Contents lists available at ScienceDirect

Blockchain: Research and Applications

journal homepage: www.journals.elsevier.com/blockchain-research-and-applications

Research Article

Hyperledger fabric platform for healthcare trust relations—Proof-of-Concept

Aleksandar Nedaković [a], Anton Hasselgren [b,*], Katina Kralevska [a], Danilo Gligoroski [a]

[a] Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway
[b] Department of Neuromedicine and Movement Science, Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway

ARTICLE INFO

Keywords:
Blockchain
Hyperledger fabric
Health systems
Distributed ledger

ABSTRACT

In recent years, blockchain technologies have expanded from the finance field to other areas that rely on trust-based solutions. The healthcare industry represents one such area, as digital transformation disrupts relationships between patients, healthcare professionals, and healthcare institutes. Patients and healthcare institutes lack a proficient tool to verify the credentials of medical professionals in a digital environment. Furthermore, healthcare professionals lack a tool where they are in control over their credentials. The first contribution of this paper is a proposal of a solution that leverages the private permissioned Hyperledger Fabric blockchain and smart contracts to provide a source of transparent trust for relationships within the healthcare industry. Second, we pave the ground for GDPR compliance by storing only the hash values on the blockchain. Third, we solve the problem of patient authentication by utilizing cryptographic techniques. Finally, we prove the usability of the proposed solution by implementing a user interface and creating a live deployment.

1. Introduction

In order to be integrated with the increasingly virtualized world [1], the healthcare sector is becoming rapidly digitalized. Consequently, the increased digitalization of the healthcare industry is introducing a new scope of challenges. One challenge area is the trust relationships between patients, physicians, and healthcare institutes. These trust relations are becoming crucial amidst the increase in remote work and virtual consultations. We emphasize the importance of such trust relations, as they represent the foundation of well-perceived healthcare services and patient contentment [2]. There is also a need to evaluate the outcomes of given care. That should be provided by a platform that enables patients to submit patient-reported outcomes after treatment. The healthcare authorities could use data in this platform to verify the outcome of a given healthcare service. Additionally, the platform could also be used by other stakeholders in the healthcare system.

Amid globalization, it is becoming more frequent for physicians to move between countries and jurisdictions [3]. Finding themselves in new regulatory environments, they are often first required to prove their expertise and the validity of their certifications.

All these reasons call for developing a service that would provide easy verification of certification and experiences for healthcare professionals operating across borders and jurisdictions. This paper describes a blockchain-based solution for the abovementioned problems regarding trust relations in healthcare systems. We name the solution VerifyMed 2.0, as it builds on the old proof-of-concept VerifyMed solution [4], which we refer to as VerifyMed 1.0.

Our contribution: The main contribution of this work is the proposal of a platform that addresses the challenges in healthcare trust relations in a virtualized healthcare environment. The VerifyMed 2.0 platform is built on Hyperledger Fabric and enables patients to verify the credibility of physicians' certificates, experience, and competence. Healthcare institutes govern the VerifyMed network by voting on network configuration and membership matters. The solution architecture is scalable and allows the addition of new network users and their corresponding Hyperledger Fabric peers. Furthermore, the solution does not impose transaction fees compared to other smart contract solutions. The corresponding software system was created from the ground up. The application's codebase is located on GitHub in a private repository.

The paper is organized as follows: Section 1 introduces the topic and presents the related work. In Section 2, the methodology of our work is described, and we define the system's requirements. Then, Section 3 presents the results by describing the design, architecture, implementation details, and outcomes of a test set implementation. We

* Corresponding author.
E-mail address: anton.hasselgren@ntnu.no (A. Hasselgren).

https://doi.org/10.1016/j.bcra.2023.100156
Received 14 January 2023; Received in revised form 27 June 2023; Accepted 11 July 2023
Available online 1 August 2023

discuss the results and potential future work of the paper in Section 4. Finally, Section 5 concludes the paper.

1.1. Related work

The proposed applications of blockchain in the healthcare sector are often related to data management problems. Blockchain can be used to provide data integrity, access control, data versioning, and non-repudiation. Recent review papers on applications of blockchain in healthcare [10–13] point out that the majority of the current research revolves around patient data management (focus on the management of Electronic Medical Records (EMR) [7,14,15] and Electronic Health Records (EHR) [5,6,16]), remote patient monitoring [17,18], pharmaceutical supply chains [19,20], health insurance [21], and vaccination certifications [8,9].

VerifyMed 1.0: As we can see in Table 1, there is a clear research gap for applications of blockchain for trust relations between physicians, healthcare institutes, and patients. This gap, addressed by VerifyMed 1.0 [4], represents the original proof-of-concept solution that provides a trusting relationship between a patient and a physician in a virtualized setting. It is implemented in Ethereum smart contracts and functions through interaction with the public Ethereum blockchain. Trust relationships are achieved by sharing data representing different types of evidence for trust.

1) **Evidence of authority** represents data that affirm a physician's ability to practice as a healthcare worker. These data comprise formal licenses and certifications that trusted healthcare institutes must approve. Furthermore, this evidence also includes information about physician employment within one of the trusted healthcare institutes.
2) **Evidence of experience** contains proof of experience that a physician has collected throughout years of dealing with various patient health issues. This evidence is composed of a list of treatments provided by the physician.
3) **Evidence of competence** represents a qualitative measure of treatments provided by the physician. It is provided through patient reviews of provided treatments. While evidence of experience only conveys the sore numbers of provided treatments, the evidence of competence represents patients' satisfaction with the outcomes of those treatments.

However, as VerifyMed 1.0 represents a proof-of-concept solution, it has limitations. The main limitations of VerifyMed 1.0 that are solved in VerifyMed 2.0 are as follows.

- **High transaction fees** which caused by the usage of the Ethereum public blockchain that imposes monetary costs for all submitted transactions. The price to submit a treatment and get it approved ranged from 1 USD to 405 USD, depending on the Ether and Ethereum gas price.
- **Scalability issues** stemmed from the finite number of transactions that can fit within a single block of the public Ethereum ledger. This results in the theoretical maximum of 1.7 treatment submissions per second if the whole public Ethereum network was only mining VerifyMed blocks, which is difficult to achieve.
- **Data privacy limitations and GDPR incompatibility** which caused by storing health domain-related data stored on the public Ethereum blockchain.
- **Missing patient identification** allows the creation of fake patients and fraudulent evaluations.
- **User interface** represents a bare-bone proof-of-concept interface that provides a user with tools to try and test all the potential functionalities of the VerifyMed 1.0 solution but is missing numerous features.

2. Methodology

The artifact in this research work was developed using the principles of design science [22] and requirement engineering. This methodological approach is described in this section, starting by describing the demands from the domain.

2.1. Actors

After a thorough analysis of the architecture of system actors of VerifyMed 1.0, we notice that the same entity will have the role of multiple actors in most real-world scenarios. For example, the license provider and treatment provider represent the same entity, except when the clinic uses a third-party provider of a virtualized healthcare environment. Furthermore, it is common for the same healthcare institute that provides healthcare facilities to be capable of issuing licenses for physicians. Therefore, in VerifyMed 2.0, we merge the roles of license issuer, license provider, and treatment provider into a single actor—the Healthcare Institute.

Another change of VerifyMed 2.0 is removing authority actors and assigning that role to the healthcare institutes. Therefore, compared to VerifyMed 1.0, the trust model is now flattened, as shown in Fig. 1. The VerifyMed 2.0 network is then built from two trust layers. The upper layer represents healthcare institutes that do not inherently trust each other, thus resorting to using Hyperledger Fabric consensus mechanisms.

The lower layer contains physicians and patients, which builds trust

Table 1
Comparison of VerifyMed to the selected research papers.

Publication	Application area	Platform	GDPR	Consensus mechanism	Anonymity	Unlinkability	Track of data ownership	User interface
Wang and Qin [5]	EHR	Hyperledger Fabric	×	Raft	×	×	✓	–
Stamatellis et al. [6]	EHR	Hyperledger Fabric	✓	Raft	✓	✓	×	CLI
Abdul-Moheeth et al. [7]	EMR	Hyperledger Indy	×	RBFT	✓	Experimental via Indy's Anoncreds 2.0	✓	iOS App
de Vasconcelos Barros et al. [8]	Vaccination proof	Hyperledger Indy	✓	RBFT	✓	Experimental via Indy's Anoncreds 2.0	✓	website
Abid et al. [9]	Vaccination proof	Private Ethereum	✓	proof-of-stake (ex proof-of-work)	✓	✓	✓	CLI & QR codes
VerifyMed1.0 [4]	Trust relations	Public Ethereum	×	proof-of-stake (ex proof-of-work)	✓	✓	×	website
VerifyMed2.0	Trust relations	Hyperledger Fabric	✓	Raft	✓	✓	✓	website

EHR: Electronic health record, CLI: Command line interface, EMR: Electronic medical record, RBFT: Redundant Byzantine Fault Tolerance.

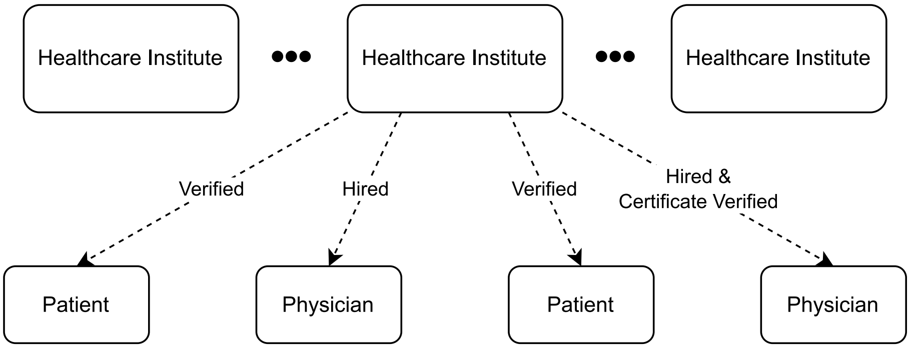


Fig. 1. Diagram illustrating a trust model between actors in VerifyMed 2.0 system.

relations with the upper layer through trust bonds saved in the Hyperledger Fabric blockchain ledger.

Therefore, in VerifyMed 2.0, users are classified into one of the following four roles.

- **Healthcare Institute** represents an abstraction of any healthcare organization that can facilitate interactions between physicians and patients or certify physicians to work in the healthcare domain. They represent the backbone of the VerifyMed 2.0 network, as they build the foundation of the trust relations framework. They can verify physicians' certificates, hire physicians, verify patients' identities, or allow the inclusion of new healthcare institutes into the network. Each healthcare institute represents an organization in the Hyperledger Fabric blockchain and has at least one Hyperledger Fabric peer assigned to it.
- **Physician** is a healthcare practitioner who provides treatments to patients. As in VerifyMed 1.0, the system stores evidence of authority, experience, and competence for physicians. Evidence of authority is represented through physicians' certificates verified by the healthcare institute that issued them. A list of treatments stored in the system represents evidence of experience, while patient reviews of these treatments serve as evidence of competence.
- **Patient** can use the system to access their treatments and leave reviews for their physicians.
- **Unauthenticated website visitor** represents any user who accesses the VerifyMed 2.0 website without authenticating and getting a role assigned. An unauthenticated visitor has access to the minimal set of functionalities accessible by any other authenticated actor of a system. Such features revolve around the ability to search and filter physicians based on their ratings.

2.2. Functional requirement

In order to design and implement the new and improved version of VerifyMed, the trusted provider in the healthcare domain, we first define a list of functional requirements that need to be fulfilled. For the sake of consistency, we use the Connextra template to specify the functional requirements as user stories [23]. This template is the most common way of expressing the functional requirements in the software development industry [24], as it clearly provides information about the goal of the functionality and the role that requires it. The template has the following wording: "As a ⟨ role ⟩, I want ⟨ goal ⟩, [so that ⟨ benefit ⟩]". Table 2 contains all functional requirements of VerifyMed 2.0 grouped by the roles that have access to them.

2.3. Non-functional requirement

While designing the VerifyMed 2.0 solution, in addition to functional requirements, we also need to follow some quality attributes by fulfilling the non-functional requirements. These requirements are somewhat similar to those of VerifyMed 1.0. We use the same classification in four categories.

1) **Privacy requirements: Untraceability of patient identity.** The identity of patients in the network should be treated as highly confidential. Adversaries should not be able to link to treatments and treatment reviews of patients, as it would reveal the private health condition of the patient.
   Anonymity of treatment instructions. The detailed instructions stored in the treatment should be readable only by the designated patient and the physician who created the treatment. These instructions keep specific details that could compromise the patient's privacy.
   Physician control of privacy. All physicians in the network should have complete control of the visibility of their data, as it is publicly accessible through the VerifyMed 2.0 website. They should have options to temporarily hide the data and permanently delete the account.
2) **Security requirements: access control.** All functionalities provided by the system should have multi-layered access control, thus preventing unauthenticated and unauthorized users from accessing the functionalities that they are not supposed to access.
   Fraudulent treatments. Physicians should not be able to add treatments for the healthcare institutes that have not hired them.
3) **Fraudulent patients:** Patients should not be able to create reviews for treatments in healthcare institutes that have not verified their identity. Otherwise, physicians would be able to create fake patient accounts and fake reviews for their treatments to boost their ratings.
   Fraudulent reviews. Every treatment can have only one review, and reviews cannot be created without going through the treatment with a physician first.
4) **Availability requirements: No downtime when the network is updated.** The parameters of the blockchain network should be configured dynamically if most healthcare institutes agree to do so. No downtime should occur when the configuration is updated. Furthermore, there should be no downtime when the majority of healthcare institutes decide to add or remove a healthcare institute.
   Recoverability when minority misbehaves. In case the minority of healthcare institutes misbehave or become permanently unavailable, the rest of the network should be able to remove them.
   Consensus nodes downtime. The network should stay functional as long as the majority of the nodes in the consensus mechanism are active (e.g., ordering service of the Hyper ledger Fabric network).
   Recoverability of the data when nodes are lost. As long as there is at least one active node in the network, the public information of the blockchain should be reconstructible.
5) **Scalability requirements: Minimal amount of data on the blockchain.** The blockchain ledger should not be used to store large quantities of data, as it is immutable, and its size only grows over time.

Table 2
All functional requirements of VerifyMed 2.0 grouped by actors.

Unauthenticated Website Visitor Functionalities	Authenticated User Functionalities	Healthcare Institute Functionalities	Physician Functionalities	Patient Functionalities
Any authenticated user can also access functionalities accessible to an unauthenticated website visitor. • As a website visitor, I want to access the landing page, so that I can use the website. • As a website visitor, I want to log in using my credentials, so that I can access the rest of the website. • As a website visitor, I want to register as a healthcare institute, so that I can access healthcare institute functionalities. • As a website visitor, I want to register as a patient, so that I can access patient functionalities. • As a website visitor, I want to register as a physician, so that I can access patient functionalities. • As a website visitor, I want to search through all physicians in the network, so that I can find the one I want to contact. • As a website visitor, I want to view a profile of a physician, so that I can see their data and contact them.	Authenticated user functionalities represent functionalities accessible by physicians, patients, and healthcare institutes of the network. • As a logged-in user, I want to change my login password, so that I can log in again with a new password. • As a logged-in user, I want to log out from the website, so that I can log in again with a different account. • As a logged-in user, I want to search through all treatments provided by physicians in the network, so that I can estimate the type and style of treatments that the specific physician offers. • As a logged-in user, I want to search through all patients in the network, so that I can find the one whose identity I want to verify.	Healthcare institutes of the VerifyMed 2.0 network have access to the following functionalities. • As a healthcare institute, I want to create a hiring request for a physician, so that they can join my organization. • As a healthcare institute, I want to accept/reject physicians' certificate verification requests, so that their credibility can be trusted. • As a healthcare institute, I want to verify patients' identity, so that they can review treatments created by physicians hired by me. • As a healthcare institute, I want to search through all patients in the network, so that I can find the one whose identity I want to verify. • As a healthcare institute, I want to vote on the proposal, so that the network members or parameters can be updated or remain unchanged in case there are not enough votes. • As a healthcare institute, I want to create a proposal about adding or removing a healthcare institute in the network, so that all healthcare institutes can vote on it. • As a healthcare institute, I want to create a proposal about changing the channel parameters of the network, so that all healthcare institutes can vote on it.	Physicians of the VerifyMed 2.0 network have access to the following functionalities. • As a physician, I want to create a new treatment for a patient, so that they can follow the detailed instructions and review it. • As a physician, I want to filter only the treatments that I have provided and to see the detailed instructions that I have provided so that I can check for patients' reviews and get feedback. • As a physician, I want to upload my certificate and request its verification from a chosen healthcare institute, so that the healthcare institute can verify it. • As a physician, I want to accept/reject the health care institute's hiring request, so that I can be either hired or not by the healthcare institute. • As a physician, I want to temporarily hide all my data (profile information, treatments, employment information, and treatment reviews) from the public website, so that I can protect my privacy. • As a physician, I want to permanently delete my data (profile information, treatments, employment information,and treatment reviews) from the network, so that I can protect my privacy. • As a physician, I want to decide which of my data should be hidden or publicly shown.	Patients in the VerifyMed 2.0 system can use the following functionalities. • As a patient, I want to filter only the treatments created for me and to see the detailed instructions, so that I can create a review for them. • As a patient, I want to review a treatment created for me, so that I can give feedback to the physician and rate them.

Vertical and horizontal scalability. Administrative and geographical borders do not bind the VerifyMed 2.0 solution, thus making it a potentially cross-country service. It is of great importance that the solution is scalable to facilitate the potential growth of the network. The solution should be both vertically and horizontally scalable. Vertical scaling refers to the increase in power of already deployed nodes, while horizontal scaling indicates the addition of new nodes.

3. VerifyMed 2.0—Hyperledger fabric-based platform

In this section, we describe the design and architecture of the VerifyMed 2.0 solution. The proposed VerifyMed 2.0 platform serves as a facilitator of trust relations between healthcare institutes, physicians, and patients. This is achieved by storing the evidence of authority, experience, and competence for all the physicians on the network. Healthcare institutes represent the authorities in the network and are responsible for creating trusting relations with physicians and patients. Furthermore, healthcare institutes can govern the network by proposing and voting on potential changes to the network configuration.

In the first subsection, we discuss the usage and choice of the underlying blockchain technology used in VerifyMed 2.0. Then, we explain the design decisions for each of the four main improvements over the old VerifyMed 1.0 solution. Finally, we elaborate on how the proposed architecture fulfills the non-functional requirements.

3.1. The usage of blockchain technology

The idea of VerifyMed 2.0 is to enable trust relations between multiple healthcare institutes and their physicians and patients. This problem is decentralized by nature, as no single global authority governs all healthcare institutes that could be used as a trusted third party. In contrast, every healthcare institute represents a sovereign organization that might not trust physicians and patients outside their organization. Therefore, our solution utilizes blockchain technology to facilitate trust relations between such independent organizations. Furthermore, the usage of blockchain provides us with the following features.

- **Integrity**—data submitted to the blockchain ledger are immutable and cannot be fraudulently modified by adversaries;
- **Transparency**—all activity is recorded on the blockchain, thus enabling any new organization to inspect the trustworthiness of the current state of the blockchain ledger;
- **Availability**—blockchain ledger is readily available to all parties that are part of the network, and there is no single point of failure;
- **Persistence**—the availability of the blockchain does not depend on a single organization, thus providing the potential for much superior longevity and persistence compared to centralized solutions.

However, the usage of blockchain technology has some significant downsides in terms of performance. As already pointed out, the

VerifyMed 1.0 has issues with throughput and high transaction fees. Therefore, it should be kept in mind that blockchain technology should be used only when a centralized database is unusable because of mistrust issues between database users.

Now, we describe possible blockchain solutions that can be used to implement VerifyMed 2.0. Then, we explain why we opted for the Hyperledger Fabric blockchain.

1) Possible Blockchain Solutions: Ethereum, being the most developed and documented blockchain platform [25], is also the most frequently used blockchain for healthcare-related applications, followed by Hyperledger Fabric [26]. Therefore, our study focuses on these two platforms. By referring to Würst and Gervais [27], we concluded that we should create a private permissioned blockchain network for the VerifyMed 2.0 solution. In contrast, the VerifyMed 1.0 solution was built on the public Ethereum blockchain, which resulted in poor scalability and high transaction fees of the solution. Another problem of building on top of public blockchains is the dependency that is formed between the platform and the underlying blockchain network. Solutions built on top of such blockchains depend directly on the stability of an underlying cryptocurrency of blockchain. In case of instability, miners leave the blockchain network, thus hindering the functionality of all projects built on that blockchain.

The simplest solution for VerifyMed 2.0 would be just to migrate the old VerifyMed 1.0 solution to a private Ethereum blockchain network. However, this approach has multiple downsides, as the Ethereum blockchain is inherently developed with the public network in mind. The Ethereum consensus mechanism significantly reduces transaction throughput, introduces transaction fees, and does not benefit from the fact that all blockchain network nodes are known in advance. Furthermore, in the native Ethereum blockchain, it is impossible to restrict access to the network only to verified nodes. Additionally, any member of the Ethereum network can freely create new smart contracts and customize the blockchain functionalities without the approval of other network members. Finally, there is an unavoidable presence of Ether cryptocurrency, which serves no purpose in the VerifyMed 2.0 solution and unnecessarily increases the complexity of the solution.

2) Hyperledger Fabric: We chose the Hyperledger Fabric blockchain as an underlying technology for the VerifyMed 2.0 solution. Hyperledger Fabric is a private permissioned blockchain by its definition; thus, it aligns with the needs of VerifyMed 2.0. The key feature of Fabric is its high customizability [28], which stems from its modular design and pluggable features that enable it to be tailored for a specific use case, thus increasing security. However, this highly adjustable design makes the systems complex and drastically increases the setup and development time [29]. Furthermore, it has many other features that are particularly beneficial for VerifyMed 2.0:
   - **Open-source** nature of the Hyperledger project enables free usage of the solution and guarantees the longevity of the project and engagement of the highly active community. Furthermore, the project is well-documented and frequently updated.
   - **IBM Cloud** [30] service enables smooth deployment and scalability control of the live VerifyMed 2.0 service. This is achieved by leveraging the IBM Blockchain Platform [31].
   - **PKI-based identity management** makes the integration of VerifyMed 2.0 into an already existing healthcare system easier, as most healthcare institutes already utilize PKIs. Furthermore, it provides all network members with public-private key pairs that can be used for encryption and signature schemes.
   - **Fabric's organizations** can be mapped on healthcare institutes of the VerifyMed 2.0 network. Each healthcare institute represents one trust domain and has its peers in the network.
   - **Private data collection enables** healthcare institutes to keep their data private. Furthermore, it enables the deletion of the data from the system, thus providing GDPR compatibility.
   - **High customizability of the channel configuration** enables healthcare institutes to tune the specific aspects of the blockchain network, so it aligns better with the requirements. This is achieved through the proposal and voting system.
   - **High throughput** of the Fabric network enables the VerifyMed 2.0 network to scale and grow over time.
   - **Security** of the system is facilitated by the permissioned nature of the blockchain as well as the custom-tailored defined policies that govern the accessibility of all actors in the system [32]. As the security analysis of Hyperledger Fabric matures [33], the security of VerifyMed 2.0 will mature.

Fig. 2 depicts the high-level architecture of the VerifyMed 2.0 private permissioned Fabric blockchain. Every healthcare institute is mapped to a single network organization and controls its peers that directly interact with the blockchain channel. Furthermore, there is one unique organization that does not correspond to any healthcare institute and is named VerifyMed. The VerifyMed development team controls this organization, and it represents an access point to the blockchain network for all physicians and patients. In this way, the physicians and the patients are not individually mapped to organizations, as we cannot expect them to pay for the deployment of their peers in the network. Therefore, the identities of physicians and patients are created by the VerifyMed organization.

All website users interact with the blockchain through the WebApp server deployed and maintained by the VerifyMed organization. This server also stores identities, including private keys, of all the network's patients, physicians, and healthcare institutes. WebApp uses private keys to impersonate the user's identity and to send transactions in the user's stead. Transactions are sent through the VerifyMed organization peer to the blockchain channel.

The channel's ordering service determines the order of the submitted transactions to the network. It is controlled by ordered peers that are provided by all network organizations.

3.2. Addressing VerifyMed 1.0 limitations

In order to better explain the design decisions of VerifyMed 2.0, we refer to the limitations of the VerifyMed 1.0 solution [4] and explain how the new solution solves them. We cover each of the four main limitations of the old solution and explain which parts of the VerifyMed 2.0 design address them.

1) Addressing high fees and scalability issues: VerifyMed 1.0 uses the public Ethereum blockchain, which is based on proof-of-work consensus and entails transaction fees. The cost of these fees is highly volatile, and it depends on the current gas cost in the public network and on the price of Ether. We observed that the treatment submission and approval price could fluctuate between 1 USD and 405 USD.

The problem of high transaction fees is inherently solved in VerifyMed 2.0 by leveraging the private permissioned Hyperledger Fabric blockchain instead of the public Ethereum blockchain. The permissioned nature of Hyperledger Fabric does not require proof-of-work-based consensus; thus, no mining fees are needed. Therefore, Fabric has zero transaction fees, and the deployment costs of the orderer peers are handled by their respective organizations.

Another limitation of the public Ethereum blockchain is the low transaction throughput, as new blocks are, on average, generated once every 13 s. This dramatically decreases the potential scalability and growth of the VerifyMed 1.0 solution.

The Hyperledger Fabric blockchain does not need to wait for a

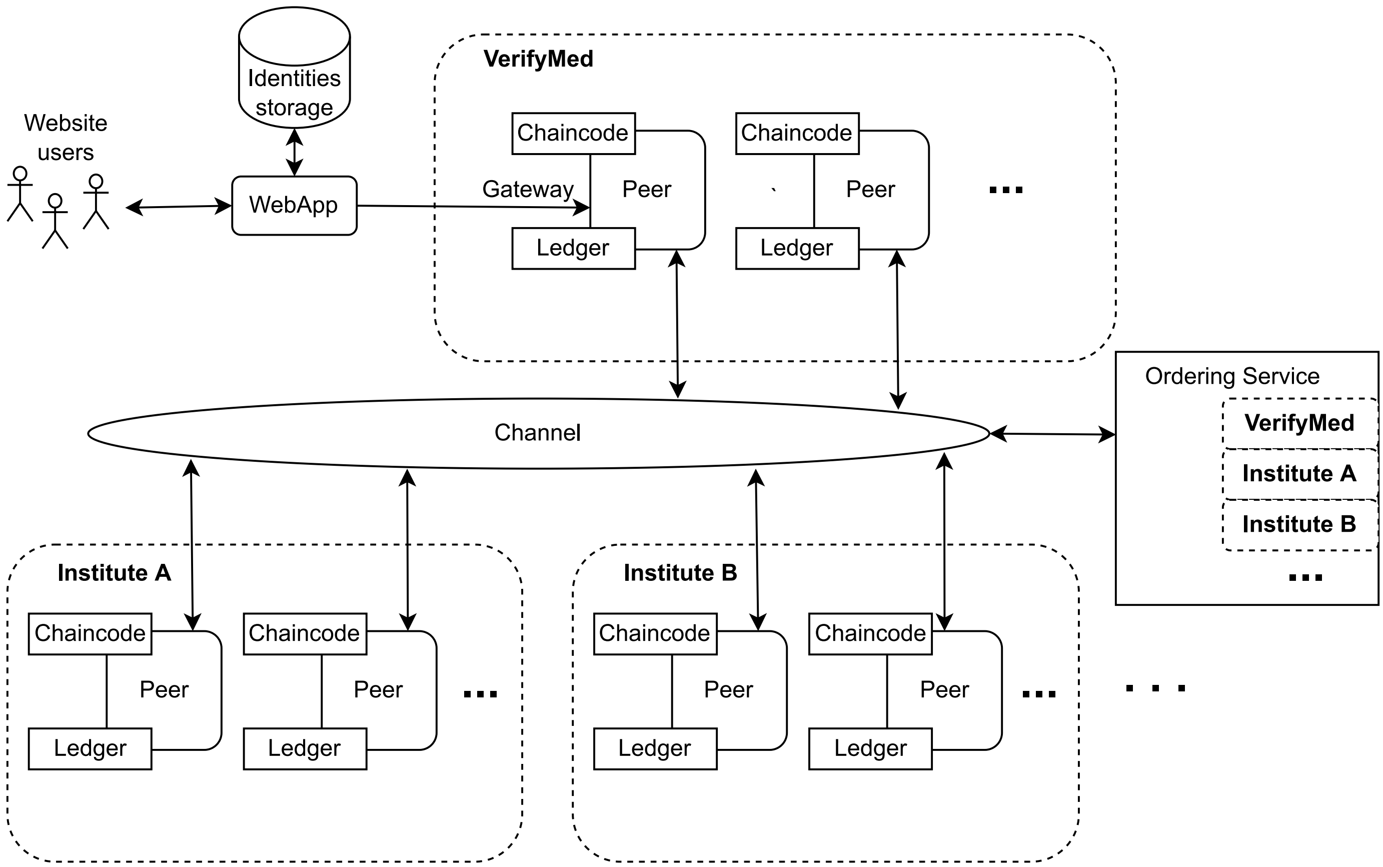


Fig. 2. Diagram showing the architecture of the VerifyMed 2.0 Fabric blockchain. Dotted lines represent organizations. It should be noted that every organization has a corresponding CA that assigns identities to all peers and users who are part of that organization.

nondeterministic mining process that takes a considerable number of seconds to generate a single block. In contrast, the blocks are constantly generated by the ordering service, and their size is determined within the channel configuration that can be updated if a majority of network organizations vote for it. Therefore, the number of transactions that can potentially be submitted in the network varies depending on the network configuration and size. The Fabric blockchain has been tested to be capable of processing more than 3500 transactions per second while scaling the network size to over 100 peers [34]. It should be noted that in a private Fabric network, all the transaction throughput is used for the VerifyMed system, while in public blockchains, one dApp can utilize only a small amount of throughput. Therefore, depending on the number of engaged peers in the network, VerifyMed 2.0 can potentially have a throughput of a few thousand treatment creations per second.

2) GDPR compatibility: The GDPR represents a regulatory framework defined by the European Union that regulates the storage, processing, and security of personal data [35]. GDPR represents an extensive regulatory framework that still has many uncertainties regarding its application to blockchain technologies [36]. Therefore, we will focus on the two most common conflict points between the GDPR and blockchain frameworks: the anonymization of personal data and the right to be forgotten.

Anonymization of Personal Data Art. 4(15) of GDPR[1] defines data concerning health as “personal data related to the physical or mental health of a natural person, including the provision of health care services, which reveal information about his or her health status.”

Therefore, all the treatment data in the VerifyMed 2.0 solution align with this definition and have to be anonymized before being shown to the public. This is achieved by encrypting sensitive data using symmetric encryption. The key is generated using Elliptic-Curve Diffie–Hellman (ECDH) between a physician’s and patient’s public-private key pairs.

Fig. 3 illustrates the treatment creation and review process. While constructing a treatment, the physician uses their private and patient’s public keys to construct a secret key that is then used to encrypt sensitive data. Later, when the patient wants to review the treatment, the patient searches through all available treatments and looks for the treatment that can be decrypted using the secret key generated as an ECDH combination of the patient’s private key and the physician’s public key. When such treatment is found, the patient can read the detailed instructions and generate a review.

Right to be Forgotten Art. 17 of GDPR[2] refers to the ability of data owners to delete all of the personal data that concern them. This right is generally hard to achieve on blockchain systems because of their inherent immutability. However, Hyperledger Fabric allows the usage of private data collections for data storage, while the blockchain ledger only keeps the hash values of the actual data.

VerifyMed 2.0 utilizes private collection for storage of all the data, as shown in Fig. 4, thus keeping no private data on the blockchain. Therefore, if any physician requests the deletion of all his/her data, this is achieved by purging that data from all peers’ private collections.

3) Identity solutions for patients: The VerifyMed 1.0 solution is missing a system for the verification of patient identity.

[1] https://gdpr-info.eu/art-4-gdpr/.

[2] https://gdpr-info.eu/art-17-gdpr/.

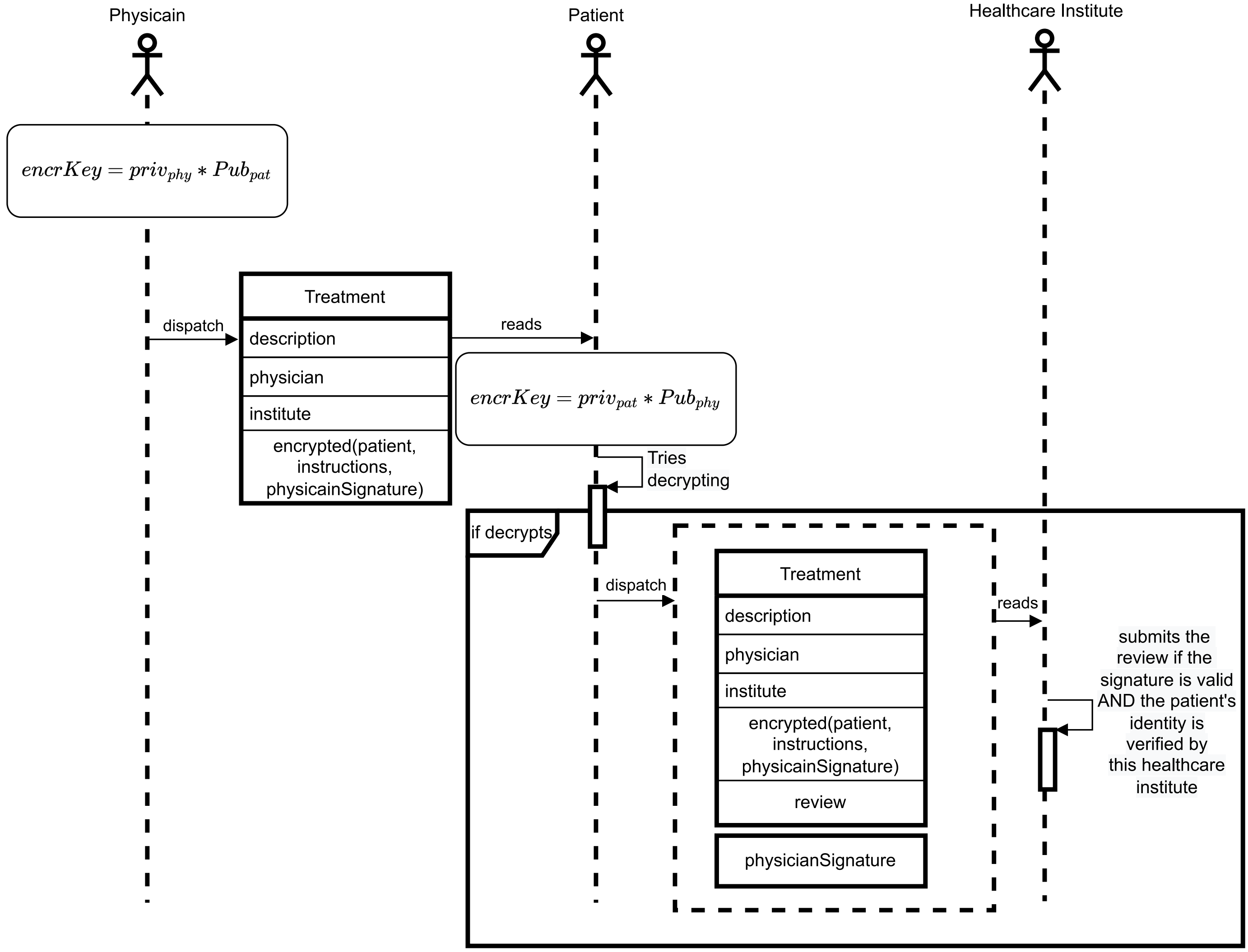

Physicain
Patient
Healthcare Institute
$encrKey = priv_{phy} * Pub_{pat}$
dispatch
Treatment
description
physician
institute
encrypted(patient, instructions, physicainSignature)
reads
$encrKey = priv_{pat} * Pub_{phy}$
Tries decrypting
if decrypts
dispatch
Treatment
description
physician
institute
encrypted(patient, instructions, physicainSignature)
review
physicianSignature
reads
submits the review if the signature is valid AND the patient's identity is verified by this healthcare institute

Fig. 3. Diagram showing how a physician creates a treatment, and then a patient finds it and provides a review. priv refers to a private key, while Pub denotes a public key.

This enables physicians to create fake patient profiles and then use them to review their treatments, thus manipulating their ratings. To combat this, in VerifyMed 2.0, when a patient submits a review, the healthcare institute checks if the patient is verified (as shown in Fig. 3) by that institute. The healthcare institute's list of verified patients is stored in its private collections (to maintain confidentiality). Hence, these transactions are performed only by peers that belong to the healthcare institute's organization.

Furthermore, the healthcare institute also checks if the patient has provided a valid physician's signature on the treatment. Only if the physician's signature is provided, the review will be accepted. The physician creates a signature of the treatment data during the treatment creation process. However, the signature is not publicly available but is encrypted with the patient's name and detailed treatment instructions. Therefore, only the patient for whom the treatment was designated can access this treatment and later use it to submit a review.

4) User interface improvements: The VerifyMed 1.0 solution provides a minimalistic user interface with many missing features. In contrast, VerifyMed 2.0 provides a proper website experience, with the landing page, registration, log-in flow, and access control. During the design process, two questions had to be addressed regarding the user interface.

Mobile Application or Website? We had to choose an adequate medium that serves the user interface of the VerifyMed 2.0 solution. Mobile applications provide better usability and accessibility, which is essential for healthcare platforms, as they are supposed to be accessible by users of all age groups and characteristics. However, mobile applications require downloading and installation processes before they can be used. In contrast, websites provide a lower level of customizability and usability than mobile applications. However, they can also be accessed from personal computer browsers. Furthermore, website usage better suits search services, as Internet search engines can automatically refer to search service results. For example, a user entering "best dermatologist near me" could be referred to VerifyMed 2.0 search results by the search engine.

For these reasons, we opted for the website application for VerifyMed 2.0. It should be noted that the concept of the Progressive Web App (PWA) has recently become more popular. PWAs allow for the usage of websites as native applications for mobile phones or other devices. Therefore, the VerifyMed 2.0 website could be used as a native application as well.

Which Frontend Framework Should Be Used? There is a huge variety of potential frontend frameworks that could be used for the development of the VerifyMed 2.0 user interface. We opted for the NextJS framework [37]. This framework is built on top of the React frontend framework

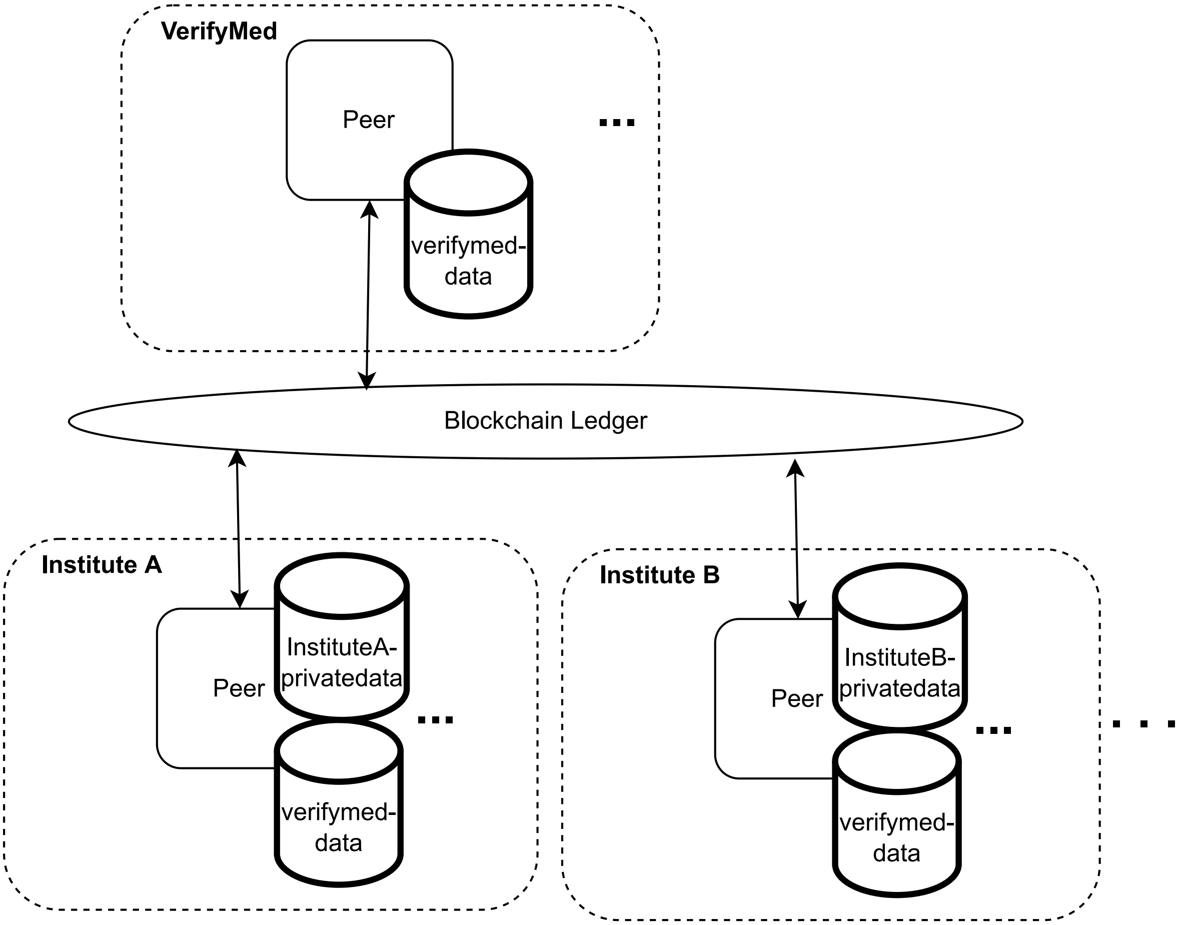


Fig. 4. Diagram illustrating private data collections of the VerifyMed 2.0 blockchain. There is a joint private collection of VerifyMed-data that stores data about all patients, healthcare institutes, physicians, and their treatments and reviews. Furthermore, each healthcare institute organization has a private collection that keeps a list of verified patients.

and additionally enables server-side rendering. This feature is useful for search services, such as VerifyMed 2.0, because the search results can be pre-rendered on the server.

Furthermore, NextJS offers a high level of Search Engine Optimization (SEO) that can increase the visibility of VerifyMed 2.0, thus increasing its growth potential. Additionally, NextJS enables users to create a single server instance that can serve as both a frontend and backend, thus simplifying the solution architecture.

3.3. *Addressing non-functional requirements*

In the previous section, we defined a list of quality attributes that the VerifyMed 2.0 architecture needs to address. We now go over these non-functional requirements and explain how the VerifyMed 2.0 solution fulfills them.

1) Privacy requirements

Untraceability of patient identity. Information about a patient's identity is encrypted inside treatment data and is readable only by the physician who created the treatment and the designated patient. Furthermore, when patients submit a treatment review, the healthcare institute performs checks and creates the review in the patient's stead, thus hiding any link between the patient and the treatment.

Anonymity of treatment instructions. The detailed treatment instructions are encrypted with the key available only to the physician who created the treatment and the designated patient.

Physician control of privacy. Physicians can temporarily hide their data from the public. This is performed by marking the data as hidden, and the VerifyMed 2.0 WebApp does not show such data on the website.

2) Security requirements

Access control. In order to prevent unauthenticated and unauthorized access, VerifyMed 2.0 leverages multi-layered access control. This means that the access control checks are performed on every layer of the system: first on the frontend of WebApp, then on the backend side of WebApp, and finally in smart contracts deployed on the Fabric channel.

Fraudulent treatments. Before treatment is created, it is checked whether the submitting physician is hired in the organization where the treatment is created. If the physician is not hired, the treatment is not created.

Fraudulent patients. As explained in Section 3.2, the healthcare institute checks whether it has verified the identity of the patient who submits a treatment.

Fraudulent reviews. In order to submit a treatment review, a physician's signature of the treatment needs to be provided. Therefore, creating a review without receiving treatment beforehand is impossible.

3) Availability requirements

No downtime when network is updated. The Hyperledger Fabric network enables organizations that are part of the same channel to propose changes to the channel parameters. The channel is reconfigured without downtime when the proposal receives an adequate number of votes. These channel updates can also be used to add or kick organizations from the channel.

Recoverability when minority misbehaves. If the minority of the channel organizations starts misbehaving, all other organizations can vote to remove disobedient organizations from the channel.

Consensus nodes downtime. Hyperledger Fabric's ordering service is fault-tolerant, and a consensus can be achieved as long as the majority of order nodes are active.

Recoverability of the data when nodes are lost. In Hyperledger Fabric, data stored in private collections are automatically distributed to all other peers that have access to them. Therefore, as long as there is at least one peer with the private collection active in the network, all other peers can reconstruct the blockchain state.

4) Scalability requirements

Minimal amount of data on the blockchain. In VerifyMed 2.0, the blockchain ledger only stores hash values of the real data stored inside private collections. Therefore, the usage of a blockchain ledger is reduced to a minimum.

Vertical and horizontal scalability. The VerifyMed 2.0 network is deployed using the IBM Blockchain Platform, which allows for dynamic vertical and horizontal scaling through the platform's interface. Furthermore, Hyperledger Fabric is a highly scalable blockchain, as demonstrated in Refs. [38,39].

Additionally, scalability is inherently embedded in the design of the architecture of the VerifyMed 2.0 blockchain. Extension of the network is a formalized process in which a new institute joins the network by adding its own peers that connect to the main channel.

3.4. Solution implementation

In this subsection, we describe the implementation of the VerifyMed 2.0 solution. We notice that the two main interacting components are the Fabric blockchain channel and the WebApp server. As shown in Fig. 5, users interact with the blockchain channel through WebApp, which submits transactions in their stead by using the private keys it keeps safe in its storage.

Therefore, we explain the implementation logic for each of these two components. In the first subsection, we focus on the blockchain channel and smart contract logic. In the second subsection, we cover the WebApp server and its deployment of frontend webpages and backend API functions.

3.5. Blockchain service

The VerifyMed 2.0 solution is built around a private permissioned Hyperledger Fabric blockchain. The Fabric blockchain is deployed on the Kubernetes cluster on the IBM Cloud service [30]. This blockchain cluster can be accessed through the interface of the IBM Blockchain Platform [31]. The user interface provides direct control over the peers, CAs, and orderers of the network.

Access to a deployed IBM Blockchain Platform can be shared with other IBM Cloud accounts, thus allowing them to create their organization and deploy their peers. Therefore, healthcare institutes that are willing to join the VerifyMed 2.0 network should first create an IBM Cloud account. Afterward, the VerifyMed team adds their account to the VerifyMed 2.0 Blockchain Platform, thus granting them the ability to deploy and control their peers inside the VerifyMed 2.0 network.

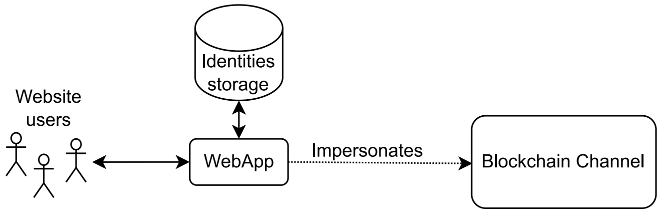


Fig. 5. Diagram represents an overview of the interaction between the website users and the implementation components.

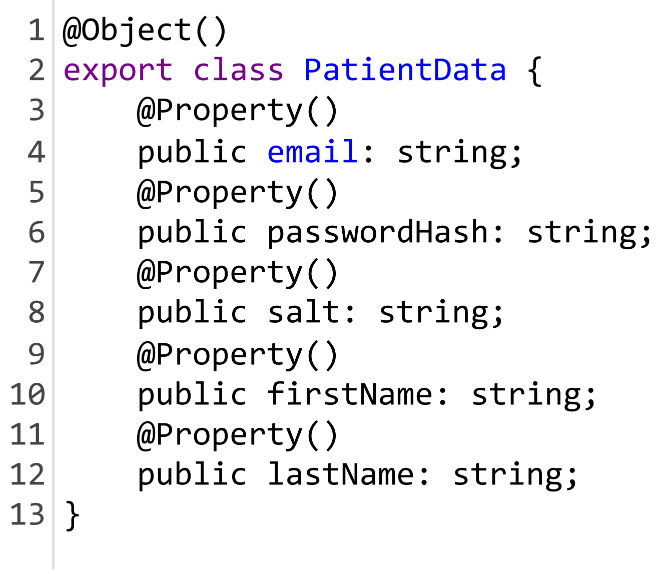
The VerifyMed 2.0 blockchain is implemented as a single Fabric channel, as shown in Fig. 2. All healthcare institutes of the VerifyMed 2.0 system control are assigned a Fabric organization and control their peers. All organizations of the channel can vote on the changes to the channel configuration and on the addition/removal of an organization from the channel. Furthermore, a majority of the votes are also required when a new smart contract is added to the channel or an existing one is updated.

We now take a deeper look into private collections that store data of the VerifyMed 2.0 blockchain. Afterward, we present the smart contract responsible for the functionality of the VerifyMed 2.0 solution.

1) Private Collections: The data of VerifyMed 2.0 are held in private collections. These collections are implemented as CouchDB[3] databases that store JSON mapped to a key value. As shown in Fig. 4, one private collection is shared by all organizations of the blockchain network and is named VerifyMed-data. Additionally, each healthcare institution has its private collection to store private data accessible by that organization only. We will now describe the data stored in these private collections.

The VerifyMed-data private data collection contains data about all healthcare institutes, physicians, and patients of the VerifyMed 2.0 network. All peers in the network can access this collection and write data into it by invoking corresponding smart contract functions. JSON objects of the user data are mapped to a key value in the database. The key value for all users is the email address that is used to log into the VerifyMed 2.0 system (see Fig 6).

Healthcare institute private data collection. Every healthcare institute controls a single organization in the VerifyMed 2.0 network. In Hyperledger Fabric, all organizations have an implicit private collection assigned to them. We use these implicit private collections to store the private data of healthcare institutes, thus making it hidden from all



```
@Object()
export class PatientData {
    @Property()
    public email: string;
    @Property()
    public passwordHash: string;
    @Property()
    public salt: string;
    @Property()
    public firstName: string;
    @Property()
    public lastName: string;
}
```

Fig. 6. TypeScript class defining a JSON object of patient data in VerifyMed 2.0.

[3] https://couchdb.apache.org/.

other organizations. Each healthcare institute holds a list of all the patients it has verified in this collection. These data are stored in an object mapped to the key verified patients. This object then represents a mapping of verified patient email addresses and a Boolean value true.

2) Smart contract: The business logic of the VerifyMed 2.0 solution is held in the smart contract deployed in the Fabric channel. The functions of this smart contract can be invoked by all members of the blockchain network, thus enabling interaction with the blockchain. All functions have access control that restricts which users can invoke the function. Some functions only read data from private collections, meaning they do not result in a transaction registered on the blockchain.

3.6. *Web application server*

The users of VerifyMed 2.0 interact with the website application or WebApp for short. As shown in Fig. 2, WebApp then interacts with the VerifyMed 2.0 blockchain in the user's stead. WebApp had its live deployment on the Heroku platform, but it was taken down after tests and experiments due to cloud operation costs. All applications deployed on Heroku were run inside Linux containers called *dynos*. Dynos provide scalability to applications based on their resource demands. Therefore, WebApp can be scaled vertically by upgrading its dyno type to have more resources, or it can be scaled horizontally by provisioning additional dynos.

WebApp is implemented in the NextJS framework [37]. This React-based framework is mainly used for the development of frontend features. However, the NextJS API route feature also enables the development of the backend API code that is not bundled and sent to the client's side. We next present the backend and the frontend implementation of the VerifyMed 2.0 WebApp.

1) Backend API routes: The main goal of the WebApp backend is to handle calls coming from the frontend by invoking a corresponding function on the blockchain smart contract. This is achieved through communication with the VerifyMed organization peers in the blockchain network. In case the function parameters contain private information that is being saved in the healthcare institute's private data collection (Section 3.5), only the peers from that healthcare institute organization are called to endorse the invoked transaction. Furthermore, WebApp impersonates blockchain users when invoking smart contract functions. Therefore, WebApp locally stores the identities and private keys of all users of the VerifyMed 2.0 network. This represents a privacy concern and can be solved by storing keys in the hardware security module (HSM). It should be noted that the WebApp backend APIs also perform access control before delegating the calls to the blockchain smart contract.
2) Frontend user interface: The VerifyMed 2.0 user interface is rendered on the server-side and sent to a client by the deployed NextJS WebApp. Server-side rendering improves the content's loading speeds and SEO. The user interface also implements access control, thus allowing access to webpages only to logged-in users of an appropriate role. Therefore, depending on the logged-in user, the user interface has different features and elements. In addition to backend validation, all user interface forms also validate the entered data. A complete overview of the VerifyMed 2.0 user interface can be seen in Ref. [40].

4. Discussion

We evaluate the implemented solution by running functional tests and measuring the solution's performance.

1) Functional test: Back in Table 2, we defined the list of functionalities that the VerifyMed 2.0 solution needs to provide for its users. In order to evaluate the implementation of these functionalities, we create unit and integration tests for the smart contract used. These tests evaluate each function regarding access control, validation checks, business logic checks, proper data storage, and appropriate return values. They test the isolated behavior of functions, as well as their interoperability. There are a total of 113 written tests, which result in 100% test coverage of the smart contract code.
2) Performance measurements: We measure the performance of the VerifyMed 2.0 solution using the Hyperledger Caliper [41] testing framework. We define Caliper benchmarks for six different VerifyMed smart contract functionalities. The tests are performed by connecting the Caliper client directly with the deployed smart contract on the IBM Cloud and measuring the performance of the functionalities.

The deployed IBM Cloud Kubernetes cluster comprises a single worker pool that contains two worker nodes, each utilizing 4 vCPUs and 16 GB RAM. The performance tests are run for 60 s, and the Caliper benchmark is configured to use 4 benchmark workers in parallel. The results can be seen in Table 3.

As can be noted, approximately 0.3 treatment submissions per second can be achieved when the system is overloaded (addTreatment function). This is a considerable improvement over the VerifyMed 1.0 solution, where the theoretical maximum while using the whole public Ethereum network as miners was 1.7 treatment submissions per second.

It should be noted that the achieved results can be further optimized. The main performance bottleneck of the current solution is the data structure architecture. In private data collections, most of the data are saved under a single key to a single physician. Furthermore, Hyperledger Fabric uses optimistic locking to keep the data collections consistent, thus resulting in frequent read/write conflicts that cap the potential transactions per second (TPS).

3) Social impact: As demonstrated in Section 1.1, VerifyMed 2.0 makes a significant contribution to the existing research landscape as the pioneering permissioned blockchain solution that establishes trust relationships among physicians, patients, and healthcare institutes. The VerifyMed 2.0 solution potentially has a significant social impact in the emerging virtualized healthcare domain. The goal is to start as a small system comprising a few healthcare institutes and their physicians and slowly gain size and scale up as more users join the network. The larger the scale of the VerifyMed 2.0 network is, the more information and usability the system provides. It should be noted that one of the healthcare institutes of the VerifyMed 2.0 network can be owned by the VerifyMed project team. Such an institute could enable the VerifyMed team to verify physicians' certificates, thus making it easier to expand the network when it is small. While the VerifyMed 2.0 network grows in size, new potential use cases for the platform may arise. Some examples of such use cases are as follows:
   - The integrity of the data saved in the blockchain provides non-repudiation of data that have been submitted. Therefore, legislation services may refer to data submitted in the VerifyMed 2.0 platform as proof of a physician's negligence toward a patient. The patient can decrypt the hidden instructions of the treatment and the physician's signature, thus proving the physician's malpractice.
   - As the VerifyMed 2.0 solution incorporates multiple separate healthcare institutes, it eases labor mobility for physicians. Healthcare institutes will give more credibility to physicians who have been part of the VerifyMed 2.0 network, as they can quickly verify their authority, experience, and competence.
   - The potential global scale of VerifyMed 2.0 would enable easier comparison of physician performance and patient satisfaction across different countries and regions, thus providing data useful for study and analysis.

Table 3
Measurement results of Hyperledger Caliper benchmarks performed over VerifyMed 2.0 smart contract functions. Send rate and throughput are expressed in transactions per second (TPS). Average Latency is the average transaction latency of the 60 s benchmark, during which the system is put through maximum TPS. Single transaction latency represents the average latency when a single transaction is submitted to the system.

Smart contract function	Send transactions	Send rate (TPS)	Throughput (TPS)	Average latency (s)	Single transaction latency (s)
getPhysicians	81	1.4	1.2	9.42	1.87
getTreatments	80	1.4	1.3	6.87	0.7
getPatients	860	14.3	13.4	6.29	0.21
addTreatment	20	0.4	0.3	9.68	3.62
submitReview	20	0.4	0.3	12.09	3.22
switchHiddenPhysicianData	20	0.4	0.3	10.78	3.46

4) Potential improvements: The VerifyMed 2.0 solution represents an improvement of the shortcomings of the old VerifyMed 1.0 project. However, there is still space for future work. We now present the potential improvements and propose how they can be solved.
   - Storage of user private keys—The identities and private keys of all VerifyMed 2.0 network users are stored inside the WebApp server, which represents a significant security and privacy vulnerability. If an adversary, through some methods, gains access to the WebApp server, the private keys of all users can be compromised. The standard solution used in the industry to safely store certificates and private keys is the hardware security module (HSM), which can easily be integrated into the current VerifyMed 2.0 system.
   - Execution environments—The WebApp server, during its runtime, has access to highly confidential data, such as private keys and sensitive treatment data. The industry standard way to combat this issue is using the trusted execution environment (TEE), which guarantees the confidentiality and integrity of the code executed inside it.
   - Authentication service—The user password hash values are stored inside the private collections that are accessible by all the peers of VerifyMed 2.0, thus exposing them to unnecessary security risks. The more accessible and secure solution would be the usage of third-party single sign-on (SSO) solutions that provide a single digital identity for a user that can be used across multiple different services.
   - Testing different network configurations—The current VerifyMed 2.0 blockchain network uses the default configuration for the network parameters. Therefore, changing the parameters or experimenting with different numbers of orderers in the network could further boost the performance. Furthermore, the IBM Cloud platform allows for experimenting with different Kubernetes constellations by varying the number of deployed pods or the resource allocation of each pod.
   - Other minor improvements
     - A more extensive security analysis of the solution and underlying technologies is needed. As of now, only the basics are analyzed, as this is an early-phase project.
     - The use of CouchDB indexing inside the private collection of the blockchain can increase the performance of data queries.
     - Adding pagination in the smart contract functions and the WebApp backend would increase the load times when significant amounts of data are to be served.
     - Currently, physicians can only temporarily hide all of their data at once. Giving them more granular control, where they could choose which information to hide, would give them more authority over their privacy. However, physicians who hide their data should not be allowed to manipulate their rating and public approval.
     - Storing additional information about the system users, such as profile pictures of physicians and healthcare institutes, can increase the usability and appeal of the website.
     - Adding confirm dialogues in the user interface can prevent many unintentional user errors.

5. Conclusions

In this paper, we leveraged Hyperledger Fabric to create the VerifyMed platform that facilitates trust relations in the healthcare industry. It enables patients to verify the credibility of physicians' certification, experience, and competence. Additionally, physicians are provided with an online tool that stores and verifies their licenses, thus enabling ease of movement between different healthcare institutes and countries. The proposed platform enables healthcare institutes to govern the VerifyMed network by voting on network configuration and membership matters.

The solution does not impose transaction fees compared to other smart contract solutions. Furthermore, the solution architecture is scalable and allows the addition of new network users and their corresponding Hyperledger Fabric peers. The solution utilizes various cryptographic techniques to provide user authentication while remaining GDPR compatible. The solution was deployed, and its usability was tested through the created user interface. Furthermore, we utilized unit and integration tests to affirm the solution's functionalities. Additionally, we measured that the current deployment enables 0.3 treatment submissions per second, thus being a considerable improvement over the old VerifyMed solution. Nonetheless, there is still space for future work and improvements to the VerifyMed platform, as explained in Section 4, such as addressing the storage and handling of users' credentials and private keys.

Credit author statement

Aleksandar **Nedaković**: Data curation, software, visualization, writing—draft preparation; **Anton Hasselgren**: Supervision, writing—draft reviewing, conceptualization; **Katina Kralevska**: Writing—draft reviewing, supervision, conceptualization; **Danilo Gligoroski**: Writing—draft reviewing, supervision, conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H.K. Kim, C.W. Lee, Relationships among healthcare digitalization, social capital, and supply chain performance in the healthcare manufacturing industry, Int. J. Environ. Res. Publ. Health 18 (4) (2021) 1417, https://doi.org/10.3390/ijerph18041417.

[2] Y. Zhou, S. Chen, Y. Liao, et al., General perception of doctor–patient relationship from patients during the covid-19 pandemic in China: a cross-sectional study, Front. Public Health 9 (2021), 646486, https://doi.org/10.3389/fpubh.2021.646486.

[3] A.S. Tankwanchi, A. Hagopian, S.H. Vermund, African physician migration to high-income nations: diverse motives to emigrate ("we are not florence nightingale") or stay in Africa ("there is no place like home") comment on "doctor retention: a cross-sectional study of how Ireland has been losing the battle", Int. J. Health Pol. Manag. 10 (10) (2021) 660, https://doi.org/10.34172/2Fijhpm.2020.219.

[4] J.-A.H. Rensaa, D. Gligoroski, K. Kralevska, et al., Verifymed-a blockchain platform for transparent trust in virtualized healthcare: proof-of-concept, in: Proceedings of the 2020 2nd International Electronics Communication Conference, ACM, 2020, pp. 73–80, https://doi.org/10.1145/3409934.3409946.

[5] Q. Wang, S. Qin, A hyperledger fabric-based system framework for healthcare data management, Appl. Sci. 11 (24) (2021), 11693, https://doi.org/10.3390/app112411693.

[6] C. Stamatellis, P. Papadopoulos, N. Pitropakis, et al., A privacy-preserving healthcare framework using hyperledger fabric, Sensors 20 (22) (2020) 6587, https://doi.org/10.3390/s20226587.

[7] M. Abdul-Moheeth, M. Usman, D.T. Harrell, et al., Improving transitions of care: designing a blockchain application for patient identity management, Blockchain in Healthcare Today 5 (2022), https://doi.org/10.30953/2Fbhty.v5.200.

[8] M. de Vasconcelos Barros, F. Schardong, R. Felipe Custódio. Leveraging Self-Sovereign Identity, Blockchain, and Zero-Knowledge Proof to Build a Privacy-Preserving Vaccination Pass. arXiv. 2022. preprint. arXiv: 2202.09207.

[9] A. Abid, S. Cheikhrouhou, S. Kallel, et al., Novidchain: blockchain-based privacy-preserving platform for covid-19 test/vaccine certificates, Software Pract. Ex. 52 (4) (2022) 841–867, https://doi.org/10.1002/spe.2983.

[10] M. Attaran, Blockchain technology in healthcare: challenges and opportunities, Int. J. Healthc. Manag. 15 (1) (2022) 70–83, https://doi.org/10.1080/20479700.2020.1843887.

[11] J. Andrew, D.P. Isravel, K.M. Sagayam, et al., Blockchain for healthcare systems: architecture, security challenges, trends and future directions, J. Netw. Comput. Appl. 215 (2023), 103633, https://doi.org/10.1016/j.jnca.2023.103633.

[12] A. Rejeb, H. Treiblmaier, K. Rejeb, et al., Blockchain research in healthcare: a bibliometric review and current research trends, J. Digit. Inf. Manag. 3 (2021) 109–124, https://doi.org/10.1007/s42488-021-00046-2.

[13] L. Soltanisehat, R. Alizadeh, H. Hao, et al., Technical, temporal, and spatial research challenges and opportunities in blockchain-based healthcare: a systematic literature review, IEEE Trans. Eng. Manag. 70 (1) (2020) 353–368, https://doi.org/10.1109/TEM.2020.3013507.

[14] A. Azaria, A. Ekblaw, T. Vieira, et al., Medrec: using blockchain for medical data access and permission management, in: Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), IEEE, 2016, pp. 25–30, https://doi.org/10.1109/OBD.2016.11.

[15] K. Fan, S. Wang, Y. Ren, et al., Medblock: efficient and secure medical data sharing via blockchain, J. Med. Syst. 42 (8) (2018) 1–11, https://doi.org/10.1007/s10916-018-0993-7.

[16] Y. Sun, R. Zhang, X. Wang, et al., A decentralizing attribute-based signature for healthcare blockchain, in: Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2018, pp. 1–9, https://doi.org/10.1109/ICCCN.2018.8487349.

[17] J. Hathaliya, P. Sharma, S. Tanwar, et al., Blockchain-based remote patient monitoring in healthcare 4.0, in: Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), IEEE, 2019, pp. 87–91, https://doi.org/10.1109/IACC48062.2019.8971593.

[18] K.N. Griggs, O. Ossipova, C.P. Kohlios, et al., Healthcare blockchain system using smart contracts for secure automated remote patient monitoring, J. Med. Syst. 42 (7) (2018) 1–7, https://doi.org/10.1007/s10916-018-0982-x.

[19] T. Bocek, B.B. Rodrigues, T. Strasser, et al., Blockchains everywhere—a use-case of blockchains in the pharma supply-chain, in: Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, 2017, pp. 772–777, https://doi.org/10.23919/INM.2017.7987376.

[20] K.A. Clauson, E.A. Breeden, C. Davidson, et al., Leveraging blockchain technology to enhance supply chain management in healthcare: an exploration of challenges and opportunities in the health supply chain, Blockchain in healthcare today 1 (2018), https://doi.org/10.30953/bhty.v1.20.

[21] C. Thomas, V. Bindu, A.A. Aby, et al., Blockchain-based medical insurance storage systems, in: A.K. Tyagi, A. Abraham (Eds.), Recent Trends in Blockchain for Information Systems Security and Privacy, CRC Press, Boca Raton, FL, 2021, pp. 219–235, https://doi.org/10.1201/9781003139737.

[22] A.R. Hevner, S.T. March, J. Park, et al., Design science in information systems research, MIS Q. 28 (1) (2004) 75–105. https://misq.umn.edu/design-science-information-systems-research.html. (Accessed 15 October 2022).

[23] A.M. Ferreira, A.R. da Silva, A.C. Paiva, Towards the art of writing agile requirements with user stories, acceptance criteria, and related constructs, in: H. Kaindl, M. Mannion, L. Maciaszek (Eds.), Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), SciTePress, 2022, pp. 477–484, https://doi.org/10.5220/0011082000003176.

[24] A.R. Amna, G. Poels, Systematic literature mapping of user story research, IEEE Access 10 (2022) 51723–51746, https://doi.org/10.1109/ACCESS.2022.3173745.

[25] M. Macdonald, L. Liu-Thorrold, R. Julien, The blockchain: a comparison of platforms and their uses beyond bitcoin, Work. Pap. (2017) 1–18, https://doi.org/10.13140/RG.2.2.23274.52164.

[26] D. Elangovan, C.S. Long, F.S. Bakrin, et al., The use of blockchain technology in the health care sector: systematic review, JMIR medical informatics 10 (1) (2022), e17278, https://doi.org/10.2196/17278.

[27] K. Wüst, A. Gervais, Do you need a blockchain?, in: Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT) IEEE, 2018, pp. 45–54, https://doi.org/10.1109/CVCBT.2018.00011.

[28] M.J.M. Chowdhury, M.S. Ferdous, K. Biswas, et al., A comparative analysis of distributed ledger technology platforms, IEEE Access 7 (2019) 167930–167943, https://doi.org/10.1109/ACCESS.2019.2953729.

[29] H. Yu, H. Sun, D. Wu, et al., Comparison of smart contract blockchains for healthcare applications, AMIA Annual Symposium Proceedings (2019) 1266, 2019.

[30] IBM cloud—set of cloud computing services, Available online: https://www.ibm.com/cloud. (Accessed 18 December 2022).

[31] IBM blockchain platform: hyperledger fabric support edition, Available online: https://www.ibm.com/blockchain/platform. (Accessed 18 December 2022).

[32] Hyperledger fabric—security model, Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/security_model.html. (Accessed 20 December 2022).

[33] S. Brotsis, N. Kolokotronis, K. Limniotis, et al., On the security and privacy of hyperledger fabric: challenges and open issues, in: Proceedings of the 2020 IEEE World Congress on Services (SERVICES), IEEE, 2020, pp. 197–204, https://doi.org/10.1109/SERVICES48979.2020.00049.

[34] E. Androulaki, A. Barger, V. Bortnikov, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, Proceedings of the thirteenth EuroSys conference. ACM. (2018) 1–15, https://doi.org/10.1145/3190508.3190538.

[35] P. Voigt, A. Von dem Bussche 1st (Eds.), The EU General Data Protection Regulation (GDPR): a Practical Guide, Springer, Cham, 2017, pp. 10–5555, https://doi.org/10.1007/978-3-319-57959-7.

[36] A. Hasselgren, P.K. Wan, M. Horn, et al. GDPR Compliance for Blockchain Applications in Healthcare. arXiv. 2020. preprint. arXiv: 2009.12913.

[37] NextJS—enables production of React applications that scale, Available online: https://nextjs.org/. (Accessed 20 December 2023).

[38] M. Kuzlu, M. Pipattanasomporn, L. Gurses, et al., Performance analysis of a hyperledger fabric blockchain framework: through-put, latency and scalability, in: Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, 2019, pp. 536–540, https://doi.org/10.1109/Blockchain.2019.00003.

[39] H.H. Pajooh, M.A. Rashid, F. Alam, et al., Experimental performance analysis of a scalable distributed hyperledger fabric for a large-scale IoT testbed, Sensors 22 (13) (2022) 4868, https://doi.org/10.3390/s22134868.

[40] A. Nedaković, Analysis and Improvements of VerifyMed—The Blockchain Solution for Virtualized Healthcare Trust Relations, 2022, p. 104+22. http://urn.fi/URN:NBN:fi:aalto-202208285217. (Accessed 18 December 2022).

[41] Hyperledger caliper—blockchain performance benchmarking for Hyperledger Besu, Hyperledger fabric, Ethereum and FISCO BCOS networks, Available online: https://hyperledger.github.io/caliper/. (Accessed 22 December 2022).