Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 11-41
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

SEMR: A Framework for Sharing Electronic Medical Records Using Emerging Technologies

Leina Abdelgalil[1], Mohamed Mejri[2], Djedjiga Mouheb[2]

[1]College of Computer Science and Information Technology, Sudan University of Science and Technology, Khartoum, Sudan
[2]Department of Computer Science, Laval University, Quebec, Canada
Email: leina.nazar@gmail.com, Mohamed.Mejri@ift.ulaval.ca, djedjiga.mouheb@ift.ulaval.ca

How to cite this paper: Abdelgalil, L., Mejri, M. and Mouheb, D. (2024) SEMR: A Framework for Sharing Electronic Medical Records Using Emerging Technologies. *Journal of Computer and Communications*, **12**, 11-41.
https://doi.org/10.4236/jcc.2024.125002

Received: March 12, 2024
Accepted: May 12, 2024
Published: May 15, 2024

Abstract

Electronic Medical Records (EMRs) play a crucial role in healthcare systems, providing secure, comprehensive medical information and reducing errors. However, they are often fragmented and stored in separate databases owned by different institutions (e.g., hospitals, labs, clinics), posing challenges for healthcare professionals in sharing, preserving, and monitoring patients' EMRs. To address the aforementioned challenges, this paper proposes a framework integrating many new emerging technologies and underscores their pivotal role in revolutionizing EMR systems. In particular, Hyperledger Indy empowers patients with complete authority over their EMRs, while Hyperledger Fabric manages authentication, authorization, and traceability. The InterPlanetary File System (IPFS) is used for secure sharing of EMRs. The Internet of Medical Things (IoMT) achieves real-time monitoring of patients' health characteristics. Finally, WebAuthn provides strong protection for used cryptographic keys and cryptographic operations.

Keywords

Hyperledger Indy, Fabric, IoMT, IPFS, WebAuthn

1. Introduction

The paramount importance of medical record administration in enhancing the safety and treatment of patients has been emphasized by world health organizations such as the American Medical Association (AMA), the Office of the National Coordinator for Health Information Technology (ONC), and the World Health Organization (WHO). Health data sharing enables patients to connect with healthcare service professionals to obtain medical history data, ultimately

allowing earlier diagnosis and appropriate treatment suggestions. The majority of healthcare organizations currently utilize centralized local databases to store the electronic medical records (EMRs) of their patients. However, these organizations often neglect to address the issues of privacy and access control when it comes to medical information, leaving patients with limited access and no control over who has the privilege of accessing their medical data generated by laboratory or IoMT devices. Nevertheless, despite their utmost efforts to safeguard their client's confidential data, this solution is plagued by numerous disadvantages. Typically, people often seek many healthcare attribute providers over their lives, resulting in their electronic medical records (EMRs) being spread elsewhere. This can lead to essential information not being accessible when it is most needed. This situation is common, even when the healthcare professionals for the patients are located in the same city. The situation will be most dire for individuals who seek medical services from healthcare providers in foreign countries. Patients often lack access to their complete medical records and have limited control over the authorization of data access. Consequently, there are instances where confidential data is inadvertently disclosed, leading to detrimental consequences for the individuals affected, such as unemployment or heightened insurance costs.

Classical cryptographic methods, such as encryption, digital signatures, and hash functions, provide fundamental components for effectively achieving certain essential security features, including secrecy, integrity, and authenticity. However, achieving additional security goals such as availability, transparency, immutability, and automation necessitates not only the use of many traditional cryptographic systems but also the incorporation of further solutions such as peer-to-peer networks, consensus algorithms, and smart contracts.

To address these challenges, various emerging technologies such as Blockchain, Hyperledger, and IPFS, have become readily available, facilitating the development of secure systems for various applications, including those with novel and highly needed characteristics. This wide array of novel construction components provides developers the ability to create safe structures for varied applications with more ease, including highly sought-after features. The unique qualities of blockchains have sparked considerable interest and led to dramatic changes in various corporate applications, including medical care. According to [1], the Blockchain market in healthcare is anticipated to increase at a compound annual growth rate (CAGR) of 52.48% between 2023 and 2028, passing from USD 2.37 billion in 2023 to USD 19.52 billion by 2028.

Blockchain technology has effectively addressed several difficulties associated with the Internet of Medical Things (IoMT) due to its decentralized processing architecture, data immutability, transactional security, and privacy features. This makes Blockchain a promising technology for the development of improved solutions for protecting, accessing, delegating, and ensuring the confidentiality, integrity, and anonymity of patient records.

In this context, the primary aim of this study is to tackle the challenges associated with existing electronic medical record management systems, especially pertaining to integrity, availability, and access control. In particular, the paper proposes a comprehensive and privacy-preserving framework, namely SEMR, for effectively sharing electronic medical records. To this end, the proposed framework encompasses several intriguing technologies, including Hyperledger Fabric, Hyperledger Indy, Self Sovereign Identity (SSI), InterPlanetary File System (IPFS), and Internet of Medical Things (IoMT). The main contributions of the proposed framework are summarized as follows:

- We design a decentralized off-chain storage system for IoMT data, where Electronic Medical Records (EMRs) may be accessed from any location worldwide. Using the IPFS technology allows EMRs to be effectively and securely safeguarded, guaranteeing the records' confidentiality and immutability, and enhancing the scalability of EMR storage systems.
- We present a blockchain-based solution for managing patients' access control policy. To this end, we employ Hyperledger Indy to empower patients with complete authority and control over their EMRs. Moreover, we leverage Hyperledger Fabric to manage the patients' access control delegations, enabling them to provide access to any individual or group at their discretion.
- We devise a mechanism for remote patient monitoring and remote anonymous medical assistance. In this regard, IoMT is employed to achieve the monitoring of patient's health characteristics by healthcare professionals through the use of wearable linked devices.

Contributions:

The primary contribution of this paper lies in its proposition of an EMR management architecture designed to deliver a comprehensive suite of key features. These include ensuring high levels of availability, confidentiality, and integrity, alongside implementing important functionalities such as zero-knowledge proof, ownership proof, revocation mechanisms, anti-correlation measures, and real-time capabilities. This architecture represents a significant advancement in addressing the multifaceted requirements of effective EMR management, promising enhanced security, privacy, and efficiency in handling medical data.

The remainder of this article is organized in the following manner. Section 2 presents a set of notions essential for comprehending the proposed approach. The framework architecture proposed for sharing medical records based on Hyperledger Indy and IoMT is illustrated in Section 3. The literature review is offered in Section 4. Finally, Section 5 provides concluding remarks and observations.

2. Preliminaries

In this section, we discuss the preliminary aspects of the topic at hand.

2.1. Blockchain Technology

The blockchain technology was first introduced to address the issue of double

spending [2]. The blockchain operates as a decentralized network, facilitating a distributed and unchangeable log of transactions that are cryptographically signed and arranged into blocks. In the blockchain system, the connection between blocks is established by the utilization of a cryptographic hash function. This function ensures that each block is associated with the preceding block. Furthermore, these blocks are extensively dispersed across a decentralized network without any centralized control, rendering it arduous to effortlessly introduce a counterfeit block or challenge the legitimacy of a block. These qualities ensure that transactions on a blockchain network cannot be altered, allowing data to be stored without any modifications or ability to trace the record history [3]. Furthermore, every blockchain has a distinct consensus method to reach an agreement on a new block and safeguard against potential assaults, such as Sybil attacks [4]. The choice of a reliable entity to create a new block in blockchain may be determined using algorithms like as proof of work [5], and proof of stake [6]. Specifically, every block consists of two main components: a header and a body. The header encompasses essential information such as the date and the hash value of the previous block, which is instrumental in generating the hash value of the current block. On the other hand, the body of the block contains the recorded transactions. The data within a blockchain is decentralized and stored over a network of numerous computers, commonly referred to as nodes.

2.1.1. Blockchain Characteristics

The key characteristics and benefits of blockchain are detailed below.

1) **Ledger**: Transactions in a blockchain, unlike traditional databases, cannot be edited or erased.

2) **Enhanced Security**: Blockchain delivers inherent traceability, integrity, and availability.

3) **Lack of Trust**: Even if blockchain nodes are run by parties who do not trust one another and have competing interests, a blockchain can meet its security standards.

4) **Decentralization**: In traditional transaction management systems, transactions are verified by a trusted agency. As a result, centralized services cause a number of concerns, including increased prices, performance issues, and single-point failure. Transactions between two peers (computers) in blockchain are validated without the involvement of a central body. Consequently, decentralization lowers the expense and danger of a single point of failure.

5) **Immutability**: A blockchain is composed of interconnected blocks, forming a sequential chain structure, whereby each block contains data pertaining to the hash of the preceding block. In the event that an individual, such as a hacker, attempts to modify the data contained within the preceding block, the integrity of the whole blockchain is compromised, rendering it invalid. Consequently, any modification or alteration may be promptly detected. This particular characteristic serves to safeguard the integrity of the data.

6) **Transparency**: All participants are granted equitable access to the block-

chain network. Furthermore, all valid network transactions are available to all participants. As a result, blockchain data is transparent and easily verified.

7) **Traceability**: Each transaction stored in the blockchain is given a timestamp. As a result, by evaluating the blockchain data with associated timestamps, users can trace the merchandise.

8) **Anonymity**: A unique address allows each user to communicate with the blockchain network.

9) **Auditability**: Because each transaction on the blockchain is saved after confirmation with a timestamp, the participant may simply review and trace past records by accessing any node in the blockchain.

2.1.2. Blockchain Classifications

There exist several classifications of blockchain. The following elaborates on the different classifications in a comprehensive manner.

1) **Public Blockchain (Permissionless)**: This network is open to all participants, allowing anyone to access and contribute to the public blockchain by reading and adding transactions [7]. Moreover, this network is accessible to anyone with an internet connection who possesses necessary authorization to engage as a miner for the purpose of block mining. Upon a user's entry into the network, it becomes possible for any individual to ascertain the existence of an entity associated with the given address, without being able to ascertain the specific identity of this entity. In this network, users have the capability to analyze transactions and engage in the process of block mining for the network. Ethereum and Bitcoin are both illustrative instances of open blockchains.

2) **Private Blockchain (Permissioned)**: A private blockchain serves the same purpose as a public blockchain. In this particular blockchain framework, only a singular organization belonging to the same group is granted authorization to access and submit transactions [7]. Nevertheless, it is governed by access control regulations. Private blockchains are frequently employed in several domains such as electronic voting, Electronic Medical record (EMR) administration, and supply chain management. Hyperledger Fabric and Ripple exemplify private blockchain solutions. Access to a private network is restricted exclusively to individuals who have received an invitation from an authorized user.

3) **Consortium Blockchain**: In this network, multiple groups in an organization can read and submit transactions [7]. Furthermore, it is exclusively accessible to pre-registered groups of nodes. No operation can be carried out by a single organization without the permission of other organizations. Consortium blockchains include Corda, Hyperledger Fabric, as well as Quorum.

Table 1 presents various blockchain types together with some characteristics such as Access, Participants, Transaction Speed, Scalability, and Example.

2.2. Hyperledger Indy

Indy [8] is a permissioned public Hyperledger, allowing unrestricted access for reading while writing limited to authorized principals. The platform is established based on Sovrin [9] and aims to provide a self-sovereign identity system

Table 1. Classifications of blockchain.

	Public	Private	Consortium
Gain Access	All participants	Single organization	Multiple selected organizations
Participants	Anonymous and permissionless	Known identities and permissioned	Known identities and permissioned
Transaction Speed	Slow	Fast	Fast
Scalability	High	Low/Medium	Low/Medium
Example	Ethereum, Bitcoin, Hyperledger Indy	Hyperledger Sawtooth, Hyperledger Fabric, Iroha	Hyperledger Fabric, Quorum, and Corda

that utilizes blockchain technology. For example, in Canada, it is used by the British Columbia Government to implement their eID solution [10] and share Electronic Medical Records [11].

The key role of Indy is to enable users to prove their ownership of their provided identity (a set of attributes). Every identity contains a Decentralized Identifier (DID) which is linked to one of the owner's public keys and saved into the hyperledger. More precisely, Indy could be seen as a ledger that contains certificates linking DIDs to public keys and signed by a Trust Anchor TA. A TA is an individual or an organization (government, insurance, hospital, clinic, etc.) that are already known and trusted by the Hyperledger Indy. They act as certification authorities, adding a new Trust Anchor to the ledger. They have the right to write in the ledger. Whenever a TA delivers a document (an Identity) containing a fresh DID, they add a link (certificate) between the DID and its public key to Indy, as shown in Table 2.

For instance, the first entry in the ledger attests that the identity owner containing the DID D_a^0 is the one who knows the private key associated with k_a^0. The main information of this entry is "D_a, k_a^0, DID-TA, $\left\{H\left(D_a, k_a^0, \text{DID-TA}\right)\right\}_{k_{\text{TA}}^{-1}}$", where DID-TA is the DID of the Trust Anchor that has signed this record. Indy contains many transactions having this kind of record. The same TA could sign many associations between DIDs and public keys, as shown in Table 2.

The Hyperledger Indy also contains information related to TAs. We find a certificate for each trusted authority containing its DID, its service endpoint (URI or IP address, transport protocol, and port), public key, the schema of its delivered documents (the list of attributes that could be found in the credential), etc. Therefore, having the DID of a document, we can easily get the producer's public key and join him through its service endpoint.

Any part of an Identity could be considered as a credential containing claims implemented by a list of pairs (attribute, value). The credential also contains the

Table 2. Links between DIDs and public keys in the Indy Hyperledger.

DID	PUBLIC-KEY	DID-TA	$\{Hash(\text{DID}, \text{PUBLIC-KEY}, \text{DID-TA})\}_{k_{TA}^{-1}}$
D_a^0	k_a^0	D_{TA_1}	$\left\{H\left(D_a^0, k_a^0, D_{TA_1}\right)\right\}_{k_{TA_1}^{-1}}$
D_a^1	k_a^1	D_{TA_1}	$\left\{H\left(D_a^1, k_a^1, D_{TA_1}\right)\right\}_{k_{TA_1}^{-1}}$
D_b	k_b	D_{TA_2}	$\left\{H\left(D_b, k_b, D_{TA_2}\right)\right\}_{k_{TA_2}^{-1}}$
D_c	k_c	D_{TA_3}	$\left\{H\left(D_c, k_c, D_{TA_2}\right)\right\}_{k_{TA_3}^{-1}}$
⋮	⋮	⋮	⋮

proof of claims involving the issuer's signature, as shown in Figure 1.

Assume that an agent B wants to prove to A that he has an attribute included in a credential containing the DID D_b; then, they need to proceed as follows:

- The agent B sends the credential containing the DID D_b to A.
- The agent A reads from the Hyperledger Indy, the public key k_b associated with the DID D_b.
- The agent A asks B to prove that he owns the private key associated with k_b using a challenge-response protocol. Figure 2 gives an example of a simple challenge-response protocol, but it could be replaced by any other one in the proposed architecture.

At the first step, A sends to B a challenge containing a fresh random number N_a (called a nonce). At the second step, B returns the challenge signed using the appropriate private key k_b^{-1}. To verify if the challenge was answered correctly, the agent A checks if was signed using the private key associated with k_b.

DIDs and public keys are randomly generated by the document's owner, in which the DID appears, and they are not necessarily correlated to the name of their owners, allowing them to receive anonymous services.

To streamline the framework presentation, hereafter, we use

$$D_a, k_a^0, \left\{H\left(D_a, k_a^0\right)\right\}_{k_{TA^{-1}}}$$

instead of

$$D_a, k_a^0, \text{DID-TA}, \left\{H\left(D_a, k_a^0, \text{DID-TA}\right)\right\}_{k_{TA^{-1}}}$$

when DID-TA is clear from the context.

2.3. Self-Sovereign Identity (SSI)

Self-sovereign identification [12] emerged in 2016 as a digital identity paradigm that can enable secure and trusted online transactions by letting individuals and organizations to have control of their digital identities the way they do with their physical ones. With SSI, people can share the minimum required identity information needed for a particular transaction without intermediaries. SSI recognizes

Credential		
Metadata	:	issuer, expiration date, etc.
claim(s)	:	$attribute_1 : value_1, \ldots, attribute_n : value_n$
proof(s)	:	signature information

Figure 1. Credential format in Hyperledger Indy.

$$\begin{array}{lllll} 1. & A & \rightarrow & B & : & A, B, N_a \\ 2. & B & \rightarrow & A & : & \{H(A, B, N_a)\}_{k_b^{-1}} \end{array}$$

Figure 2. Challenge-response protocol.

that identification encompasses more than just the act of checking in. Identity may be broadened to include more applications via the use of verifiable attestations, known as credentials, to substantiate various aspects of one's personal information. SSI employs credentials that are both verified and reliable. Immutability and control may be achieved by implementing IDs in a decentralized way on blockchains specifically intended for identity operations, such as Hyperledger Indy.

2.4. Hyperledger Fabric

Hyperledger Fabric [13] is an open-source blockchain framework, developed in 2015 under the auspices of the Linux Foundation. The mentioned technology is a Hyperledger framework that facilitates fine-grained access management. The primary objective of this system is to function as a decentralized Hyperledger, possessing the capability to support a diverse array of applications and serve as a foundation for the development of Decentralized Applications (DApps). Hyperledger Fabric serves as a fundamental framework for constructing applications or solutions with a modular structure. Hyperledger Fabric enables the seamless integration of components, such as consensus and membership services using a plug-and-play mechanism.

2.5. Smart Contract

Smart contracts [14] represent a substantial advancement in the field of blockchain technology. According to [15], smart contracts were first introduced in 1990 as a digital transaction protocol designed to execute the conditions of an agreement. Smart contracts are software applications that possess the ability to execute themselves, and are designed to define the regulations and limitations for transactions that occur within a blockchain system. The data is stored and performed on the blockchain platform to generate more information for the distributed ledger. The primary objective of smart contracts is to facilitate the automated execution of conditional transaction checks and enhance the speed of transactions. Many users lack control over smart contracts. Smart contracts are

employed in several industries, including healthcare, trade finance, voting, insurance, and shipping, among others. Hyperledger Fabric, alternatively referred to as *Chaincode*, is implemented in several programming languages such as Solidity, Java, Go, and JavaScript. A smart contract, once placed in the blockchain, possesses the characteristic of immutability, rendering it incapable of being altered or eliminated.

2.6. Inter Planetary File System (IPFS)

The IPFS system [16] is constructed using open source software and functions on a decentralized network, where information is dispersed across many nodes instead of being centralized. IPFS's content-addressing guarantees data integrity by assigning a unique hash to each file and block, therefore certifying that the content has not been altered. IPFS gets data from the closest or most efficient nodes, hence minimizing bandwidth consumption and enhancing performance. IPFS [16] files are stored as discrete entities known as objects, whereby each object is interconnected with other objects through a system of links. IPFS significantly propels the evolution of internet design by an immense distance. Content addressing, in contrast to HTTP, enables the separation of data from its specific server location, allowing for the loading of files from several servers simultaneously. IPFS employs a peer-to-peer approach to simultaneously fetch files from numerous sources, enabling the following:

1) **Fingerprinting data:** Material identifiers (CIDs) provide a unique hash address to each item of material, data, or file, allowing to securely preserve it by "pinning" it.

2) **Storing and distributing content freely:** The peer-to-peer based network connects several nodes, enabling location of all nodes containing the desired material and assisting others in discovering the owned information.

3) **Optimizing content delivery:** IPFS utilizes local servers to cache and store data, resulting in more efficient distribution and delivery and reducing bandwidth use.

2.7. Electronic Medical Record (EMR)

Electronic Medical Records (EMRs) [17] refer to a collection of both organized and unorganized data points that are maintained digitally inside healthcare provider offices or hospital facilities. Once curated, this data is an excellent source of information for constructing a patient chronology that includes the medical and treatment histories of patients. It is an electronic rendition of the physical charts found in the clinician's or hospital office.

3. Proposed Framework

To enhance understanding of the framework's diverse components, we introduce them gradually. Initially, our focus is solely on integrating Hyperledger Indy, where we delve into a thorough examination of its characteristics. Following

this, we seamlessly incorporate additional technologies, including Hyperledger Fabric [13] and IPFS [16], and engage in a comprehensive discussion of their respective benefits.

We start by introducing some key actors of the healthcare systems:

1) **Patients**:

The overarching goal of the framework is to empower patients with full control over their Electronic Medical Record (EMR). Access to their records should be limited to authorized individuals and is granted only when necessary. Simultaneously, patients should have the capability to remotely prove the ownership of their EMRs.

2) **Healthcare Attribute Providers (HAP)**:

These entities encompass any attribute providers such as laboratories and Internet of Medical Things (IoMT) devices, each of which is regarded as a trusted entity from the perspective of attribute consumers. It is essential for each entity to have its own public key duly registered in Hyperledger Indy by a trusted organization. For instance, the healthcare ministry might register the public key of laboratories in Indy. Additionally, alongside public keys, the registration process can include the authorization of attributes that any producer (IoMT, Laboratory, etc.) is permitted to generate.

3) **Healthcare Attribute Consumers (HAC)**:

An attribute consumer pertains to any stakeholder in the health system, including doctors or nurses, who seeks specific attributes related to a patient. Their primary concern lies in verifying the authenticity of attribute providers and confirming the ownership of patients.

3.1. EMRs Management Based on Indy

Traditional certification techniques, such as Certificate Authorities (CA) and Public Key Infrastructure (PKI), have long been pivotal in upholding online security. However, they are no longer appropriate or sufficient for various contemporary application domains, particularly when addressing the intricacies of digital identities, including Electronic Medical Records (EMRs). Recognizing the evolving landscape, it is important to discern the constraints of conventional methods and explore alternatives offered by emerging technologies. For example, Public Key Infrastructure (PKI) presents specific limitations that can be effectively addressed by the Hyperledger Indy. Some of these limitations include:

- **Revocation:** It is not immediate in PKI and necessitates a delay.
- **Correlation:** Two HACs can collaborate to identify patients who have visited both entities. Moreover, when a patient provides two distinct EMRs to dishonest HACs, they can share them.
- **Selective Disclosure:** Patients have the flexibility to withhold certain attributes from HACs. In fact, in some instances, only specific sections of the Electronic Medical Record (EMR) may be required for disclosure.
- **Zero-Knowledge Proof:** Sometimes, patients may need to demonstrate that the value of an attribute (e.g. age) respects a specified property (e.g. age $\geq$ 18)

without divulging its exact value.

- **Key Rotation:** Given the potential compromise of a private key, it is advisable to periodically change keys as a proactive measure to mitigate associated risks.

As an emerging technology, Hyperledger Indy has been introduced to collectively address these limitations, offering a self-sovereign solution tailored for digital identities. Additionally, being a blockchain-based solution, it inherently delivers heightened availability, traceability, transparency, decentralization, distribution, and immutability when compared to classical database systems combined with traditional PKI.

Hyperledger Indy:

Let's delve into a tangible example to showcase the practical application of Hyperledger Indy in EMR management. An overview of the proposed architecture is illustrated by Figure 3 and further detailed in the following paragraphs.

1) **Initialization step:** This phase is detailed by steps *I*.1 to *I*.3 of Table 3.

In step *I*.1, the patient engages with the Healthcare Attributes Provider (HAP) to measure the values of specific attributes $a_1,\cdots,a_n$. Additionally, the patient provides a public key k_0, which will serve as the means for authenticating himself in subsequent interactions.

In the patient's wallet, the following information is securely stored: *HAP*, $a_1,\cdots,a_n$, k_0, and $\left\{k_0^{-1}\right\}_{pwd}$, as outlined in step *I*.2 of Table 3. This record empowers the patient to identify the HAP to which he provides attributes $a_1,\cdots,a_n$, along with the associated authentication key k_0. The private key k_0 corresponding to k_0 is also preserved in the wallet, safeguarded by a password *pwd*, which can be extracted from certain biometric features.

The attributes' names, values, and their corresponding sampling times are linked to k_0 and stored in a local database of the HAP, safeguarded by a key k, as shown by step *I*.3. The key k may be derived from a password or, alternatively, or a random value protected by the public key of the HAP.

It's noteworthy that the patient is not required to disclose personally identifiable information (such as full name, social security number, address, etc.) and can maintain anonymity when interacting with both the HAP and the HAC.

Subsequently, the tuples $(a_1,v_1,T_1),\cdots,(a_n,v_n,T_n)$ will be referred to as the Electronic Medical Record (EMR).

2) **Patient requests an EMR:** This phase is detailed by steps *D*.1 to *D*.6 of Table 4.

At any point following the initialization step, the patient can remotely request specific attributes managed by a HAP. As illustrated in step *D*.1 of Table 4, the patient supplies their public key k_0, used for authentication, an additional a *DID* and its key k_b. The *DID* will be linked to the requested EMR through the HAP signature.

In step *D*.2 of Table 4, the HAP prompts the patient to prove ownership of k_0 and k_b by issuing a challenge N_P. In response, the patient signs $H(k_0,N_P)$ using

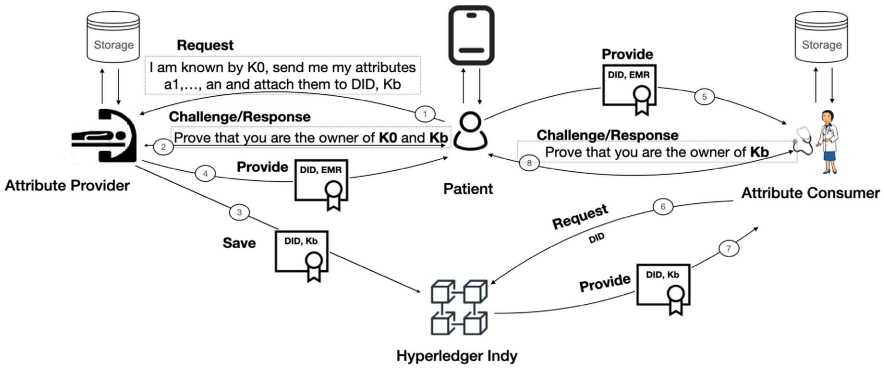

Figure 3. EMRs management based on Indy.

Table 3. Patient registers his attributes with HAP.

Patient provides to HAP his attributes $a_1,\cdots,a_n$ and attach them to k_0				
I.1	Patient	$\rightarrow$	HAP	: $k_0,(a_1,v_1),\cdots,(a_n,v_n)$
Patient saves in his wallet the necessary information allowing him to request the values of attributes $a_1,\cdots,a_n$				
I.2	Patient	$\rightarrow$	Wallet: *HAP*, $a_1,\cdots,a_n$, k_0; $\{k_0^{-1}\}_{pwd}$	
HAP saves in his database collected EMR and attaches it to k_0				
I.3	HAP	$\rightarrow$	HAP-DB	: k_0, $\underbrace{\{(a_1,\upsilon_1,T_1),\cdots,(a_n,\upsilon_n,T_n)\}k}_{EMR}$, $\{k\}_{k_c}$

Table 4. Indy-EMR: Patient requests an EMR from a HAP.

The patient furnishes his identifier k_0 to the HAP and requests an EMR containing $a_1,\cdots,a_n$ and linked to *DID* and k_b				
D.1	*Patient*	$\rightarrow$	*HAP*	: k_0, *DID*, k_b, $a_1,\cdots,a_n$
HAP challenges the patient to prove that he is the owner of k_0 and k_b				
D.2	*HAP*	$\rightarrow$	*Patient*	: k_0, N_p
D.3	*Patient*	$\rightarrow$	*HAP*	: $\{H(k_0,N_p)\}_{k_0^{-1}},\{H(k_b,N_p)\}_{k_b^{-1}}$
HAP links *DID* to k_b and stores the result in Indy				
D.4	*HAP*	$\rightarrow$	*Indy*	: *DID*, k_b, $\{H(DID,k_b)\}_{k_p^{-1}}$
HAP provides the requested *EMR*, protected by k, to the patient				
D.5	*HAP*	$\rightarrow$	*Patient*	: $\underbrace{DID,\{EMR\}_k}_{M}$, $\{H(M)\}_{k_p^{-1}}$, $\{k\}_{k_0}$
Patient stores the requested *EMR*, protected by k, in his wallet				
D.6	*Patient*	$\rightarrow$	Wallet:	$\underbrace{DID,\{EMR\}_k}_{M}$, $\{H(M)\}_{k_p^{-1}}$, $\{k\}_{k_0}$

the private key associated with k_0, signs $H(k_b,N_P)$ using the private key linked to k_b, and subsequently returns the concatenation of these results to the HAP in step *D*.3.

Once the patient is successfully authenticated, the HAP proceeds to store, at step *D*.7 of Table 4, the association between *DID* and k_b in the Indy ledger. Sub-

sequently, in step *D*.5 of Table 4, the HAP transmits the requested *EMR*, secured by a randomly generated key *k*, itself encrypted by the patient's public key. Finally, at step *D*.6 of Table 4, the patient saves the received encrypted *EMR* in their wallet.

3) **Patient provides attributes:** This phase is detailed by steps *P*.1 to *P*.5 of Table 5.

In step *P*.1 of Table 5, the patient retrieves his certified EMR from his wallet and encrypts it with a fresh random key *k*. This key *k* is itself encrypted by the public key k_c of the HAC and appended to the transmitted message.

In step *P*.2 of Table 5, the HAC extracts the *DID* from the received message and queries Indy to locate its associated k_b in step *P*.3. Subsequently, the HAC prompts the patient to validate his ownership of k_b by transmitting the *DID* and a nonce N_c in step *P*.4 of Table 5. Finally, during the step *P*.5 of Table 5, the patient signs the hash of the concatenation of the *DID* and N_c using the private key linked to k_b and returns the result to the challenge of the HAP.

The straightforward architecture depicted in Figure 3 brings forth a multitude of advantages resumed in Table 6. Patients wield complete control over their Electronic Medical Records (EMRs) through their wallets. Anonymity is preserved, as patients are not obligated to divulge their names to access healthcare services. The Zero-Knowledge Proof (ZKP) techniques offered by Indy empower patients to validate specific properties related to their attributes without divulging the actual details (e.g., proving that the patient's blood sugar level falls between 0.70 g/l and 1.10 g/l). Moreover, the architecture enables remote healthcare services, allowing the remote verification of EMR ownership. However, the availability, integrity, and confidentiality of EMRs are contingent on the protection measures implemented on different actors' ends.

Finally, it's noteworthy that Indy also offers the revocation of DIDs in the event of a lost or compromised wallet.

3.2. WebAuthn

In preceding architecture of Figure 3, involved private keys are securely stored in wallets of actors (Patients, HACs and HAPs) in an encrypted format, with encryption keys derived from biometric features or a user-generated password. This encryption ensures that even if a hacker gains access to the wallet's contents, decrypting the private key remains hardly attainable. Nonetheless, this protection can be jeopardized if a hacker installs keylogger (a malware that intercept any keystroke) on the user's smart device (smartphone, computer, tablet, etc.) to record its activities looking for the most vulnerable moment when the user enters the wallet password to capture it, enabling theft of the remaining private data contained in the wallet.

An effective mitigation technique consists the wallet in a special cryptographic USB key (compatible with WebAuthn or Fido2) that also has the capacity of signing token such as a challenge or a key. This USB key is itself protected against lost or theft by built in fingerprint sensor that is used to unlock it. In

Table 5. Indy-EMR: Patient provides his attributes to HAC.

Patient provides a protected EMR attached to *DID* to *HAC*				
P.1	*Patient*	→	*HAC*	: $\underbrace{DID, \{EMR\}_k}_{M}$, $\{H(M)\}_{k_p^{-1}}$, $\{k\}_{k_c}$
HAC looks for k_b linked *DID* in *Indy*				
P.2	*HAC*	→	*Indy*	: *DID*
P.3	*Indy*	→	*HAC*	: *DID*, k_b, $\{H(DID, k_b)\}_{k_p^{-1}}$
HAC challenges *Patient* to prove that he is the owner of k_b				
P.4	*HAC*	→	*Patient*	: *DID*, N_c
P.5	*Patient*	→	*HAC*	: $\{H(DID, N_c)\}_{k_b^{-1}}$

Table 6. Features of the solution based on Indy.

Features / Approach	Patient Controls Access to his EMRs	Anonymous Healthcare Service	EMR-Availability	EMR-Confidentiality	EMR-Integrity	EMR-Access Control Delegation	EMR-Zero-Knowledge Proof	EMR-Selective Disclosure	EMR-Anti-Correlation	EMR-Ownership Proof	EMR-Revocation
Indy EMRs	✓	✓	++	++	++	✓	✓	✓	✓	✓	✓

- means that the proposed approach has the corresponding feature.
- An empty case means that the proposed approach hasn't the corresponding feature.
- +++: high ++: medium +: low.

other words, the signature of any token by a private key in the wallet cannot happen only after unlocking the USB key by providing the owner's fingerprint.

Hereafter, we provide additional insights into incorporating a WebAuthn USB key into the architecture depicted in Figure 3.

1) **Initialization step:** This phase is elaborated in steps *I*.1 to *I*.5 as illustrated in Table 7.

In step *I*.1, the patient interacts with their cryptographic USB to generate a pair of public/private keys (k_0 and k_0^{-1}), which locally stores the tuple (*HAP*, k_0, k_0^{-1}), and returns k_0 to the patient. The remainder of the protocol follows the process outlined in Figure 3, except that the patient is relieved from storing any private key in his wallet.

2) **Patient requests for attributes:** This phase is detailed by steps *D*.1 to *D*.12 of Table 8.

The primary modification from the earlier protocol depicted in Table 4 is that whenever the patient requires a new pair of keys (e.g., in step *D*.1), they engage

Table 7. WebAuthn: Patient register his attributes with HAP.

Patient asks his cryptographic *USB* to create a key pair and attaches it to *HAP*				
I.1	*Patient*	→	*USB—Patient*	: *HAP*
I.2	*USB*	→	*Patient*	: k_0
Patient provides the values of his attributes $a_1,\cdots,a_n$ to *HAP* and links them to k_0				
I.3	Patient	→	*HAP*	: k_0, $(a_1,\upsilon_1),\cdots,(a_n,\upsilon_n)$
Patient saves in his *Wallet* the provider, *HAP*, of his attributes $a_1,\cdots,a_n$ and his identification key k_0				
I.4	*Patient*	→	*Wallet*	: *HAP*, $a_1,\cdots,a_n$, k_0
HAP stores, in his database, the attributes $a_1,\cdots,a_n$, their values, their collection times and their identification key k_0				
I.5	*HAP*	→	*HAP—DB*	: k_0; $\underbrace{\{(a_1,\upsilon_1,T_1),\cdots,(a_n,\upsilon_n,T_n)\}k}_{EMR}$

Table 8. Indy-EMR: Patient requests his attributes from HAP.

Patient generates new pair of asymmetric keys				
D.1	*Patient*	→	*USB—Patient*	: *DID*
D.2	*USB—Patient*	→	*Patient*	: k_b
Patient requests attributes for patient identified by k_0				
D.3	*Patient*	→	*HAP*	: k_0, *DID*, k_b, $a_1,\cdots,a_n$
Patient proves his ownership of public keys k_0 and k_b via a challenge/response				
D.4	*HAP*	→	*Patient*	: k_0, N_p
D.5	*Patient*	→	*USB—Patient*	: *HAP*, $H(k_0, N_p)$
D.6	*USB—Patient*	→	*Patient*	: $\{H(k_0,N_p)\}_{k_0^{-1}}$
D.7	*Patient*	→	*USB—Patient*	: $DID, H(k_b,N_p)$
D.8	*USB*—Patient	→	*Patient*	: $\{H(k_b,N_p)\}_{k_b^{-1}}$
D.9	*Patient*	→	*HAP*	: $\{H(k_0,N_p)\}_{k_0^{-1}}, \{H(k_b,N_p)\}_{k_b^{-1}}$
HAP saves the link between DID and kb in Indy				
D.10	HAP	→	*Indy*	: *DID*, k_b, $\{H(DID,k_b)\}_{k_p^{-1}}$
HAP sends the EMR to patient in a protected form				
D.11	HAP	→	*Patient*	: $\underbrace{DID,\{EMR\}_k}_{M}, \{H(M)\}_{k_p^{-1}}, \{k\}_{k_0}$
Patient saves the EMR in his wallet				
D.12	*Patient*	→	*Wallet*	: $\underbrace{DID,\{EMR\}_k}_{M}, \{H(M)\}_{k_p^{-1}}, \{k\}_{k_0}$

with their USB. Similarly, when a participant needs to sign a message (as in steps *D*.5 and *D*.7), they interact with their USB. In these interactions, the participant

provides the identifier associated with the key and the message to sign, subsequently receiving the outcome of the signature.

3) **Patient provides attributes:** This phase is as detailed by steps *P*.1 to *P*.7 of Table 9.

The EMR is stored in the patient's wallet, encrypted by a random symmetric key k, which is itself protected by the patient's private key. To transmit this EMR to a HAC, the key k needs to be secured by the public key k_c of the HAC, as shown in step *P*.1 of Table 9. However, if the patient decrypts $\{k\}_{k_0}$ and encrypts it with k_c while his wallet is compromised by a keylogger malware, the attacker can expose k, subsequently revealing the content of the wallet. To address this vulnerability, a commutative encryption system like RSA can be employed. In such a system, $\{k\}_{k_c} = \left\{\left\{\{k\}_{k_0}\right\}_{k_0^{-1}}\right\}_{k_c} = \left\{\left\{\{k\}_{k_0}\right\}_{k_c}\right\}_{k_0^{-1}}$. The remaining steps of the protocol mirror those in Table 5; whenever a signature is required (as in *P*.5), the cryptographic USB is invoked.

WebAuthn delivers robust protection for private keys and signatures, enhancing properties such as confidentiality, authenticity, and support for remote healthcare services that rely on proof of ownership through private keys. The additional benefits of WebAuthn are encapsulated in Table 10.

3.3. IPFS

As a distributed and decentralized file storage and sharing system, IPFS offers substantial advantages for applications such as Electronic Medical Record Management (EMRM), effectively addressing some of the limitations and challenges associated with traditional databases. Notably, since IPFS does not suffer from a central point of failure, and files stored redundantly at different levels, it significantly enhances availability. Additionally, files in IPFS are referenced using their hash values, ensuring inherent integrity; any alteration produces a hash different from its address, facilitating the detection of changes. Furthermore, IPFS supports versioning, as changes to a file result in a new hash, the updated version is stored at a new address while retaining access to the previous version. Table 11 summarizes the benefits of IPFS.

Hereafter, we highlight the modifications involved by the integration of IPFS into the existing architecture.

1) **Initialization step:** It will mirror Table 7, with the distinction being that EMRs will now be stored in IPFS, as depicted by Table 12.

2) **Patient requests for attributes:** This phase is detailed through steps *D*.1 to *D*.19 as outlined in Table 13.

The main modification from the previous protocol in Table 8 is that whenever the patient requires a new pair of keys (e.g., in step *D*.1) or needs to sign a message (as in steps *D*.5, *D*.7, and *D*.10), he engages his cryptographic USB. He supplies the identifier associated with the key and the message to sign, subsequently receiving the outcome of the signature.

Table 9. Indy-EMR: Patient provides his attributes to HAC.

Patient provides an *EMR* linked to *DID* to *HAC*				
P.1	*Patient*	→	*HAC*	: $\underbrace{DID, \{EMR\}_k}_{M}, \{H(M)\}_{k_p^{-1}}, \{k\}_{k_c}$
HAC retrieves the key k_b associated to *DID* from Indy				
P.2	*HAC*	→	*Indy*	: *DID*
P.3	*Indy*	→	*HAC*	: *DID*, k_b, $\{H(DID, k_b)\}_{k_p^{-1}}$
Patient proves ownership of k_b via challenge/response				
P.4	*HAC*	→	*Patient*	: *DID*, N_c
P.5	*Patient*	→	*Patient—USB*	: *H*(*DID*, N_c)
P.6	*Patient—USB*	→	*Patient*	: $\{H(DID, N_c)\}_{k_b^{-1}}$
P.7	*Patient*	→	*HAC*	: $\{H(DID, N_c)\}_{k_b^{-1}}$

Table 10. WebAuthn contributions.

Features \ Approach	EMR-Confidentiality	EMR-Authenticity	Remote Healthcare Service
WebAuthn	+++	+++	+++

+++: high ++: medium +: low.

Table 11. IPFS contributions.

Features \ Approach	EMR-Integrity	Availability	Versioning
IPFS	+++	+++	+++

+++: high ++: medium +: low.

Table 12. WebAuthn-IPFS: Patient registers his attributes with HAP.

Patient asks his cryptographic *USB* to create a key pair and attaches it to HAP				
I.1	*Patient*	→	*USB—Patient*	: *HAP*
I.2	*USB—Patient*	→	*Patient*	: k_0
Patient provides his attributes to HAB and links them to k_0				
I.3	*Patient*	→	*HAP*	: k_0, $(a_1, \upsilon_1), \cdots, (a_n, \upsilon_n)$
Patient saves in his Wallet information allowing to request his provides attributes				
I.4	*Patient*	→	*Wallet*	: *HAP*, $a_1, \cdots, a_n$, k_0
HAP protects and saves the values of collected attributes and their collection times in IPFS				

Continued

I.5	*HAP*	→	*IPFS*	: $\underbrace{\{(a_1,\upsilon_1,T_1),\cdots,(a_n,\upsilon_n,T_n)\}_k}_{F}$
HAP saves in his database the location of attributes attached to k_0				
I.6	*HAP*	→	*HAP—DB*	: k_0, $H(F)$, $\{k\}_{k_c}$

Table 13. Indy-WebAuthn-IPFS: Patient requests his attributes from HAP.

Patient asks his cryptographic USB to create a key pair and attaches it to DID				
D.1	*Patient*	→	*USB—Patient*	: *DID*
D.2	*USB—Patient*	→	*Patient*	: k_b
Patient sends his identifier k_0 to *HAP* requests his attributes $a_1,\cdots,a_n$ and asks him to link them do *DID* and k_b				
D.3	*Patient*	→	*HAP*	: k_0, *DID*, k_b, $a_1,\cdots,a_n$
HAP challenges patient to prove that he is the owner of k_0 and k_b				
D.4	*HAP*	→	*Patient*	: k_0, N_P
D.5	*Patient*	→	*USB—Patient*	: *HAP*, $H(k_0, N_P)$
D.6	*USB—Patient*	→	*Patient*	: $\{H(k_0,N_p)\}_{k_0^{-1}}$
D.7	*Patient*	→	*USB—Patient*	: *DID*, $H(k_b, N_P)$
D.8	*USB—Patient*	→	*Patient*	: $\{H(k_b,N_p)\}_{k_b^{-1}}$
D.9	*Patient*	→	*HAP*	: $\{H(k_0,N_p)\}_{k_0^{-1}},\{H(k_b,N_p)\}_{k_b^{-1}}$
HAP asks his *USB* to sign the link between *DID* and kb and stores the result in Indy				
D.10	*HAP*	→	*USB—HAP*	: *HAP*, $H(DID, k_b)$
D.11	*USB—HAP*	→	*HAP*	: $\{H(DID,k_b)\}_{k_p^{-1}}$
D.12	*HAP*	→	*Indy*	: *DID*, k_b, $\{H(DID,k_b)\}_{k_p^{-1}}$
HAP looks for the address of the *EMR* in his database and requests it from *IPFS*				
D.13	*HAP*	→	*HAP—DB*	: k_0
D.14	*HAP—DB*	→	*HAP*	: k_0, $H(F)$, $\{k\}_{k_p}$
D.15	*HAP*	→	*IPFS*	: $H(F)$
D.16	*IPFS*	→	*HAP*	: $\{EMR\}_k$
HAP attaches *EMR* to *DID* and saves it in *IPFS*				
D.17	*HAP*	→	*IPFS*	: $\underbrace{\underbrace{DID,\{EMR\}_k}_{M},\{H(M)\}_{k_p^{-1}}}_{F_1}$
HAP sends to Patient the *IPFS* address of the requested *EMR* with its protection key				
D.18	*HAP*	→	*Patient*	: $H(F_1)$, $\{k\}_{k_0}$
Patient saves the name of attributes, the address of their *EMR* and their protection key in his Wallet				
D.19	*Patient*	→	*Wallet*	: $a_1,\cdots,a_n$, $H(F_1)$, $\{k\}_{k_0}$

3) **Patient provides attributes:** This phase aligns with the process outlined in Table 9, with an important divergence: the EMR is now stored in IPFS rather than the patient's wallet. The patient provides the HAC with the EMR's address and the associated protection key, as illustrated in Table 14.

Table 14. Indy-WebAuthn-IPFS: Patient provides his attributes to HAC.

Patient provides to *HAC* the address, $H(F_1)$, of his *EMR* in *IPFS* as well as the key allowing to decrypt it				
P.1	*Patient*	→	*HAC*	: $H(F_1)$; $\{k\}_{k_c}$
HAC retrieves *EMR* from *IPFS* and reads its *DID*				
P.2	*HAC*	→	*IPFS*	: $H(F_1)$
P.3	*IPFS*	→	*HAC*	: $\underbrace{\underbrace{DID, \{EMR\}_k}_{M}, \{H(M)\}_{k_p^{-1}}}_{F_1}$
HAC retrieves the key k_b associated to *DID* from Indy				
P.4	*HAC*	→	*Indy*	: *DID*
P.5	*Indy*	→	*HAC*	: *DID*, k_b, $\{H(DID, k_b)\}_{k_p^{-1}}$
HAC challenges Patient to prove that he is the owner of k_b				
P.6	*HAC*	→	*Patient*	: *DID*, N_c
P.7	*Patient*	→	*Patient—USB*	: $H(DID, N_c)$
P.8	*Patient—USB*	→	*Patient*	: $\{H(DID, N_c)\}_{k_b^{-1}}$
P.9	*Patient*	→	*HAC*	: $\{H(DID, N_c)\}_{k_b^{-1}}$

An abstracted version of the architecture using Indy, WebAuthn and IPFS is given by Figure 4.

3.4. IoT

Wearable healthcare IoT devices offer several advantages that significantly enhance patient healthcare, including continuous and remote monitoring, health data collection, early detection of health problems, fall detection, emergency response, and more. Consequently, an Electronic Medical Record Management (EMRM) solution should possess the capability to seamlessly integrate these devices.

The existing architectures can be effortlessly enriched with IoT by considering it as a micro-HAP. However, some IoT devices may lack the capability to connect to the internet and save data in platforms like IPFS. In such cases, a hub can bridge this gap, and the device (smartphone, tablet, etc.) containing the patient's wallet may assume this role.

Importantly, from a protocol standpoint, no modifications are required to integrate IoT devices.

3.5. Smart Contract

Smart contracts offer enhancements in both automation and security, providing benefits such as improved availability, immutability, transparency, and traceability.

As illustrated in Figure 5, blockchains that support smart contracts, like Fabric or Ethereum, can effectively manage access control. All permissions, encompassing Role-Based Access Control (RBAC) or Attribute-Based Access

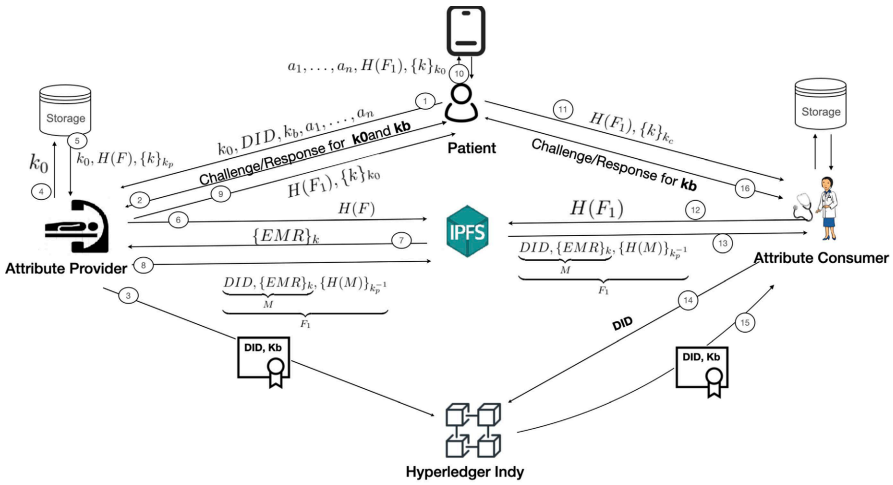

Figure 4. Indy-WebAuthn-IPFS architecture.

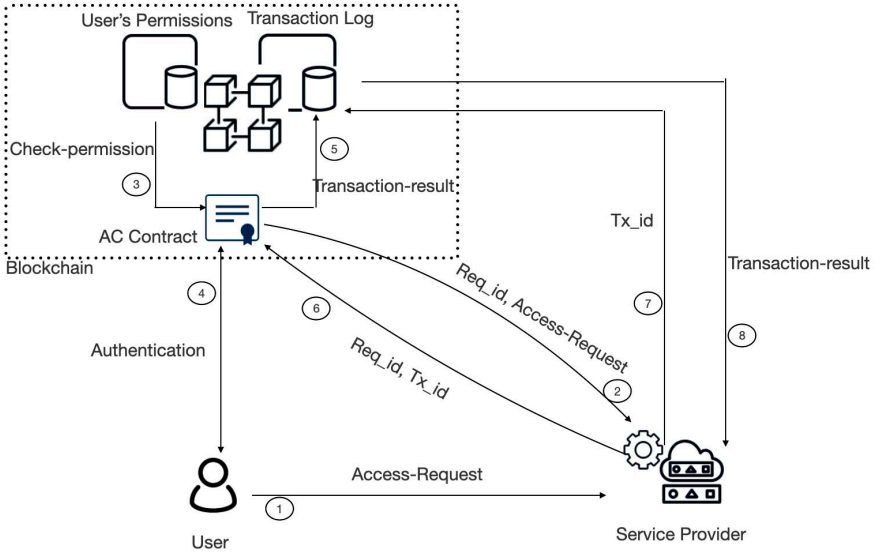

Figure 5. Access control using smart contract.

Control (ABAC) rules, are stored within the smart contract. When service providers receive access requests (subject, object, access), they forward them to the Access Control (AC) smart contract. The smart contract validates whether the request is authorized and authenticates the subject, determining whether access should be granted. Upon a positive decision, a transaction is recorded in the blockchain as proof of acceptance. The transaction identifier (Tx_id) is then communicated to the service provider which grants or denies access accordingly.

Additional insights into the progression of these steps are provided in Table 15. In step *S*.1, the user sends a signed request incorporating the subject (S), object (O), requested access (A), and a nonce (N_s) serving as the request identifier. Moving to step *S*.2, the service provider appends its own identifier (N_p) to the request and submits it to the Access Control (AC) smart contract. The smart contract scrutinizes the user's permissions rules, saved in the blockchain, to

Table 15. Automation of access control using smart contract.

S.1	*Subejct*	→	*Service—Provider*: N_s;	$\underbrace{(S,O,A)}_{M},\{H(M)\}_{k_s^{-1}}$
S.2	*Service—Provider*	→	*AC—Contract*	: N_p, N_s; $\underbrace{(S,O,A)}_{M},\{H(M)\}_{k_s^{-1}}$
S.3	*AC—Contract*	→	*Subject*	: N_c, N_s
S.4	*Subject*	→	AC—*Contract*	: $\underbrace{N_c,N_s}_{N},\{H(N)\}_{k_s^{-1}}$
S.5	*AC—Contract*	→	*Blockchain*	: Tx_{id}, N_p, (S, O, A)
S.6	*AC—Contract*	→	Service—*Provider*: Tx_{id}	
S.7	*Service—Provider*	→	*Blockchain*	: Tx_{id}, N_p, (S, O, A)

determine if the requested access is authorized. If affirmative, a challenge-response protocol unfolds with the subject in steps *S*.3 and *S*.4. Upon successful authentication, a transaction (Tx_{id}, N_p, (S,O,A)) is stored in the blockchain at step *S*.5. Finally, at step *S*.6, the transaction identifier is communicated to the service provider, enabling them to verify the result and make decisions accordingly.

3.6. Global Architecture

Figure 6 summarize the proposed architectural design that adeptly integrates numerous emerging technologies pivotal for effective EMR (Electronic Medical Records) management. The figure not only showcases the intricate interplay between various components but also underscores their pivotal role in revolutionizing EMR systems.

4. Literature Review

In this section, we review existing research on sharing EMR systems. The surveyed approaches are classified based on blockchain types such as Hyperledger Indy, Hyperledger Fabric, Ethereum Blockchain, and Consortium Blockchain.

4.1. Hyperledger Indy-Based Approaches

The Healthblock architecture was designed by Abdelgalil and Mejri [11] with the aim of facilitating the secure exchange of Electronic Medical Records (EMRs) while ensuring the preservation of patients' privacy. In order to achieve these objectives, a range of technologies have been integrated. IPFS is used to secure and disseminate EMRs in a decentralized off-chain storage system, ensuring the enduring existence of these documents. Hyperledger Indy, on the other hand, empowers patients with full authority over their EMRs. Lastly, Hyperledger Fabric is responsible for managing patient-access control rules and delegates.

4.2. Hyperledger Fabric-Based Approaches

O. Attia *et al.* [18] propose a framework based on Blockchain technology to

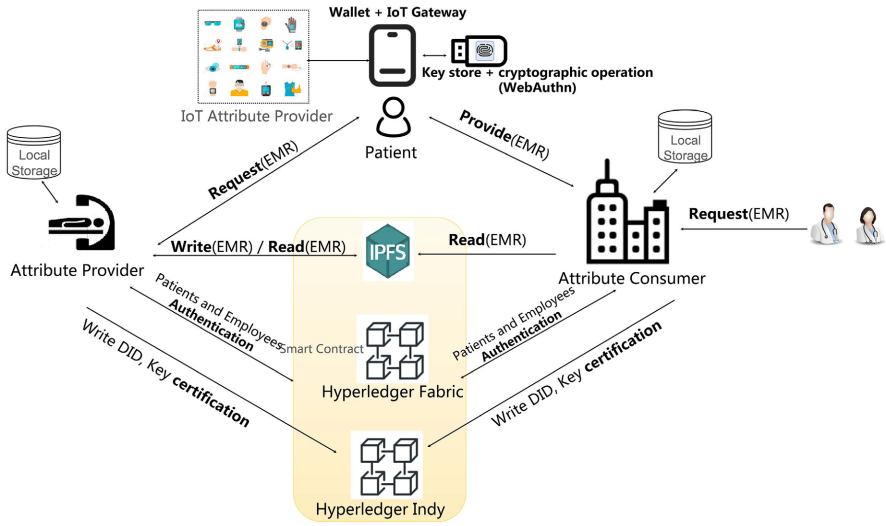

Figure 6. Global architecture.

enhance the security of healthcare applications. Using the Hyperledger Fabric, the proposed framework enables distributed secure access to all device data. The most significant shortcoming of this strategy is that it only solves a subset of IoT security concerns and does not consider IoT attacks. Abdelgalil and Mejri [11] used Hyperledger Fabric to propose Healthblock framework. Abdelgalil and Mejri [11] utilize IPFS in contrast to O. Attia *et al.* [18].

4.3. Ethereum Blockchain-Based Approaches

Sun *et al.* [19] introduce a technique to effectively and securely archive electronic medical records. The researchers utilize attribute-based encryption methodologies in conjunction with blockchain technology to regulate access to digital health information. Electronic Medical Records are securely encrypted and stored within a distributed IPFS database. This approach ensures both privacy and scalability, since it avoids reliance on a single point of failure. Nguyen *et al.* [20] introduce a novel decentralized health framework that combines mobile edge computing and blockchain technology to facilitate data offloading and data sharing in dispersed hospital networks. They used a decentralized authentication system linked to a distributed IPFS store to enhance the quality of their service. Akkaoui *et al.* [21] introduce a safe health data sharing architecture called "EdgeMediChain" that utilizes blockchain technology in edge computing. The framework addresses the essential needs for scalability, security, and privacy in the medical ecosystem. A. Dwivedi *et al.* [22] present a hybrid strategy that ensures both data security and privacy. The presented system is commonly referred to as hybrid due to its integration of blockchain technology alongside the use of both private and public keys, as well as the incorporation of advanced cryptographic functionalities. One of the drawbacks associated with this proposed approach is the restricted resources of blockchain technology, rendering it unsuitable for a significant number of IoT devices. There were other concerns pertain-

ing to the expenses associated with blockchain technology, such the increased need for bandwidth and computational resources. Buccafurri *et al.* [23] present a method to integrate smart contracts and blockchain using the Attribute-Based Encryption (ABE) scheme. While this technique successfully implements access control and meet the objectives for service delivery with accountability, it is not compatible with systems that have computational limitations.

Other contributions such as Sun *et al.* [19], Nguyen *et al.* [20], Akkaoui *et al.* [21], Dwivedi *et al.* [22], and Buccafurri *et al.* [23] have used the Ethereum Blockchain to provide their respective solutions. Furthermore, Sun *et al.* [19], Nguyen *et al.* [20], and Akkaoui *et al.* [21] all used IPFS in their suggested solutions, however, Dwivedi *et al.* [22] and Buccafurri *et al.* [23] did not utilize IPFS. As for access control management, Sun *et al.* [19] and Buccafurri *et al.* [23] employ ABE, Nguyen *et al.* [20] utilize a smart contract-based, Akkaoui *et al.* [21] utilize RBAC, and Dwivedi *et al.* [22] rely on hashing encryption.

4.4. Consortium Blockchain-Based Approaches

Zhang and Wang [24] propose a consortium blockchain to facilitate the exchange of medical data. To establish access control, the implementation of attribute-based access control approach is used. This technique involves patients setting unique access rules for their medical records based on attributes, while record requesters are identified by a set of characteristics. Lin *et al.* [25] suggest a cryptographic strategy using blockchain technology for secure and confidential data transmission in the e-healthcare system. The proposed approach ensures data secrecy by concealing the condition inside the re-encryption key, preventing the proxy from acquiring any knowledge of the condition. Both Zhang and Wang [24] and Lin *et al.* [25] have used the Consortium Blockchain to provide their respective solutions. It is worth noting that Zhang and Wang [24] use IPFS in their suggested solution and ABAC for access control, however Lin *et al.* [25] did not utilize IPFS and use CPRE for access control.

4.5. Other Approaches

Nie *et al.* [26] introduce an innovative data-sharing system that utilizes blockchain technology to enable safe and privacy-preserving profile matching. A bloom filter, using hash functions, is intended to authenticate the legitimacy of keyword ciphertext. The safe profile matching is accomplished by using the Key-policy attribute-based encryption (KPABE) method and smart contracts. Li *et al.* [27] propose a system to address the issues of medical data connectivity and resource sharing, enhance the efficiency and efficacy of illness diagnosis, mitigate the conflict between physicians and patients, and enable personal health management. Bai *et al.* [37] propose a method to integrate smart contracts with blockchain using the Attribute-Based Encryption (ABE) technique. While this technique successfully implements access control and meet accountability criteria for service delivery, it is not compatible with some systems due to computa-

tional limitations.

Kumar *et al.* [28] develop a new data-sharing framework called PBDL to address the security and privacy concerns associated with ongoing communication over public networks. The system is meant to be both safe and efficient. In their further research, they also suggested optimization strategies [29]. E. M. Abou-Nassar *et al.* [30] suggest an architecture for IoT healthcare that provides trustworthiness by using smart contracts to check and validate the integrity of other nodes in the system. The arrangement is divided into four separate tiers. The first stage involves gathering data, calibrating sensor readings, and using actuators. Data transmission in the second layer takes place by using gateways and interconnecting networks. The health boundary between the technology and application levels corresponds to the third layer. The last stratum inside the networking protocol stack is sometimes referred to as the application layer. Kumar *et al.* [31] suggest the use of decentralized off-chain storage as an efficient method to handle medical data related to patients. The medical data were kept on the InterPlanetary File System (IPFS), while the indexes were maintained on the blockchain. The suggested plan aims to ensure the coherence, soundness, and accessibility of medical data.

F. Jamil *et al.* [32] propose a blockchain-driven system specifically developed to monitor patient vital indicators via the use of smart contracts. The system in question utilizes Hyperledger Fabric, a distributed ledger platform specifically built for corporate applications based on blockchain technology. Furthermore, the system provides patients with other advantages, including a lasting documentation of their medical background and effortless retrieval of their medical data from any location worldwide. S. Chakraborty *et al.* [33] present a method that combines blockchain with IoT technology. To guarantee the security and reliability of data communicated via IoT devices inside the healthcare system, it is necessary to deploy suitable procedures. The use of blockchain technology facilitates the identification of individuals participating in the transactions. Furthermore, the healthcare industry has implemented the use of blockchain technology to ensure data security and privacy, as well as to provide healthcare professionals with accurate and comprehensive patient information. An inherent limitation of using this method is its need to be implemented specifically inside the confines of this academic publication.

Niu *et al.* [34] introduce a method that utilizes a permissioned blockchain framework in order to enhance the security of Electronic Medical Record (EMR) sharing. The system enables many users to search for information, guarantees data security by using attribute encryption using ciphertext, enforces precise access control, and effectively reduces the possibility of unauthorized doctors supplying false data by enforcing permitted access. In their study, Liu *et al.* [35] introduce BPDS, a system that utilizes blockchain technology to facilitate the exchange of electronic medical records (EMRs) while upholding privacy measures. In order to mitigate the risk of tampering with electronic medical records (EMRs), the primary EMRs are securely stored in cloud storage, with their ad-

ministration governed by smart contracts. Additionally, the indexes of these EMRs are saved inside a consortium blockchain. Finally, Dubovitskaya *et al.* [36] propose a blockchain-driven framework specifically developed to streamline the sharing of electronic medical records (EMRs) with individuals diagnosed with cancer. Permissioned blockchain features were used to provide immutable, responsible, and highly comprehensive access control for EMRs. Within this specific architectural framework, EMRs undergo encryption and are thereafter stored on a cloud server, using the public keys of the corresponding patients. To retrieve the data, one must have the decryption key owned by the data owner. It is worth mentioning that among these approaches, only Bai *et al.* [37] and Kumar *et al.* [31] employ IPFS in their proposed solutions.

4.6. Comparative Analysis

Table 16 presents a summary and comparison of existing work based on the technologies used. In particular, we compare existing work based on their use of the following technologies: Decentralized IoMT data, Hyperledger Indy, WebAuthn, Zero-Knowledge Proof for IoMT data, blockchain type, smart contract, blockchain stored data, Access control method, data Privacy, and IPFS technology.

We also evaluate the existing literature on sharing Electronic Medical Records with blockchain technology based on certain pertinent criteria as shown by Table 17:

- **Self-Sovereign EMRs:** Patients securely store their Electronic Medical Records (EMRs) in encrypted form, either inside their own wallets or in designated repositories. They exercise control over the access privileges of certain attributes within their EMRs, deciding who is authorized to read them and when such access is granted. In addition, Patients have full control over the configuration of Micro Laboratories; for example, patients configure Micro Laboratories to encrypt collected data and send it to specific frequencies or locations to save that data.
- **Anonymous Medical Assistance:** The healthcare system aims to provide medical assistance while maintaining the privacy and confidentiality of patients. The use of this strategy would be highly advantageous if it facilitates patients' ability to request anonymous assistance at any given moment. Patients diligently monitor a range of identity-related qualities, such as electronic medical records (EMRs), and exhibit just the essential information. For instance, individuals seeking a COVID-19 vaccination may be obliged to provide evidence of meeting certain criteria, such as surpassing a designated age threshold, possessing citizenship in the respective locality where the vaccine is sought, and confirming their prior non-receipt of the vaccine. In our architecture, patient can register with HAP and HAC using a random keys and DIDs making them anonymous.
- **EMR-Availability:** Ensuring constant and universal accessibility to EMRs, regardless of device damage or destruction, is of paramount importance.

Table 16. Electronic medical records with blockchain technology.

Authors	Decentralized IoMT Data	Hyperledger Indy	WebAuthn	ZKP for IoMT Data	Blockchain Type	Smart Contract	Blockchain Stored Data	Access Control Method	Data Privacy	IPFS
Li *et al.* [27]					Private and Consortium		Encrypted health data	AC policies	Encryption	No
Zhang *et al.* [24]					Consortium		Relevant metadata	ABAC	Asymmetric encryption	Yes
Lin *et al.* [25]							Encrypted index	CPRE		No
Attia *et al.* [18]					Fabric		Medical device data	Not specified		
Bai *et al.* [37]							Other information	Proxy re-encryption		
Abdelgalil and Mejri [11]					Indy and Fabric		Hash of EMRs and access control policy	ABAC	Encryption technology	Yes
Sun *et al.* [19]						Yes	Hash of EMRs	ABE		
Nguyen *et al.* [20]					Ethereum		Hash value is kept in the smart contract	Smart contract-based		
Akkaoui *et al.* [21]							Hash value	RBAC		
Dwivedi *et al.* [22]	No	No	No	No			Data hash	Hashing Encryption		
Buccafurri *et al.* [23]							Hash value	ABE		
A. Nassar *et al.* [30]							Data in Ripple chain	Not specified	Hash-based data storage	No
Jamil *et al.* [32]							Vital sign	ACL rules		
Nie *et al.* [26]							Keyword ciphertext	KPABE		
Liu *et al.* [35]							Reserve indexes of EMRs	Content extraction signature	Encryption technology	
Niu *et al.* [34]					Not specified		EMR hash values	ABE		
Dubovitskaya *et al.* [36]						No	Metadata and protocol for access control	RBAC		
Chakraborty *et al.* [33]							Data hash	Not specified	Not specified	
Kumar *et al.* [31]							Encrypted representation of medical records	Not specified	Data storage using hash functions	Yes
Proposed Solution	Yes	Yes	Yes	Yes	Hyperledgers Indy and Fabric	Yes	Hash of EMRs and secure access control of EMR collected from IoMT	ABAC	Encryption technology	Yes

Table 17. Comparative analysis.

Features / Approach	Self-Sovereign EMRs	Anonymous Medical Assistance	EMR-Availability	EMR-Confidentiality	EMR-Integrity	EMR-Access Control Delegation	EMR-Zero-Knowledge Proof	EMR-Selective Disclosure	EMR-Anti-Correlation	EMR-Ownership Proof	EMR-Traceability	EMR-Revocation
S. Chakraborty *et al.* [33]			+-	+-	+-	✓						
O. Attia *et al.* [18]			++	++	++	✓						
A. Dwivedi *et al.* [22]			++	++	++	✓						
E. M. Abou-Nassar *et al.* [30]			++	++	++	✓						
F. Jamil *et al.* [32]			++	++	++	✓						
Sun *et al.* [19]			+++	+++	+++	✓						
Buccafurri *et al.* [23]			+++	+++	+++	✓						
Liu *et al.* [35]			++	++	+++	✓						
Niu *et al.* [34]			++	++	+++	✓						
Dubovitskaya *et al.* [36]			+++	++	++	✓						
Kumar *et al.* [31]			+++	+++	+++	✓						
Kumar *et al.* [28]			++	++	++	✓						
Abdelgalil *et al.* [11]	✓	✓	+++	++	++	✓	✓	✓	++	++	++	+++
Buccafurri *et al.* [23]			++	++	++	✓						
Nie *et al.* [26]			++	++	++	✓						
Lin *et al.* [25]			++	++	++	✓						
Li *et al.* [27]			++	++	++	✓						
Bai *et al.* [37]			++	++	++	✓						
Zhang *et al.* [24]			++	++	++	✓						
Nguyen *et al.* [20]			++	++	++	✓						
Akkaoui *et al.* [21]			++	++	++	✓						
Proposed Framework	✓	✓	+++	+++	+++	✓	✓	✓	+++	+++	+++	+++

1) ✓ means that the proposed approach has the corresponding feature. 2) An empty case means that the proposed approach hasn't the corresponding feature. 3) +++: high ++: medium +: low.

- **EMR-Confidentiality:** EMRs contain very sensitive information that, if leaked, might result in major consequences for their owners, such as job loss and increased insurance costs.
- **EMR-Integrity:** It is imperative to guarantee that EMRs have not undergone unintended or intentional modifications. Without data integrity, EMRs are

rendered ineffective.

- **EMRs' Access Control:** Patients should have the capability to readily grant authorization for the disclosure of any medical information associated with their electronic medical records through the utilization of an EMR management system.
- **EMR-Zero-Knowledge Proof (ZKP):** Patients may benefit by displaying some characteristics associated with their personal traits while refraining from disclosing their individual beliefs. One such method of verification that individuals can employ is providing evidence of being above the legal age threshold of 18 years while refraining from explicitly revealing their specific age. In order to incorporate this particular functionality, several blockchains frequently employ Zero-Knowledge Proof (ZKP) techniques.
- **EMR-Selective Disclosure:** If an Electronic Medical Record (EMR) comprises numerous attributes, a patient may choose to disclose only a subset of them or specific details related to particular properties.
- **EMR-Anti-Correlation:** Various HACs entities should not have the capability to collaborate and unveil additional information about shared patients by correlating their respective data.
- **EMR-Traceability and Audit:** Having the ability to view all access and versions associated with an Electronic Medical Record (EMR) is valuable in diverse situations, such as gaining a deeper understanding of incidents when discrepancies occur.
- **EMR-Revocation:** Patients have the option to revoke access to their Electronic Medical Records (EMR) in the event of key loss or any other valid reason.

5. Conclusions

This paper illustrates the integration of emerging building blocks, including hyperledgers (Indy, Fabric, etc.), WebAuthn, IPFS, and smart contracts, to construct an Electronic Medical Records Manager equipped with compelling attributes such as zero-knowledge proof, anonymity, revocation, selective disclosure, remote ownership proof, and automation. Additionally, these building blocks significantly enhance classical security properties like confidentiality, availability, and integrity.

As part of future work, our intention is to implement this architecture and conduct a comprehensive analysis of its performance and scalability.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Blockchain in Retail Market Size & Share Analysis—Growth Trends & Forecasts

(2024-2029).
https://www.mordorintelligence.com/industry-reports/blockchain-in-retail-market

[2] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

[3] Niranjanamurthy, M., Nithya, B.N. and Jagannatha, S. (2019) Analysis of Blockchain Technology: Pros, Cons and SWOT. *Cluster Computing*, **22**, 14743-14757. https://doi.org/10.1007/s10586-018-2387-5

[4] Kaushik, A., Choudhary, A., Ektare, C., Thomas, D. and Akram, S. (2017) Blockchain-Literature Survey. *Proceedings of the* 2017 2*nd IEEE International Conference on Recent Trends in Electronics*, *Information and Communication Technology* (*RTEICT*), New York, 19-20 May 2017, 2145-2148. https://doi.org/10.1109/RTEICT.2017.8256979

[5] Jakobsson, M. and Juels, A. (1999) Proofs of Work and Bread Pudding Protocols (Extended Abstract). In: Preneel, B., Ed., *Secure Information Networks*, Springer, Boston, 258-272. https://doi.org/10.1007/978-0-387-35568-9_18

[6] Saleh, F. (2021) Blockchain without Waste: Proof-of-Stake. *The Review of Financial Studies*, **34**, 1156-1190. https://doi.org/10.1093/rfs/hhaa075

[7] Paul, P., Aithal, P.S., Saavedra, R. and Ghosh, S. (2021) Blockchain Technology and Its Types—A Short Review. *International Journal of Applied Science and Engineering*, **9**, 189-200. https://doi.org/10.30954/2322-0465.2.2021.7

[8] Indy. Hyperledger Indy. https://www.hyperledger.org/use/hyperledger-indy

[9] Sovrin Foundation. Sovrin. https://sovrin.org

[10] Sovrin Foundation. Use Case Spotlight: The Government of British Columbia Uses the SovrinNetwork to Take Strides towards a Fully Digital Economy. https://sovrin.org/usecase-spotlight-the-government-of-british-columbia-uses-the-sovrin-networkto- take-strides-towards-a-fully-digital-economy/

[11] Abdelgalil, L. and Mejri, M. (2023) HealthBlock: A Framework for a Collaborative Sharing of Electronic Health Records Based on Blockchain. *Future Internet*, **15**, Article No. 87. https://doi.org/10.3390/fi15030087

[12] Self-Sovereign Identity (SSI). https://sovrin.org/faq/what-is-self-sovereign-identity

[13] Linux Foundation. Hyperledgr Fabric. https://www.hyperledger.org/use/fabric

[14] Ream, J., Chu, Y. and Schatsky, D. (2016) Upgrading Blockchains: Smart Contract Use Cases in Industry. https://www2.deloitte.com/us/en/insights/focus/signals-for-strategists/using-blockchain-for-smart-contracts.html

[15] Szabo, N. (1997) The Idea of Smart Contracts. *Nick Szabo's Papers and Concise Tutorials*, **6**, 199.

[16] IPFS. IPFS Powers the Distributed Web. https://ipfs.io

[17] Kumar, R., Abrougui, K., Verma, R., Luna, J., Khattab, A. and Dahir, H. (2023) Chapter 12. Digital Twins for Decision Support System for Clinicians and Hospital to Reduce Error Rate. In: El Saddik, A., Ed., *Digital Twin for Healthcare*, Academic Press, Cambridge, 241-261. https://doi.org/10.1016/B978-0-32-399163-6.00017-2

[18] Attia, O. (2019) An IoT-Blockchain Architecture Based on Hyperledger Framework for Healthcare Monitoring Application. *Proceedings of the* 2019 10*th IFIP International Conference on New Technologies*, *Mobility and Security* (*NTMS*), Canary Islands, 24-26 June 2019, 1-5. https://doi.org/10.1109/NTMS.2019.8763849

[19] Sun, J., Yao, X., Wang, S. and Wu, Y. (2020) Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS. *IEEE Access*, **8**, 59389-

59401. https://doi.org/10.1109/ACCESS.2020.2982964

[20] Nguyen, D.C., Pathirana, P.N., Ding, M. and Seneviratne, A. (2021) BEdgeHealth: A Decentralized Architecture for Edge-Based IoMT Networks Using Blockchain. *IEEE Internet of Things Journal*, **8**, 11743-11757. https://doi.org/10.1109/JIOT.2021.3058953

[21] Akkaoui, R., Hei, X. and Cheng, W. (2020) EdgeMediChain: A Hybrid Edge Blockchain-Based Framework for Health Data Exchange. *IEEE Access*, **8**, Article ID: 113467. https://doi.org/10.1109/ACCESS.2020.3003575

[22] Dwivedi, A., Srivastava, G., Dhar, S. and Singh, R. (2019) A Decentralized Privacy-Preserving Healthcare Blockchain for IoT. *Sensors*, **19**, Article No. 326. https://doi.org/10.3390/s19020326

[23] Buccafurri, F., De Angelis, V., Lax, G., Musarella, L. and Russo, A. (2019) An Attribute-Based Privacy-Preserving Ethereum Solution for Service Delivery with Accountability Requirements. *Proceedings of the* 14*th International Conference on Availability, Reliability and Security*, Canterbury, 26-29 August 2019, 1-6. https://doi.org/10.1145/3339252.3339279

[24] Zhang, D., Wang, S., Zhang, Y., Zhang, Q. and Zhang, Y. (2022) A Secure and Privacy-Preserving Medical Data Sharing via Consortium Blockchain. *Security and Communication Networks*, **2022**, Article ID: 2759787. https://doi.org/10.1155/2022/2759787

[25] Lin, G., Wang, H., Wan, J., Zhang, L. and Huang, J. (2022) A Blockchain-Based Fine-Grained Data Sharing Scheme for e-Healthcare System. *Journal of Systems Architecture*, **132**, Article ID: 102731. https://doi.org/10.1016/j.sysarc.2022.102731

[26] Nie, X., Zhang, A., Chen, J., Qu, Y. and Yu, S. (2022) Blockchain-Empowered Secure and Privacy-Preserving Health Data Sharing in Edge-Based IoMT. *Security and Communication Networks*, **2022**, Article ID: 8293716. https://doi.org/10.1155/2022/8293716

[27] Li, C., Liu, J., Qian, G., Wang, Z. and Han, J. (2022) Double Chain System for Online and Offline Medical Data Sharing via Private and Consortium Blockchain: A System Design Study. *Frontiers in Public Health*, **10**, Article ID: 1012202. https://doi.org/10.3389/fpubh.2022.1012202

[28] Kumar, R., Kumar, P., Tripathi, R., Gupta, G.P., Islam, A.N. and Shorfuzzaman, M. (2022) Permissioned Blockchain and Deep Learning for Secure and Efficient Data Sharing in Industrial Healthcare Systems. *IEEE Transactions on Industrial Informatics*, **18**, 8065-8073. https://doi.org/10.1109/TII.2022.3161631

[29] Kumar, P., Kumar, R., Gupta, G.P., Tripathi, R., Jolfaei, A. and Islam, A.N. (2023) A Blockchain-Orchestrated Deep Learning Approach for Secure Data Transmission in IoT-Enabled Healthcare System. *Journal of Parallel and Distributed Computing*, **172**, 69-83. https://doi.org/10.1016/j.jpdc.2022.10.002

[30] Abou-Nassar, E.M., Iliyasu, A.M., El-Kafrawy, P.M., Song, O.-Y., Bashir, A.K. and El-Latif, A.A.A. (2020) DITrust Chain: Towards Blockchain-Based Trust Models for Sustainable Healthcare IoT Systems. *IEEE Access*, **8**, 111223-111238. https://doi.org/10.1109/ACCESS.2020.2999468

[31] Kumar, R., Marchang, N. and Tripathi, R. (2020) Distributed Off-Chain Storage of Patient Diagnostic Reports in Healthcare System Using IPFS and Blockchain. *Proceedings of the* 2020 *International Conference on Communication Systems & Networks* (*COMSNETS*), Bengaluru, 7-11 January 2020, 1-5. https://doi.org/10.1109/COMSNETS48256.2020.9027313

[32] Jamil, F., Ahmad, S., Iqbal, N. and Kim, D.-H. (2020) Towards a Remote Monitor-

ing of Patient Vital Signs Based on IoT-Based Blockchain Integrity Management Platforms in Smart Hospitals. *Sensors*, **20**, Article No. 2195. https://doi.org/10.3390/s20082195

[33] Chakraborty, S., Aich, S. and Kim, H.-C. (2019) A Secure Healthcare System Design Framework Using Blockchain Technology. *Proceedings of the* 2019 21*st International Conference on Advanced Communication Technology* (*ICACT*), Pyeong-Chang, 17-20 February 2019, 260-264. https://doi.org/10.23919/ICACT.2019.8701983

[34] Niu, S., Chen, L., Wang, J. and Yu, F. (2019) Electronic Health Record Sharing Scheme with Searchable Attribute-Based Encryption on Blockchain. *IEEE Access*, **8**, 7195-7204. https://doi.org/10.1109/ACCESS.2019.2959044

[35] Liu, J., Li, X., Ye, L., Zhang, H., Du, X. and Guizani, M. (2018) BPDS: A Blockchain Based Privacy-Preserving Data Sharing for Electronic Medical Records. *Proceedings of the* 2018 *IEEE Global Communications Conference* (*GLOBECOM*), Abu Dhabi, 9-13 December 2018, 1-6. https://doi.org/10.1109/GLOCOM.2018.8647713

[36] Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M. and Wang, F. (2017) Secure and Trustable Electronic Medical Records Sharing Using Blockchain. *AMIA Annual Symposium Proceedings*, **2017**, 650-659.

[37] Bai, P., Kumar, S., Kumar, K., Kaiwartya, O., Mahmud, M. and Lloret, J. (2022) GDPR Compliant Data Storage and Sharing in Smart Healthcare System: A Blockchain-Based Solution. *Electronics*, **11**, Article No. 3311. https://doi.org/10.3390/electronics11203311