, the system allows for distributed computation of complex code. Data is split between different nodes that compute functions together without leaking information to other nodes. No single party ever has access to data in its entirety; instead each party has a meaningless piece of the data. The technical specifics by which the computation and information sharing is implemented are beyond the scope of this paper. Enigma indicates that their system is scalable. The proposal of Enigma sounds very promising. However, they only refer to smaller BC-based systems, such as E-voting and cryptocurrencies. Moreover, there is no evaluation that can prove Enigma's scalability. Therefore, we cannot conclude that Enigma's complexity would be beneficial for the large volumes of HD.

Conclusion

Centralized storage techniques provide fast transmissions, convenient sharing, easy access and low costs for a large storage capacity, which is necessary for the volume of HD. Decentralized storage techniques are implemented to increase security and integrity of the HD, which current centralized systems are not yet able to achieve. In order for decentralized storage techniques to have similar efficient operations as centralized storage techniques, more complex systems are required which are not yet evaluated for large scale BBHSs. For now, centralized storage in combination with existing security techniques are suitable for a BBHS. In the future however, decentralized storage becomes inevitable due to the risk of hackers being able to sell the HD to employers.

2.3 Related work

Section 2.1 indicates that off-chain storage solutions are more suitable than on-chain storage solutions. Being compliant with the European GDPR means that no private data should be stored on a permissionless blockchain, to protect against the potential risk of decryption in the future. In this section, we discuss ongoing efforts that implement BC-based systems for the healthcare domain. We specifically focus on how off-chain storage methods are implemented and what disadvantages still remain.

Xu et al. propose a privacy-preserving scheme for fine-grained access control of large-scale off-chain HD based on BC called Healthchain [28]. Their proposal introduces two BCs for both users' HD and doctors' diagnoses. They also decouple the encrypted data and the corresponding keys to achieve flexible key management. Identity keys are stored in transactions for accountability. Furthermore, Healthchain offers a privacy feature to easily revoke doctors' access to EMRs. Healthchain works with single identities of users and doctors. However, it is hard for a patient to decide who specifically should have access to the EMR. The use of roles would make it a lot easier to give consent and allow third parties to read alternative versions (e.g., anonymous) of the patient's EMR. Moreover, the introduced BCs do not perform as a ledger, which is necessary for a more secure consortium BC system.

Fan et al. propose MedBlock [29], which is a hybrid BC-based architecture to secure EMRs. MedBlock successfully resolves the problem of large-scale data management and sharing in an EMR system. Also, data sharing and collaboration via BC can help hospitals get a prior understanding of patients' medical history before consolation. Its consensus protocol is a variant of the PBFT consensus protocol. However, the authors do not explicitly explain the access control policy to allow third-party researchers to access medical data. Moreover, a third party is required to manage certificates and accessibility. The third party also has access to the blockchain for supervision.

Dagher et al. propose Ancile [30], which is an Ethereum-based record management system that utilizes SCs for access control. Ancile keeps the patients' EMRs in existing DBs of providers and reference addresses to these records along with permissions stored in the BC network. They propose an access control policy which allows third-party researchers to access HD. However, the authors do not provide details about their consensus protocol and incentive mechanism. Additionally, no experimental results are explained in the paper. The authors only execute a performance analysis of computational costs to compare Ancile and MedRec.

MedRec [11] is a decentralized EMR management system proposed by Azaria et al. They manage permissions, authorization and data sharing between participants. MedRec proposes an additional encryption in the off-chain synchronisation steps, safeguarding against accidental or malicious content access. Additionally, they distribute the provider's identity across the network. Even though MedRec takes security into account, they do not concern security at the level of provider DBs. Other problems that MedRec has not yet taken into account are scalability, vulnerabilities in SCs and they do not consider third parties, such as researchers. Moreover their proof-of-authority protocol gives all power to data providers and not all events are recorded on the ledger.

The current existing BC-based system which is most suitable for the healthcare system and the closest to a succesfully working system is Hyperledger Fabric [**?**]. Hyperledger Fabric is one of the most prominent permissioned BCs which offers significantly higher throughput compared to Bitcoin and Ethereum, and achieves a reasonably fast consensus. However, Hyperledger Fabric is still of limited applicability for large scale IoT applications. The large number of transactions in a BBHS can quickly degrade the overall performance.

Conclusion

There are some promising solutions for secure and accountable BC-based systems. Hyperledger Fabric comes closest to the general structure of a BBHS, but does not perform well for large scale BC applications. Many solutions still need a lot of work to deal with the scale of HD. The ongoing efforts do not focus on storage methods and techniques. Therefore, proper in-depth investigations on the actual performance of their proposal is required. Perhaps their proposal performs better using a centralized cloud-based storage technique if S&P can be maintained.

3 Research methodology

This Section provides the key requirements for data collection, storage and accessibility. Additionally, an explanation for the importance of each requirement regarding storage is given. Second, we will investigate the presence of these requirements in existing work. At last, we will propose a storage system for a BBHS where these requirements are present.

3.1 Data storage requirements

For secure data storage, collection and accessibility, there are some key requirements that we have to take into account. These are:

- R_1**: Data location and format** is an important feature of data storage. Currently, there is no general structure for an EMR, which makes it hard for doctors and researchers to find the HD they require. Also, a patient's HD gets scattered over multiple locations when a patient visits multiple medical facilities. Therefore, we recommend to have one EMR per patient. Furthermore, these EMRs need to have a global and general structure. Another solution would be a query handler where DB manager nodes return the correct HD after a global search request. In general, a single request for the network should return the correct HD where little processing of the HD is required.

- R_2: **Storage security and access mechanisms** are also an important requirement for data storage, because EMRs contain private data. We can assume that existing security techniques are sufficient to secure stored HD. However, most security issues occur at the access level. If a malicious user pretends to be a doctor, the user can gain access to certain HD. When access mechanisms are not set-up correctly, a DB manager will decrypt the HD for that malicious user. Therefore, proper mechanisms to access off-chain storage are required.

- R_3: **Third parties** are an interesting concept for BBHSs, because BC assumes that none of the peers trust each other. In existing work, third parties most often refer to researchers or doctors from other medical facilities. Consent mechanisms are applied providing third parties with access to EMRs. A patient can for instance allow researchers to read an anonymous version of her EMR. Another type of third party would for instance be a certificate manager provided by another organisation. We do not recommend to give administrative roles to third parties in a private HD network. Therefore, proper consensus mechanisms need to be applied and administrative roles should only be given to trusted parties in the BC network.

- R_4: **Storage purpose and access logs** are important features for the ledger, because the ledger is used as reference to the behavior of users. Moreover, European GDPR policies oblige users to indicate for what purpose an event has been executed. Another aspect of storage purpose is what HD should be stored and why. Some ongoing efforts argue that sufficient purpose of why certain HD needs to be stored is required. Preferably, all HD should be accessible via the BC network. However, a reduced size of the BC will improve the performance of the BBHS. Ethics behind the storage purpose is beyond the scope of our project and should definitely be discussed with corresponding stakeholders.

- R_5: **Operation policies** are important requirements for the BC network, because operations such as READ or WRITE affect private HD. Under GDPR Article 5.1 [31], no personal data can be accessed for an indefinite period of time. Therefore, auditability and verification mechanisms of all the operations should be explicitly mentioned. These operations need to be logged for activity tracking and can be used as accountability for data breaches. Furthermore, access mechanisms of operations should be included in policies defined by parties in the BBHS. Therefore, all parties agree that proper security mechanisms are provided for operations which are performed by the BBHS.

- R_6: **Stored data integrity** is an important storage requirement, because HD should remain honest and complete. Distributed storage techniques allow for more data integrity, because multiple copies of pieces of data are distributed over data providers in the network. Centralized storage techniques require back-up techniques to recover unavailable HD. However, data recovery for large volumes of data takes a lot of time. Therefore, we recommend online copies of the HD, which distributed storage techniques already provide. Furthermore, the metadata tag and hash value of off-chain data indicate if the corresponding HD is tampered with. If data is tampered with, the BC network cannot recover the original copy of the HD, because the ledger does not directly store HD on-chain. Therefore, data integrity is very important for BBHSs and techniques should be implemented to deal with integrity issues that may occur.

3.2 Analysis of existing storage techniques

We have performed an investigation on the presence of the requirements from Section 3.1 in ongoing efforts. An overview of the results are present in table 1.

First, we consider BBDS [18]. BBDS has an elaborate evaluation of existing security techniques for storing and accessing data. They propose a series of cryptographic keys for the different assignments or events of their system. Their authentication phase includes a system setup, key exchange, and an authentication and key agreement. Furthermore, the purpose of their proposed ledger is to have good references for future transactions. Additionally, BBDS address the access control challenges associated with sensitive data. However, their symmetric key mechanism for secure accessibility is insufficient in comparison to other techniques that can be applied. Moreover, BBDS is still very conceptual. They do not describe the role of third parties in their system and they do not go into depth about data integrity and data format.

Second, we discuss the proposal of HyperBSA [8]. HyperBSA uses two different formats of data: continuous data and state data. They also propose a separate location for the storage of the types of data. Furthermore, they have performed a proper investigation on operation policies by using different cache strategies. Additionally, HyperBSA indicates that the ledger will be used as a log for reference by their proposed FileLog manager. The in-depth analysis on storage is included in their proposal. However, the accessibility and involved parties are not apparent.

Third, we evaluate the storage techniques implemented by Storj [20]. Storj performs an in-depth investigation on distributed storage techniques. They describe use cases where storage security and access mechanisms are properly explained. Furthermore, their system uses distributed storage techniques which provide better data integrity. Additionally, Storj does not restrict any data format. Nonetheless, Storj primarily focusses on a distributed storage system. Their system could possibly be integrated in a BBHS. However, they do not describe the third parties, storage purpose and access logs in their proposal.

Fourth, we consider Healthchain [28]. Healthchain proposes IPFS-based storage nodes which focusses on a secure and distributed storage mechanism. IPFS allows any data format to be stored in the system and can be maintained by the consortium healthcare providers. Furthermore, Healthchain describes use cases by giving pseudo code for access mechanisms. However, operation policies are not present in these use cases. Moreover, Healthchain has not performed an investigation on the role of third parties and the storage purpose and access logs.

Fifth, we discuss the proposal of Medblock [29]. The authors propose an access control protocol for which psuedo code is given. Furthermore, they propose a distributed storage system where each hospital is a data provider and no general data format is required. Their storage purpose for users is to have fast accessibility to EMRs. However, the authors do not explicitly explain the access control policy to allow third-party researchers to access these EMRs. Also, operation policies are not present in their proposal for a BBHS. Due to the GDPR regulations, Medblock has decided to not have a third party claim ownership of the HD. Nevertheless, a third party manages certificates of the consortium BC system and has supervision over the ledger.

At last, the proposal of MedRec [11] is considered. MedRec gives a proper explanation on their access mechanisms. They propose a DB Gatekeeper that implements an access interface to the patient node's local DB, governed by permissions stored on the BC. They also assume that 'provider' nodes already store data on networked servers with a high degree of security. However, MedRec does not seek to address the security concerns at the level of provider DBs. Nevertheless, operation policies, storage purposes and third parties are not present in the MedRec proposal. MedRec does describe that a distributed storage technique will be implemented. However, MedRec is very conceptual and we cannot give proper confirmation that the requirements are present from the many assumptions they make.

Table 1: Presence of requirements in existing work

proposal	R_1	R_2	R_3	R_4	R_5	R_6
HyperBSA [8]	✓	-	-	-	✓	-
MedRec [11]	-	-	-	-	-	-
BBDS [18]	-	✓	-	✓	-	-
Storj [20]	✓	✓	-	-	✓	✓
Healthchain [28]	✓	✓	-	-	-	✓
MedBlock [29]	✓	✓	✓	✓	-	✓

Table 1 displays what requirements are present in ongoing efforts. A requirement is marked as present once the requirement has been explained with sufficient details or an implementation is given.

Many existing work has a primary focus on either storage or access mechanisms for their proposed system. Nevertheless, many suggestions and concepts are given with little detail. The small amount of evaluation on the performance of different storage techniques make it hard to conclude what technique should be implemented in a BBHS. Therefore, we will propose our own architecture to ensure that at least all of the aforementioned requirements are present in the system. Additionally, the arguments for certain design choices by existing work will be be considered as well.

3.3 Proposed blockchain-based architecture

For the proposed BC-based architecture, we include all of the requirements from Section 3.1 and explain how they are implemented in the BBHS.

We have decided that one single cloud DB is not beneficial for a BBHS. Hackers will be a large threat to the system, because all private HD is stored in one location. Moreover, the DB will be provided by a third party which we do not trust. Therefore, each medical facility (MF) will provide its own cloud DB. The database manager nodes (DBMNs) connect with each other and the blockchain to construct a network for the BBHS.

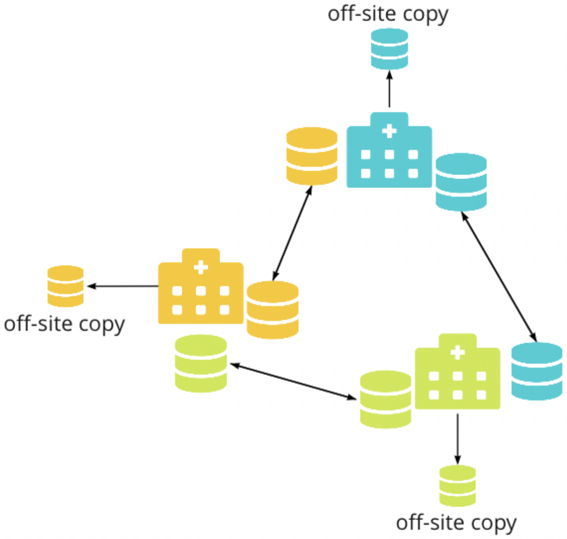

Figure 1: Database architecture for three medical facilities.

Another decision we made is to not use distributed storage techniques. The levels of complexity cause the BBHS to become much more inefficient. Consequently, more complex security techniques can be applied on the access mechanisms (R_2) of the BBHS. Also, existing techniques for DB security [32][33] are sufficient and already exist (R_2).

Data integrity (R_6) can be achieved by storing an additional copy of the DB at another medical facility. This indicates that two live copies of HD are present in the network. These live copies are immediately accessible. In case MF A goes offline, MF B, who stores a copy of the HD of MF A, can take charge of MF A's traffic. Additionally, an off-site copy of the DB is stored for recovery. If MF A shuts down, the off-site DB copy of MF A can be used for recovery. Because the recovery procedure takes much time for large volumes of HD, the second live copy is stored in the network.

An example scheme is given in Figure 1. The figure depicts the database scheme for three MFs. Each of these facilities have their own cloud DB of which a copy is stored at a neighboring MF. Additionally, each MF has its own off-site copy for recovery. There are no pairs in the system, such that each MF is dependent on another MF. Two MFs cannot decide to drop, because they affect two other MFs, and so on. By doing this, all MFs are connected through a large network. Of course, with multiple MFs, the network grows and the same DB structure is maintained.

The users of the BBHS use an interface to connect with the network and perform operations. The interface links the user to the corresponding MFs where their HD is stored. The MF will keep track of its users by storing their credentials and certificates. Now, patients can ask to view their EMRs. Because the interface is direcly linked to the correct DB, there is a faster query response. Furthermore, the traffic is limited to a smaller part of the network, such that the complete network is not directly accessible by all users. For doctors and researchers this is quite different. When a researcher requests to read specific HD from EMRs, a request is sent to the network. The database manager nodes check the EMRs of their patient and their corresponding consent form in order to see what data can be returned. The processing will be done locally, such that the network can keep performing. After this process, the researcher can read the acquired data through the interface.

Each MF has its own data format (R_1) for their cloud DB. It does not matter if data formats differ, as long as all operations can be performed. The operation policy (R_5) will be such that a general query request can be broadcast to the network, to which the DB manager nodes can respond with the correct correpsonding HD. Operation policies regarding access management are performed by the MFs themselves. An operation can only be performed, once the request is validated by the network. If a request is made, a transaction (SC) will be added to the ledger. Additionally, the return state of the request is added to the ledger. When the return state grants permission, the request can be executed.

As described in our proposal, no third party (R_3) is required to provide authority, DBs or any other system of the BBHS. The MFs are only dependent on each other. In the scope of this research, a third party would be a doctor or researcher from another medical facility that requests a certain operation. When the request is validated and added to the ledger, the network gets informed and the operation can continue. Each MF has its own DB manager node to process the request locally, such that the third party only receives the HD it is supposed to receive.

The storage purpose (R_4) of this proposal is to store the timestamp and stakeholders related to a transaction. This is required due to the GDPR and provides accountability for operations. All HD can be stored as it already is stored, because each MF has its own DB. Changes to the DB are immediately reported to the ledger. Once the change is validated, the DB can be updated. The MF which stored second copy of the DB gets notified via the ledger such that the the copy is up to date.

The complete overview of the proposed architecture can be found in Figure 2. Figure 2 displays the complete storage architecture of the BBHS. Each MF provides a DB manager node. Each node is connected to the consortium BC network. The HD can be obtained via a user interface (UI). Via the UI, users can perform operations on the HD. The UI is linked to its corresponding DB manager node. The DB system of a MF consists of an off-site storage which can be used for backup, and an online cloud storage. The online cloud DB stores credentials, certificates, consent forms and HD.

Additional benefits of our proposed system are that centralized storage gives an overall better performance than distributed storage. This allows us to have more complex security techniques implemented in the BBHS. The double database system allows for more data integrity and supervision over the participants in the network. Each private network has its own traffic. This means that a single system will not be overloaded with requests. Additionally, the query response time is a lot faster, because most of the search requests also only happen in the Private network. At last, the proposed architecture is much more scalable compared to ongoing efforts due to the centralized storage techniques and access mechanisms.

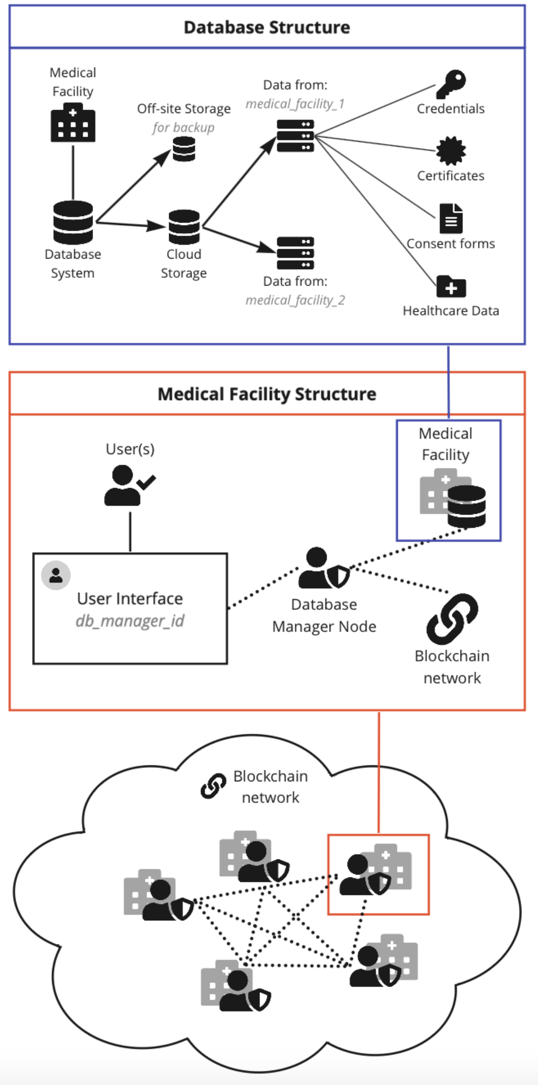

Figure 2: Two nodes with min. one connection

4 Discussions and future work

This section describes how existing work needs to be improved. Additionally, some discussion is given regarding data integrity and security. Furthermore, suggestions for future systems for a BBHS are given.

First, existing literature for proposed storage systems requires more evaluation of their implementation. The existing work should definitely test their system on real-life models, especially HyperBSA [8], for more realistic test results. The outcomes of the results can indicate that their performance improvements can be implemented in future BBHSs.

Second, many systems indicate that distributed storage techniques will be implemented in their BBHS. They need to look at its impact on the performance of the BBHS. If security techniques for storing and accessing the HD are insufficient or if the techniques cannot be found, distributed storage techniques should be reconsidered. Distributed storage techniques provide much higher integrity than centralized storage. Of course, there is always the possibility that a large part of the network shuts down and HD gets lost. More integrity in our proposed architecture can be achieved by storing additional live copies of the HD. However, losing efficiency by upscaling the BBHS might not be worth it. Therefore, experts need to perform a good evaluation regarding the data integrity of our proposal.

Third, we have seen many existing techniques to securely store HD. Therefore, we need to focus on security techniques for access mechanisms in order to retrieve this HD. The DB manager node should be set-up in such a way that it checks the role of the user who creates a request. It should not be possible for a user to convince the DB manager node that it has access to certain data. The exact techniques are out of the scope of this project, but this is a crucial part of the BBHS that needs to be taken into account. Existing work should also focus more on their access mechanisms and provide use cases where the security techniques on data storage and accessibility are properly explained.

Fourth, it is preferred to store all available HD in the BBHS. At first glance, it may seem beneficial to have all data available. However, a reduction of the stored HD could improve the performance of the BBHS. Experts should perform an in-depth investigation on what HD is required for the system and what HD could possibly be left out. Additionally, the MFs should have a look into a general structure for an EMR. HD is scattered and doctors find it hard to locate and process the data for proper diagnoses. If possible, the general structure for the EMR can be set-up in such a way that tags can be included for faster search queries. Also, all data can be found in the EMR, which means that all HD regarding a single patient is stored in a single document. For future storage of HD in BBHSs, only the EMRs will have to be stored if it includes all necessary HD. These tags make the storage system of the BBHS more structured and efficient.

Fifth, when creating new accounts for users in the system and certificates are assigned, an additional parameter should be implemented to indicate that the BC system has validated this user. This prevents each system from having to check the ledger every time a user logs into the system. When updates are made regarding the user, this parameter should of course be changed.

At last, because of the privacy of HD, the process towards an operational BBHS takes a long time. In order to speed up the process, some minimal requirements for security and privacy should be given in order to provide researches with the means to work towards a complete system. When minimal requirements hold, the complete system can properly be evaluated with performance tests and other experiments. The systems can afterwards be improved once we have more experience on the complete BBHS and its functionalites. We cannot implement this directly for actual MFs. However, we believe that we should start testing complete set-ups for faster improvements of the system. For now, existing work can be combined to get closer to such an operational BBHS, because their security techniques provide sufficient S&P for data storage and access management.

4.1 Future Work

Besides mentioning what is not yet done or incomplete in certain literature, some aspects need future research. These aspects have no definite solutions yet, but are interesting for BBHSs and should therefore be investigated.

- A **General method of proof** should be discussed in order to ensure a good policy of validation. Ethereum based systems use Proof-of-work for consensus, which means that mining is required for the system. This is computationally very inefficient for the scale of the system. Therefore, other types such as PBFT are introduced for BBHSs. Even Proof-of-authority has been proposed. However, this proof gives the data providers a lot of power in the system which should go to the users who are owners of their personal HD. Fortunately, consensus mechanisms are implemented to fasten up this validation process, yet a method of proof is still required. We suggest to find a minimal method of proof that is sufficient for validating transactions and other operations in the BBHS. Proof-of-stakeholders would be a good name for it, because the stakeholders are involved in the transaction. For proper consensus, the same amount of neutral users should also validate the transaction. The structure is very similar to a SC and can be used to automate the process. Less parties are involved, but the neutrality of other medical facilities can ensure that involved parties do not conspise with each other. However, a different group of validators is required per different transaction.
- The **impact of storage techniques** is highly important in future work. Assumptions are being made that certain storage techniques perform better than others or they are too complex for the BBHS. Each technique has its own security aspects which indicate how secure the storage technique is. A conclusion needs to be drawn in order to understand what storage sytem performs best and what their impact is in a BBHS. To do this, experts need to execute performance experiments in realistic test set-ups. However, this can only be achieved once an operational BBHS has been implemented.
- **Live data** comes from monitors, for instance, that continuously check the heart rate of a patient. The machines are live which means that each millisecond the HD gets updated. In the proposed BBHS, we preferably update the ledger every time HD gets updated. Therefore, we suggest to look for techniques where the ledger does not get overloaded with transactions about monitor data. A suggestion would be to update the file off-chain. Once the data is requested, the final version of the data can be shared and then a transaction can be added to the ledger.

Perhaps, the usage of this specific type of HD is not required by any party from other MFs. This could mean that the monitor updates the HD off-chain. Once a patient is no longer monitored, the HD can be added to the EMR of the patient and the ledger can get notified. This will reduce the number of transactions. We believe that multiple cases exist where these design choices can be made.

5 Responsible Research

For this paper, we have made an effort to conduct responsible research. During research, existing literature of ongoing efforts in BC and BBHSs have been evaluated. Our work provides small summaries of the relevant content of this existing literature. These summaries are present in this paper and used to propose an architecture of our own. For fair evaluation of the existing literature, the key requirements of Section 3.1 were selected in advance.

Our proposed architecture is conceptual due to time constraints. An idea is suggested which has been made visible in Figure 2. Therefore, a textual explanation is given in order for future researchers to reproduce this architecture.

6 Conclusions

BC technology experienced major developments over the last couple of years. Its transparency, immutability, confidentiality and auditability provide good solutions towards a secure storage of EMRs. However, there is still plenty of work to be done. In this paper, we have evaluated and investigated existing proposals for a BBHS. Furthermore we have selected six key requirements which are important for the storage system of such a BBHS. None of the investigated systems meet all six requirements identified in this study. Therefore, we have proposed a storage architecture of our own.

Off-chain storage is compulsory due to GDPR regulations. Multiple suggestions have been done on how to store the EMRs off-chain. Most ongoing efforts prefer distributed storage for more integrity of the data. Others indicate that a cloud DB provides better efficiency of the overall system. During the evaluation, we discovered that both solutions provide their own benefits. Nevertheless, with the existing security techniques that securely encrypt the data, one can provide secure storage and accessibility of EMRs in cloud DBs. In our proposal, each MF provides its own cloud DB and they create a network. Data integrity is improved by storing an additional copy of the database at another MF. There are still some discussions that need to be taken into account when considering a properly working BBHS, because in this paper we only discussed the storage part of a BBHS. There is more work to be done. For instance the choice on what data needs to be published and the impact of the storage technique still need to be evaluated in a working test-model for a BBHS.

We believe that, given the suggestions in Section 4, that BC will be a good technology for healthcare systems. In our opinion, it would be best to start working towards a complete BBHS and improve the efficiency of the BBHS once all components are functional. Only then proper tests can be held and the working BBHS can be improved.

References

[1] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar, "Multimedia big data analytics," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–34, Apr. 2018. [Online]. Available: https://doi.org/10.1145/3150226

[2] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, "Clinical big data and deep learning: Applications, challenges, and future outlooks," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 288–305, Dec. 2019. [Online]. Available: https://doi.org/10.26599/bdma.2019.9020007

[3] X. Ma, C. Wang, and L. Wang, "The data sharing scheme based on blockchain," *Symposium on Blockchain and Secure Critical Infrastructure*, 2020.

[4] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "A survey on security and privacy issues in modern healthcare systems: Attacks and defenses," 2020.

[5] "Privacy provision in collaborative ehealth with attribute-based encryption: Survey, challenges and future directions," *IEEE Access*, vol. 7, pp. 89 614–89 636, 2019.

[6] E. J. D. Aguiar, B. S. Faiçal, B. Krishnamachari, and J. Ueyama, "A survey of blockchain-based strategies for healthcare," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–27, Jul. 2020. [Online]. Available: https://doi.org/10.1145/3376915

[7] T. Hepp, M. Sharinghousen, P. Ehret, A. Schoenhals, and B. Gipp, "On-chain vs. off-chain storage for supply- and blockchain integration," *it - Information Technology*, vol. 60, 11 2018.

[8] X. Chen, K. Zhang, X. Liang, W. Qiu, Z. Zhang, and D. Tu, "Hyperbsa: A high-performance consortium blockchain storage architecture for massive data," *IEEE Access*, vol. 8, pp. 178 402–178 413, 2020.

[9] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3190508.3190538

[10] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, "Medchain: A design of blockchain-based system for medical records access and permissions management," *IEEE Access*, vol. 7, pp. 164 595–164 613, 2019.

[11] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016, pp. 25–30.

[12] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "Bpds: A blockchain based privacy-preserving data sharing for electronic medical records," 2018.

[13] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Information*, p. 9, 2019. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[14] M. Harikrishnan and K. Lakshmy, "Secure digital service payments using zero knowledge proof in distributed network," in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, 2019, pp. 307–312.

[15] H. Wang and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 42, no. 8, Jul. 2018. [Online]. Available: https://doi.org/10.1007/s10916-018-0994-6

[16] M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings," in *Security and Privacy in Communication Networks*, S. Jajodia and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 89–106.

[17] J. L. Fernández-Alemán, I. C. Señor, P. A. O. Lozoya, and A. Toval, "Security and privacy in electronic health records: a systematic literature review," *Journal of biomedical informatics*, vol. 46, no. 3, p. 541—562, June 2013. [Online]. Available: https://doi.org/10.1016/j.jbi.2012.12.003

[18] Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, Apr. 2017. [Online]. Available: https://doi.org/10.3390/info8020044

[19] D. Schwartz, N. Youngs, A. Britto *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, no. 8, p. 151, 2014.

[20] S. Wilkinson, "Storj a peer-to-peer cloud storage network," in *Storj A Peer-to-Peer Cloud Storage Network*, 2014.

[21] K. Wehrle, S. Götz, and S. Rieche, "7. distributed hash tables," in *Peer-to-Peer Systems and Applications*. Springer Berlin Heidelberg, 2005, pp. 79–93. [Online]. Available: https://doi.org/10.1007/11530657_7

[22] H. Do and W. Ng, "Blockchain-based system for secure data storage with private keyword search," *2017 IEEE World Congress on Services (SERVICES)*, pp. 90–93, 2017.

[23] X. L. Yu, X. Xu, and B. Liu, "Ethdrive: A peer-to-peer data storage with provenance," in *CAiSE-Forum-DC*, 2017.

[24] T. McConaghy, R. Marques, A. Muller, D. de Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: A scalable blockchain database," in *BigchainDB: A Scalable Blockchain Database*. GmbH, Berlin, Germany, 2016.

[25] R. Kumar, N. Marchang, and R. Tripathi, "Distributed off-chain storage of patient diagnostic reports in healthcare system using ipfs and blockchain," in *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2020, pp. 1–5.

[26] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 839–858.

[27] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," 2015.

[28] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770–8781, Oct. 2019. [Online]. Available: https://doi.org/10.1109/jiot.2019.2923525

[29] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and secure medical data sharing via blockchain," *Journal of Medical Systems*, vol. 42, no. 8, Jun. 2018. [Online]. Available: https://doi.org/10.1007/s10916-018-0993-7

[30] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, May 2018. [Online]. Available: https://doi.org/10.1016/j.scs.2018.02.014

[31] "General data protection regulation (gdpr)," May 2018. [Online]. Available: https://gdpr-info.eu/

[32] H. Kayarkar, "Classification of various security techniques in databases and their comparative analysis," 2012.

[33] K. Jakimoski, "Security techniques for data protection in cloud computing," *International Journal of Grid and Distributed Computing*, vol. 9, no. 1, pp. 49–56, Jan. 2016. [Online]. Available: https://doi.org/10.14257/ijgdc.2016.9.1.05