Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

ELSEVIER

10th International Conference on Computer Science and Computational Intelligence 2025 (ICCSCI 2025)

Blockchain-based for Securing COVID-19 Patient Medical Records

Abstract

In the digital age, the security and management of medical records have become increasingly critical, especially during the COVID-19 pandemic. Traditional hospital data storage systems often face challenges such as lack of interoperability, centralized vulnerabilities, and data privacy risks. This research proposes a blockchain-based solution to address these issues. By utilizing blockchain's immutability and decentralized access control through smart contracts, the proposed system ensures secure, transparent, and efficient management of COVID-19 patient medical records. The system architecture leverages Ethereum with Proof of Authority (PoA) consensus to optimize energy efficiency and scalability. Experimental implementation demonstrates that this approach enhances data integrity, patient privacy, and healthcare data interoperability. The findings highlight blockchain as a promising method for the secure exchange of sensitive medical data in a decentralized environment.

Keywords: Blockchain; COVID-19 Medical Recors; Smart Contract; Healthcare Data security; Proof of Authority; Data Privacy; Interoperability.

1. Introduction

In this digital era, the technological revolution can bring both positive and negative impacts across all sectors. Technological discoveries are driven by problems, one example being the discovery of blockchain technology in 2008 by Satoshi Nakamoto in his paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System." This technology introduces the concept of blockchain as the underlying technology supporting the digital currency Bitcoin, explaining how this system can replace traditional systems in transaction recording [1]. Over time, however, this technology has even become the foundation for the development of various applications beyond the financial sector, including in healthcare, logistics, education, and government to improve transparency, security, and efficiency of data exchange. In the medical sector, the COVID-19 pandemic has highlighted the need to manage medical records safely and efficiently. Hospitals use traditional storage techniques to store COVID-19 patients' medical records on internal servers that can only be accessed by a single party. Moreover, the lack of interoperability between hospitals hinders the process of exchanging patients' medical information, leading to delays in testing and treatment responses to infections [2]. Therefore, an innovative technology-based solution is needed that can improve data security, accelerate information exchange between health institutions, and uphold patients' privacy rights.

The implementation of blockchain technology offers a promising solution to this issue by utilizing peer-to-peer (P2P) techniques, meaning that no single party has full control over the data [3]. In this system, both the hospital and the patient share access to the transaction data, with patients retaining authority over their own medical records. Patients can also grant permission to other hospitals to view their medical information [4], allowing their medical history to serve as a reference for subsequent diagnoses and treatments.

In this paper, we will focus on the application of blockchain technology [5]. Through this approach, COVID-19 patient medical records are not only stored securely and decentralized, but can also be accessed easily and efficiently by authorized parties through a smart contract-based licensing mechanism. The blockchain is responsible for recording metadata, verifying data integrity, and managing access rights in a decentralized and transparent manner.

2. Related Works

Blockchain has become increasingly relevant in managing decentralized data, offering enhanced security and transparency without relying on a central authority, where there is no central control over access to the data but each party involved has an equal role in access control. This approach appears more appropriate for storing COVID-19 patient medical records, particularly when compared to conventional, hospital-centric systems than internal hospital storage which can only be managed by one party.

Studies have shown promising outcomes when blockchain is applied to EMR systems, particularly in improving data accessibility and operational efficiency in data usage for several purposes [6]. This system allows patients to control access to their data with a smart contract mechanism that regulates access permissions for medical personnel and various health service providers. Blockchain is suitable for reliable information sharing in EHRs because once a transaction has been entered it cannot be altered, using cryptographic functions for secure communication [7]. There are many possible scenarios for the application of blockchain-based EHR in medical data management in the COVID-19 pandemic such as applications include contact tracing and lab record management, along with the distribution of COVID-19 certificates and vaccines and other essential needs efficiently and reliably [6].

Consensus algorithms are vital to validate each transaction before it is recorded, ensuring data consistency across the blockchain network into the system [8]. Then refined with a security system on the blockchain to maintain the medical records of COVID-19 patients using cryptographic technique.

The consensus algorithm mechanism is used by blockchain to process validation operations. This algorithm ensures that all records entered into the blockchain are valid. Proof of Authority (PoA) and Practical Byzantine Fault Tolerance (PBFT) algorithms are more suitable for medical systems compared to others such as proof of work (PoW), as they are more energy efficient and reduce the problem of processing delays [8]. The difference between the two is quite noticeable where PoA allows transactions to be verified by a trusted party such as a hospital or healthcare institution, which can speed up the validation process and also reduce power usage [9]. On the other hand, the PBFT algorithm is designed to overcome system failures by achieving consensus even if there are some parties or nodes that behave dishonestly [8].

This smart contract program runs on top of the blockchain and is executed automatically according to predetermined rules. Smart contracts can be used in medical records systems to automatically manage patient data access rights [4]. Smart contracts can be used to grant or revoke hospitals’ access rights to patients' COVID-19 medical records based on digital consent verified by the blockchain, this can increase transparency and security in the exchange of medical records between hospitals [4].

Large-scale deployment of blockchain faces two main challenges, namely high energy consumption and scalability issues. The enormous power consumption on blockchains is one of the challenges especially those based on Proof of Work (PoW) [9], therefore we choose more energy-efficient consensus algorithms such as Proof of Authority (PoA) or Practical Byzantine Fault Tolerance (PBFT).

3. Methodology

This research adopts an experimental approach through the design and implementation of a blockchain-based prototype system. The objective is to evaluate how blockchain technology can enhance security, transparency, and access control in the exchange of COVID-19 patient medical records.

3.1. System Architecture

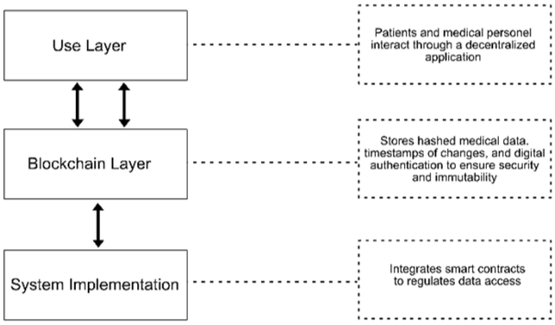
Use Layer
Patients and medical personel interact through a decentralized application
Blockchain Layer
Stores hashed medical data. timestamps of changes, and digital authentication to ensure security and immutability
System Implementation
Integrates smart contracts to regulates data access

Fig. 1. System Architecture of Blockchain Implementation in COVID-19 Medical Records.

Based on Fig.1, there are 3 main layers, such as the Use Layer, Blockchain Layer, and System Implementation Layer. The Use Layer is an interface layer that connects users (patients and hospitals) with blockchain technology. It ensures that users can securely access and manage data through a specially designed frontend application. Authentication based on a public-private key mechanism ensures that each user can only access or modify data according to the specific permissions granted to them. The Use Layer communicates with smart contract methods on the Blockchain Layer to perform operations such as adding, updating, or requesting access to data.

The Blockchain Layer is responsible for storing encrypted hashes of patients' medical data, rather than raw data, to ensure privacy and security. Access to data is managed using smart contracts with access control mechanisms, specifying who is authorized to view, edit, or audit the access history of medical records. The blockchain consensus mechanism employed is Proof of Authority (PoA), allowing only trusted users to perform transactions or update data. The blockchain platform utilized is Ethereum, a decentralized, peer-to-peer (P2P) network that supports scalability through sidechains to handle transaction surges. Additional security measures include the implementation of Public Key Infrastructure (PKI) to ensure that only authorized parties can access or modify stored data.

Through System Implementation Layer, this system adopts blockchain-based cloud computing to secure COVID-19 patient data by integrating the Blockchain Layer for auditing, security, and smart contract-based access control. Patient medical data are not stored directly on the blockchain; instead, only the hash of the medical file is recorded. Any modification to a file generates a new hash that is appended to the next block. Each block records the hash of the medical records, the timestamp of changes, and the public key of the associated hospital. Smart contracts ensure that only authorized hospitals or institutions with valid private keys can update or add data, while patients can access the record history to maintain transparency. All transactions are governed by smart contracts enforcing strict access control policies, thus ensuring the system’s security, transparency, auditability, and protection against the manipulation or theft of medical information.

3.2. Transaction

- addRecords(): Called by hospitals to add new patient data.
- getRecords(): Retrieves all record hashes associated with a specific patient from the blockchain.
- deleteRecords(): Removes access to specific data by deleting the mapping from the smart contract.

- grantAccess(): Grants access rights to new actors by updating the permission list within the smart contract.
- revokeAccess(): Revoke access rights to actors by updating the permission list within the smart contract.

3.3. Interactions

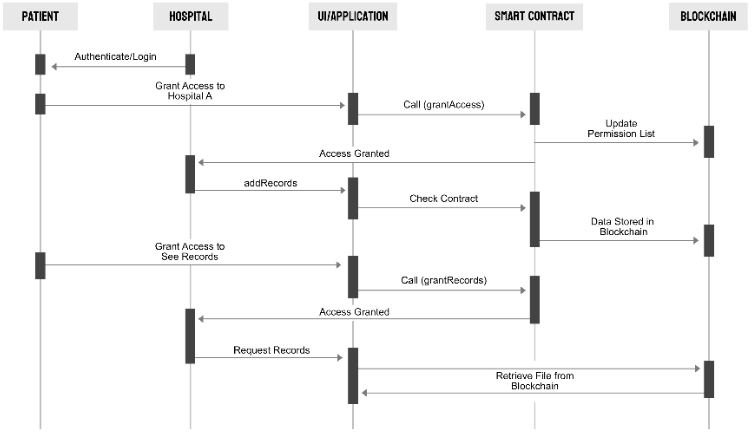

Fig. 2. Sequence Diagram of Interaction COVID-19 Medical Records Access using Blockchain and Smart Contract.

The sequence diagram activities on Fig.2. begin after entity registration and the delivery of medical records to the patient. The following are the steps:

1. The patient will log in/authenticate into the application system to access services that store their COVID-19-related medical data.
2. When undergoing treatment at a specific hospital, the patient will grant access to a designated hospital (e.g., Hospital A) to manage their medical records.
3. Through the user interface (UI) of the application, the patient will invoke the grantAccess() function within the smart contract, which will subsequently update the access permissions list as per the patient's request. Following this, the hospital will receive confirmation that access has been granted and will be able to begin adding new medical records for the patient via the application.
4. If the patient seeks treatment at another hospital and the new hospital wishes to review the patient's previous treatment history, the patient will grant permission for the hospital to access their medical records.
5. The hospital will request the records via the application, which will utilize the COVID-19 related medical data.
6. Subsequently, the hospital will be able to view the patient's medical records and use them for ongoing treatment.

3.4. Estimation Transaction Size

The quantity of transaction data in bytes following serialization using the RLP (Recursive Length Prefix) format is referred to as the transaction size. It represents the transaction's size and complexity, which is important for gauging network resource consumption and calculating storage needs.

$$Transaction\ size = \frac{Raw\ Transaction\ Length\ (\,hex)}{2} \tag{1}$$

In Equation (1), the Raw Transaction Length (hex) refers to the total number of hexadecimal characters in the RLP-encoded transaction string. Since each byte is represented by two hexadecimal characters, dividing the length by 2 gives the actual byte size of the transaction. We can determine the projected transaction size for other functions using this data payload size as a starting point. This method is distinct from the general block size covered in earlier parts and concentrates more on the data payload viewpoint.

3.5. Ethereum Gas Fee Calculation

Gas Limit is the maximum limit of gas units allowed to be used by a transaction on the blockchain network. Its main function is to limit resource consumption so that transactions do not overuse gas that could overload the network. If a transaction exceeds the specified gas limit, it will fail to execute, but the gas fee will still be charged according to the gas that has been used up to the point of failure.

Gas Used is the actual number of gas units consumed during the transaction execution process. The value of gas used reflects how complex and efficient the transaction is. In general, the gas used is always equal to or smaller than gas limit, as gas limit is the maximum allowed.

A special condition occurs if the gas used value is equal to the gas limit. This may indicate that the transaction is using all of the allocated gas.

Ethereum transaction fees are determined by the amount of gas used during the transaction and the current gas price. The calculation for the total transaction fee is expressed in Equation (2):

$$Gas\ Fee = Gas\ Used \times Gas\ Price \tag{2}$$

4. Implementation

4.1. Blockchain Platform Development

Since Ethereum is a distributed blockchain network that employs the blockchain concept seen in the well-known cryptocurrency Bitcoin, it was chosen for this study. Ethereum, the platform's exclusive digital currency, is also used. This cryptocurrency can be shared amongst Ethereum blockchain-connected accounts. Additionally, Ethereum offers Solidity, a programming language that allows developers to create their own blockchain. It was created for Ethereum's smart contracts, a crucial component [10].

External users can use it to change the status of documents or data kept on the Ethereum blockchain network. An Ethereum transaction contains [11]:

- The address of the sender (from) is 20 bytes long.
- The address of the receiver (To) is 20 bytes long.
- Value, or the sum of money sent from the sender to the recipient. The message delivered to the recipient is contained in the optional data.
- Gas, in order to complete a transaction on the Ethereum blockchain, the sender must pay a charge. We call this charge "Gas." Each validator on the Ethereum network uses the gas fee to validate transactions in a decentralized fashion. A gas price and limit are included in every transaction.

4.2. Account Setup

Hardhat is a widely used Ethereum development framework that facilitates the writing, testing, and execution of smart contracts. One of its key features is the built-in local network known as the Hardhat Network, which is lightweight and fast, making it ideal for development and testing purposes. When this network is initialized, Hardhat automatically generates several dummy Ethereum accounts, each equipped with private keys and virtual Ether balances. These simulated accounts allow developers to perform transactions and test their smart contracts without incurring any real costs. Additionally, these accounts can be easily accessed within deployment scripts, enabling smooth and efficient contract deployment to the local network.

4.3. Smart Contract Implementation

As explained earlier, smart contracts are an important part of managing Patient Covid-19 program data as they are used to perform basic operations. Following contracts:

- Covid-19 Patient Records.
- Authorization and Access Control (grant and revoke access).
- Data Integrity and Verification.

This smart contract ensures that the data in a COVID-19 patent is not changed without authorization and that all changes are clearly recorded.

Algorithm 1: Smart Contract Implementation

```
function setPatientProfile(name, birthPlaceDate, age, gender, phoneNumber, vaccinationStatus):
  if patients[msg.sender].name is not empty:
    throw "Patient already registered"
    patients[msg.sender] = Patient(name, birthPlaceDate, age, gender, phoneNumber, vaccinationStatus)

Function grantAccessToProvider(provider):
  Grant access from patient to provider
  Emit AccessGranted

Function revokeAccessFromProvider(provider):
  Revoke provider's access
  Emit AccessRevoked

Function addRecord(hospitalName, testResult, doctorName, treatment, treatmentNotes, medicineNotes, doctorNotes, patient):
  Require provider is authorized and has access from patient
  Create new record with timestamp and sender
  Append to patientRecords
  Emit RecordAdded

Function getRecords(patient):
  Return all records for patient

Function deleteRecord(patient, index):
  If index invalid: Reject
  Shift records to remove index
  Emit RecordDeleted
```

Based on Algorithm.1, to manage patient data and guarantee safe access management, the system has a number of essential features. A patient's name, date of birth, age, gender, phone number, and immunization history can all be entered into their personal profile once. Re-registration is denied if there is already a profile on file. The exclusive authority to add reputable healthcare providers to the list of approved businesses belongs to the system owner. Patients are in complete control of their data and can provide or take away access to particular providers. The system adds providers automatically if they are not previously permitted at the time access is provided. Patients' COVID-19 medical records, including test results, treatments, and doctor's notes, can be added by authorized practitioners. Every record has a time stamp and is linked to the address of the hospital that submitted it. Patient's name, date of birth, age, gender, phone number, and immunization history can all be entered into their personal profile once. Re-registration is denied if there is already a profile on file. The exclusive authority to add reputable healthcare providers to the list of approved businesses belongs to the system owner. Patients are in complete control of their data and can provide or take away access to particular providers.

4.4. Front-End Implementation

At this stage, front-end development is based on the previously designed UI. ReactJS is used due to its component-based architecture, real-time data handling, and strong ecosystem, including React Router, Axios, and Web3 libraries like ethers.js and web3.js, which facilitate blockchain integration. The user interface consists of several core components, ensuring a responsive and interactive experience.

For Patient:

- Patient data input form that will be sent to the blockchain.
- Display of medical history retrieved from the blockchain.
- Access authorization based on user role (e.g. hospital or patient).

For Authorized Providers (e.g. Hospital):

- Patient Record Data form will be sent to the blockchain by Hospital (authorized providers).
- Display of medical patient history retrieved from the blockchain.

4.5. Connecting to Blockchain Systems

Connecting the front-end with the blockchain system uses an endpoint that serves as a bridge of interaction. In this research, integration testing is also carried out to ensure that the interaction between the UI and the blockchain system runs smoothly in the COVID-19 Patient Data system as the main focus. To connect the React interface with the blockchain, libraries such as ethers.js are used to interact with smart contracts that have been deployed. The integration process includes several key stages:

- Connection to a digital wallet (such as MetaMask) to allow users to interact directly with the blockchain.
- Invocation of smart contract functions such as addRecord() or getPatientRecords() from the UI using the contract address and ABI.

Tests are conducted to ensure that:

- Data entered through the UI is successfully saved to the blockchain.
- Data can be retrieved and re-displayed in the UI based on the patient's address.
- Access is only granted to authorized addresses as specified in the smart contract.

With this architecture, the system successfully combines a modern ReactJS-based user interface with the security and transparency of blockchain technology, making the management of COVID-19 patient medical records more secure, decentralized, and accessible in a controlled manner.

5. Results and Discussions

5.1. Transaction Evaluation

Every transaction on Ethereum contains a payload data in every transaction. The transaction contains this payload data, which is utilized to invoke smart contract operations. The necessary bytes are included in the hex-serialized payload data. Payload data is a transaction's optional field that is only utilized when there is contact with the contract function. The selector function and encoded parameters are its two key components.

In evaluating transaction performance, a scripting technique that retrieves and analyzes transaction data on the blockchain network via a local JSON-RPC service provider using the ethers.js library is used to assess transaction performance. In order to undertake a quantitative assessment of the transaction performance, the analysis is carried out by extracting the primary parameters of the transaction and its receipt.

5.2. Function Selector

The function selector is the first 4 bytes of the Keccak-256 hash, which is used to identify the smart contract function being called. Function arguments include various types of static and dynamic elements that have different rules for encoding into the payload.

Table 1. Example of Keccak-26 hash of addMecicalRecords() Transaction.

Data:
" keccak256("addMedicalRecord(uint256,string,string,string,string,address)") = 0xc9fe1b37750c50b06d3f6934f7bc3e4cba08ff7c2595c46e42d5985ac49ed659"

Based on Table 1. the function selector is the 4-byte (‘0xc9fe1b37’) show function addMedicalRecord() is called.

5.3. Encoded Arguments

Function arguments are encoded according to the ABI (Application Binary Interface) standards, which ensure consistency in how data is formatted and interpreted across smart contract interactions. Each parameter is encoded to occupy a multiple of 32 bytes (256 bits), maintaining a uniform structure. For static types such as uint, address, bool, and bytes32, the values are directly represented within 32 bytes. he decoded structure of transaction input arguments, including uint256, string, and address, is displayed in Table. 2.

Table 2. Decoded Transaction Input Parameters Table.

Argument	Type	Value	Notes
1	uint256	0000000000000000000000000000000000000000000000000000000000000001	Score: 1
2	string	0000000000000000000000000000000000000000000000000000000000000140	Offset to string-1 at byte 320 (hex: 0x140)
3	string	0000000000000000000000000000000000000000000000000000000000000180	Offset to string-2 at byte 384 (hex: 0x180)
4	string	00000000000000000000000000000000000000000000000000000000000001c0	Offset to string-3 at byte 448 (hex: 0x1c0)
5	string	0000000000000000000000000000000000000000000000000000000000000200	Offset to string-4 is at byte 512 (hex: 0x200)
6	address	00000000000000000000000070997970c51812dc3a010c7d01b50e0d17dc79c8	Patient Address

5.4. Encoded Dynamic Arguments

For dynamic types (strings, bytes, arrays) contains an offset (32 bytes) at the argument position, then the offset location contains: 32 bytes of original data length and actual data, padded to a multiple of 32 bytes. Each string is encoded in the format: [string length (32 bytes)] [string content in UTF-8 + 32-byte padding]. An example of this structure is illustrated in Table. 3, which shows how the string “sicks” is encoded with a length prefix and padding.

Table 3. Example for offset 0x140 (2nd Argument)

Data:
0000000000000000000000000000000000000000000000000000000000000005 ← panjang string: 5
73616b6974000000000000000000000000000000000000000000000000000000 ← sicks + padding

5.5. Data Payload

Table 4. Data payload of transactions used in the proposed framework.

Data Payload	Content Types	Size
Payload of SetPatientProfile	(string, string, uint8, string, string, string)	352 bytes
grantAccessToProvider	(address)	32 bytes
revokeAccessFromProvider	(address)	32 bytes
addRecord	(string, string, string, string, string, string, string, address)	480 bytes

Table. 4 presents the data payload structures used in various smart contract functions within the proposed blockchain-based COVID-19 medical records framework. Function complexity affects payload sizes, which are expressed in bytes using RLP encoding. Simpler operations like grantAccessToProvider and revokeAccessFromProvider, which only require one address, have smaller payloads of 32 bytes, but addRecord, which requires several text inputs and an address, has the biggest payload (480 bytes).

Table 5. Transactions size and fee for proposed framework.

Data Payload	Total Transaction Size (bytes)	Gas Price (Wei)	Gas Limit (Unit Gas)	Gas Used (Unit Gas)	Gas Fee (ETH)
SetPatientProfile	565	3426907314	164217	164217	0.000477331753258281
grantAccessToProvider	82	2856430796	96754	96754	0.000276371105236184
revokeAccessFromProvider	81	2812164331	28808	23944	0.000067334462741464
addRecord	789	1273206077	325391	325391	0.000414289798601107

The measured transaction sizes, associated gas characteristics, and calculated gas fees are shown in Table. 5. The ethers.js library, which connects to the network via a JSON-RPC provider, was used to simulate the transactions on a local blockchain environment in order to acquire the values. Each transaction hash is used to determine the kind of transaction (legacy or EIP-1559) and to collect execution metadata like gasUsed, gasLimit, and gasPrice.

The complete transaction structure is then re-serialized to determine the transaction size in bytes for each function. This comprises metadata (such as nonce, recipient, gas limit, etc.) and the data payload. Equation (1) is used to determine the transaction size. Equation (2) is used to determine the Ether (ETH) gas fee for each transaction.

The evaluation confirms a direct relationship between the data payload size, transaction size, gas used, and the resulting transaction fee. Larger and more complex payloads, such as in the addRecord function, result in increased transaction sizes and higher gas consumption. This reinforces the importance of designing efficient smart contract functions to reduce the data footprint and improve cost-effectiveness. Such optimization is critical in ensuring the scalability and affordability of blockchain-based solutions, especially in medical and public health applications where frequent transactions are expected.

This calculation highlights the direct relationship between data payload size, transaction size, gas used, and gas fees. Efficient design of smart contract functions to minimize data payload can reduce transaction size, thereby potentially lowering gas consumption and transaction fees. This is critical for scalable and cost-effective blockchain-based frameworks, such as the proposed COVID-19 medical records system.

6. Conclusion

This paper discusses the application of blockchain technology in the management of COVID-19 medical records with a focus on security, integrity, and access control of sensitive medical data. The developed system ensures that patient data is stored securely and can only be accessed by authorized parties, thereby increasing transparency and trust in information management during the pandemic.

For future development and research, several things need to be considered so that the system can run more optimally. First, increasing scalability is essential by adopting a more efficient consensus algorithm so that the system can accommodate more users without degrading its performance. In addition, the implementation of load balancing so that the workload can be divided evenly to avoid system down due to too heavy workload and make the system more responsive. Resource management should also be done dynamically, by adjusting computing capacity based on load levels, to better handle spikes in usage.

For the next development plan, we are also considering adding a payment module to the existing system. This is important because it determines the fees that patients have to pay for consultations with doctors in this decentralized system. In addition, it is necessary to create policies and rules that are in accordance with the principles of the health sector so that the implementation runs well and safely.

By implementing these recommendations, the blockchain-based COVID-19 patient data recording system can become more reliable, scalable, and ready for large-scale use, providing a safe and transparent solution for medical data management during the pandemic.

References

[1] Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. n.d.

[2] Chang MC, Park D. How Can Blockchain Help People in the Event of Pandemics Such as the COVID-19? J Med Syst 2020;44:1–2. https://doi.org/10.1007/S10916-020-01577-8/METRICS.

[3] Marbouh D, Abbasi T, Maasmi F, Omar IA, Debe MS, Salah K, et al. Blockchain for COVID-19: Review, Opportunities, and a Trusted Tracking System. Arab J Sci Eng 2020;45:9895–911. https://doi.org/10.1007/S13369-020-04950-4/TABLES/4.

[4] Madine MM, Battah AA, Yaqoob I, Salah K, Jayaraman R, Al-Hammadi Y, et al. Blockchain for Giving Patients Control over Their Medical Records. IEEE Access 2020;8:193102–15. https://doi.org/10.1109/ACCESS.2020.3032553.

[5] Haleem A, Javaid M, Singh RP, Suman R, Rab S. Blockchain technology applications in healthcare: An overview. International Journal of Intelligent Networks 2021;2:130–9. https://doi.org/10.1016/j.ijin.2021.09.005.

[6] Reegu FA, Daud SM, Alam S, Shuaib M. Blockchain-based Electronic Health Record System for efficient Covid-19 Pandemic Management 2021. https://doi.org/10.20944/preprints202104.0771.v1.

[7] Hilal AA, Badra M, Tubaishat A. Building Smart Contracts for COVID19 Pandemic Over the Blockchain Emerging Technologies. Procedia Comput Sci 2022;198:323–8. https://doi.org/10.1016/J.PROCS.2021.12.248.

[8] Ferdous S, Jabed M, Chowdhury M, Hoque MA, Colman A. Blockchain Consensus Algorithms: A Survey. Lecture Notes in Networks and Systems 2020;595 LNNS:198–210. https://doi.org/10.1007/978-3-031-21229-1_19.

[9] Chen Y, Li M, Zhu X, Fang K, Ren Q, Guo T, et al. An improved algorithm for practical byzantine fault tolerance to large-scale consortium chain. Inf Process Manag 2022;59:102884. https://doi.org/10.1016/J.IPM.2022.102884.

[10] Shahnaz A, Qamar U, Khalid A. Using Blockchain for Electronic Health Records. IEEE Access 2019;7:147782–95. https://doi.org/10.1109/ACCESS.2019.2946373.

[11] Version B. ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER n.d.