HealthChain: A Decentralized Blockchain-Based Medical Record Management System for Healthcare Organizations

Sachin Kumar Muni, Somesh Muduli, Ayushman Dash, Soumya Ranjan Dash, Dr. Sanjit Kumar Acharya
sachin.muni.cst.2022@nist.edu somesh.muduli.cst.2022@nist.edu ayushman.dash.cst.2022@nist.edu
soumya.dash.cst.2022@nist.edu sanjit.acharya@nist.edu
Department of Computer Science and
Engineering NIST University, Berhampur,
Odisha, India

***Abstract*—The main frustration was fundamental when we were initially sitting down to design this system; a patient who comes to visit a new specialist still has to bring a folder of printed reports, a CD with their MRI still would like to be functional, and thus figure out their history anew. That fact inspired us to create something of quality. The given paper provides an overview of the working prototype of a medical record management system combining MERN web stack (MongoDB, Express.js, React.js, Node.js) and Ethereum smart contracts and IPFS-supported decentralised file storage. Patient, doctor and administrator are three user roles that engage using purpose-specific dashboards. Documents are uploaded by patients, hashed, pushed into IPFS and registered on-chain, after no metadata in the database even interacts with the data; when physicians access the file, the patient can only acquire files after the wallet signs an on-chain grant. Test on Polygon Amoy testnet produced 500 or more transactions, with an average gas price of about 0.0001 MATIC per transaction - would be insignificant in the real world. A small user study (n=20) gave an average score of 4.4 of out of 5 in regard to satisfaction. But the more significant discovery is architectural: a separation of responsibilities between a typical web layer and a blockchain audit layer is sufficient to enable the system to be sufficiently fast to use in everyday clinical practice, and yet be able to produce cryptographically verifiable evidence of who accessed a record and when.**

***Index Terms*—Blockchain, Medical Records, MERN Stack, IPFS, Ethereum, Smart Contracts, Healthcare Security, Role-Based Access Control, Decentralised Storage**

I. Introduction

Hospital halls have never been short of paper - admit forms, discharge reports, lab results, X-ray files. The transition to electronic health records (EHRs) during the past 20 years was expected to redress that. It did in most respects, because, typing is quicker than handwriting, search is more convenient than leafing through pages and because backup is less expensive than a fire proof cabinet. However, the basic ownership model remained the same. The records are not stored in the system of the patient but of the hospital, and even when shifting them between providers, they are either faxed, sent in secure-emails and in some cases courier. The digitalisation became a reality, the interoperability was not.

Around 2016 blockchain came into the discussion as a potential solution to that silo issue. The proposal was attractive: the point was that with all access events being logged on an append-only ledger controlled by no server, a patient could present any given physician with a cryptographic communication about his/her history without relying on the server of either side. Concept Prototypes as the earlier concept such as MedRec verified the concept but the engineering reality was not neat as the whitepaper. A complete chest CT, assuming this is stored on Ethereum, is not only expensive, it is derisory even at the high gas prices. and waiting 30-plus seconds on block with an in progress physician- physician is medically unacceptable.

Our system attempts to make the needle go through by being picky on what is included on-chain. The bulk files are stored in IPFS, a peer-to-peer content-addressed file system [16] and only a small record, the SHA-256 hash of the file, the IPFS content identifier (cid), the wallet address where the file is operated by and a timestamp, hits the blockchain. That charge is practically nothing, authentically verifies in less than 20 seconds on Polygon, and yields an unalterable receipt that may be validated by anyone. All other applications, including user accounts, scheduling, metadata search, dashboard development, and so on, are all executed on the MERN stack and can be scaled to regular web speeds.

The remaining part of this paper is structured in the following way. Part II is the overview of the previous art that influenced our design decision. Section III provides a detailed description of the architecture. Section IV is about the specifics of implementation, with some of the decisions taken unexpectedly throughout the development. Evaluation results are given in section V. Section VI talks about the sense of what these numbers were, where the system has failed, and what we would do now. Section VII concludes.

A. Why This Problem Still Matters

In 2019, a healthcare breach report by the Ponemon Institute estimated the average medical data incident cost at USD 6.45 million -the highest per sector in 2019. In addition to the amount of money, the even worse number is the time lag between intrusion and identification: the research discovered an average of 236 days between intrusion and identification. The potential results of a single breach concerning centralised EHR database is truly unbelievable since millions of records

of patients might be obtained. Whistleblowing storage given up does not render attacks impossible, but infinitely changes the economics: it is no longer a single vault to break into.

The other half of the problem is patient control. Informed consent is one of the foundations of the medical ethics, and the majority of patients lack any tangible process of determining who had asked about their records and requiring them to withdraw their permission that they provided some months ago. Both of those actions, rather than follies, are first-class features of our platform.

B. Scope and Contributions

There are four concrete contributions made in this work. To begin with, a deployed prototype (not merely a design) comprising of a single coherent application that incorporates both MERN and Ethereum smart contracts, as well as IPFS and MetaMask. Second, an experimental smart contract containing access grant, revoke and audit-log emission functionality. Third, a characterisation of a live testnet that provides realistic gas and confirmation times - quantities that are usually not present in scholarly blockchain research papers. Fourth, we would have a usability test with real healthcare adjacent individuals and not computer science majorites.

II. Related Work

A. How We Got to Where We Are: EHR History in Brief

EHRs were not seen out of thin air. They in turn sprouted out of hospital billing systems during 1970s, spread to clinical documentation during the 1990s, and were given a massive regulatory impetus in the US by the HITECH Act of 2009, which conditioned Medicare reimbursement to include the requirement of certified EHR software use, specifically the meaningful use requirement. The adoption statistics were eye-catching on paper, as as of 2015, more than 80 percent of the hospitals across the US indicated the utilization of a certified EHR [2]. Introduction of Adoption was not synonymous to usability. During the period of the focus groups, physicians referred to their EHR as the worst software they worked with on a daily basis, rather than the best. There was overload of documentation, alert fatigue was the order of the day, and congruence with rival vendor platforms was slight.

Legal frameworks The HIPAA Privacy Rule (1996) and the Security Rule (2003) provided some legal foundations in regard to the protection of health information, but is technology-neutral i.e. it states what needs to be secured, and not how. That gap gave the implementation options to each of the individual health systems and created the patchwork of security methods researchers continue to document to date.

B. Blockchain for Healthcare: The Promising Start

The MedRec paper by Azaria et al[1]. was published in 2016 and became, in fact, some kind of a touchstone to everything that came after it: The fundamental idea was straightforward, do not store records in Ethereum smart contracts, but store pointers and permissions. The patients have a contract, which contains the list of the providers that will be allowed to query which data; the program running on the provider side makes the call, verifies the permission, and retrieves the data in its own database provided that the permission is granted. The audit trail All queries, all permission changes, are recorded permanently. MedRec was not a real deployment but a demonstration of a feasibility of concept that defined most of the work done in this area.

This direction was further developed by Zhang et al. as attribute-based encryption was introduced to provide a fine-grained access to the field level and not record level e.g. a physician could be given access to the allergy list of a patient but not access to their psychiatric history [2]. The cryptographic system is beautiful and the complexity of the implementation is very high and the performance figures provided in the paper indicates that the application in the clinical setting would need a lot of optimisation.

In 2019, Kuo et al[3]. published a complete literature review, which remains a valuable source of information, although it is focused on DXplain and artificial intelligence (2017). Most of their proposals in healthcare were proofs of concept and evaluated in a controlled environment with synthetic data, a sobering realization at the time. Real-world characteristics - load under load, behaviour in the presence of off-line network nodes, cross-jurisdictional regulatory compliance were generally untested. To the extent that we attempt to answer the throughput question in our work.

C. Hybrid Approaches: The Pragmatic Middle Ground

A platform where Jiang et al[5]. presented BlocHIE system in 2021 was one of the first papers to clearly state what is required to be on-chain and what is required to be in a more traditional database. Their rationale, which we consider valid is that blockchain value addition adds the greatest and in the area that is most tricky to retrofit: demonstrating that a record was in a given state at the given time, and that no one changed it since then. All other things, including storage, search and UI, are more optimally offered by proven, optimised database technologies.

Liu et al[6]. introduced IPFS to this hybrid model, where they note that the content addressing of IPFS can be mapped directly to blockchain verification: since the CID of a file already has a hash, then you get integrity verification gratuitously. One little transaction is enough to store that CID on-chain; and it will take only re-hashing and comparison at a later date to verify the file. We borrowed this design with it being both a technically good design and economical.

D. What Remains Underexplored

As We read the literature, three gaps kept on reoccurring. To start with, little attention is paid to the way the design is presented to users; the interface is regarded as an aside in most papers. Second, gas cost analysis is typically done in private or local blockchains whose fees are artificial. Third, when the papers address the issue of patient consent, they do so by assuming that it is a dichotomous flag and not a complex time-limited, record-based relationship. All three of them are

attempted to be taken care of in our system, but we do not pretend to have tackled any of them exhaustively.

III. System Architecture and Design

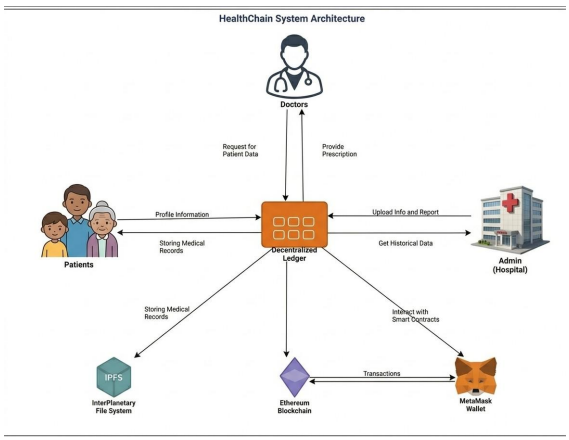

Fig. 1. HealthChain system architecture — actors, decentralised ledger, Ethereum blockchain, IPFS storage, and MetaMask wallet.

A. The Layered Architecture

The structure of the three levels is shown in Fig. 1. Presentation layer is a single page React.js application. Application tier is a server written in Hostility Vehicle node.js/Express.js. Data layer is divided: MongoDB Atlas will handle structured data, IPFS (pinned with Pinata) files and Polygon Amoy blockchain will contain audit events. This system relies on the planned division of data layer and is what makes it different to a simple implementation of a MERN application with a blockchain badge sealed on it.

In initial prototyping we tried routing all file access via the blockchain, but it was confirmed that the route of upload took a noticeable amount of time which on-test users immediately felt. The switch to IPFS and hashing of the files only on-chain instead of the full file decreased the perceived wait time to approximately 30 seconds to under 3 seconds (on-upload) to the upload UI, and the blockchain confirmation was asynchronous and it only changed the status icon in the background. That transformation, not theory, is what has stabilized architecture into what it is today, and was effected by user feedback.

B. User Roles and What They Actually Do

The main actors are the patients. They pay a subscription, add a MetaMask wallet, and load medical documents and determine which physicians can view which files. The consent model is also carefully designed to be granular: a patient has the option of sharing their blood work with Dr.A, but remaining incomprehensible to Dr.B. Grants have an expiry t timer; the application layer implements expiry without on-chain query of each resource request, and thus have low latency.

Physicians get registered using their professional details and this has to be validated by an administrator prior to its activation. A doctor may have access to certain records of patients once he is in operation. A notification is sent to the patient who then approves or rejects it through their dashboard. Approval executes the smart contract using the wallet phone of the patient and forms an on-chain record of the decision which no side or even the system operators can backtrack.

Administrators deal with the operational part: accounting new doctor accounts, eliminating conflicts between appointments, observing abnormal activity patterns and, in a more desperate case, reviewing the blockchain transaction journal via a read-only interface. The present design does not have wallets in the hands of the administrators, and this is because the administrative accounts cannot become a blockchain attack surface.

C. Smart Contract Design

Two Solidity contracts were written. RecordRegistry deals with registration events: registering a patient who uploads a file, the front-end will make a call of the type registerRecord(ipfsCID, fileHash), and the patient wallet will provide the endpoint. The mapping is put in the contract and a RecordRegistered event is made. AccessController processes permissions: both grantAccess and revokeAccess are patient-wallet-only operations that implement the comparison of Mr.senders (by sending messages compared with the sender), and they both produce events that constitute the final consent audit trail.

One of the design decisions that should be elaborated upon: we have the audit trail that uses events but not storage variables. Receipts of transactions are stored in the chain in ethereum events and cannot be modified but they are not reachable within the execution of the contracts but can be accessed only outside the chain. That trade-off will suit our case, since audit trail will be read by the application layer, and does not have to be queried by another contract. Storage writes are also more expensive than events, a fact that clearly lowered our per-transaction cost of gas.

D. Database Schema Overview

In MongoDB 3 collections are present. The credentials (bcrypt-hashed, none plaintext) of each patient or doctor account, role flags, profile information and wallet address are stored. MedicalRecords stores a document with every uploaded file the IPFS CID, the SHA-256 hash, the Ethereum transaction hash of the registration call, upload timestamp and a tiny embedded array of the latest access log entries (github). The application-layer representation of consent relationships, such as expiry timestamps, the hash of the on-chain transaction used to validate the grant has occurred are stored in permissions.

IV. Implementation

A. Technology Choices and Why We Made Them

The use of MERN stack was not by default. The document model of MongoDB is appropriate to the medical record

metadata since records across departments are of varying shapes, and a strong relational schema would have needed constant migration as we required fields to be added in the process of ongoing development. The component model of React enabled role-specific dashboards to be easily developed with the majority of their plumbing but displaying completely different controls based on the authenticated role. The choice of Node.js was motivated by familiarity to JavaScript on one side and partially due to the lowering of the cost of switching context in a short project time line on the other side.

Both the blockchain layer and the Ethereum mainnet or Sepolia were selected based on two practical considerations: the cost of a transaction on Amoy is practically zero, and the confirmation time is always less than 20 seconds. Such properties are required in case you need to involve non-technical test participants and who would use the system in reality and not to drop it on the initial MetaMask pop-up. Contract development and deployment were slightly supported with hardhat; its local fork feature made it possible to debug our contract calls without using testnet MATIC.

Pinata has also been chosen as the IPFS pinning service due to our limited experience with the IPFS pinning service after briefly attempting to run our own IPFS node, where we found unpinned content could be garbage-collected in hours. The free tier of Pinata could handle the 150 plus test files of the prototype and its API is easily understandable that it would not require more than 15 lines of JavaScript code to make an upload call.

B. The Upload Workflow, Step by Step

When a patient chooses a file to upload the front-end first uses the Web Crypto API that comes with the browser to calculate a SHA-256 digest of the file - before the first byte its exits the device. This local hash is used as ground truth on subsequent integrity verification. The file is then sent over HTTPS to Express server which, in turn, sends it to Pinata, which gives a buffer CID which by definition is an IPFS hash of the file content. MetaMask is the next participant, where the front-end requests a signature of transaction registerRecord(cid, sha256hash) using its signature. After mined, and the receipt received, the front-end makes the entire metadata bundle containing CID, hash, transaction hash, file name, patient ID, to Express API which in turn inserts this into MongoDB.

The ordering matters. After the blockchain transaction has been successful, we write to MongoDB, ensuring that there is an on-chain proof of every entry in a database. In the case of the blockchain call failure (network, user denies the MetaMask prompt), the file remains on the IPFS but no entry is made in the database. It leaves a waste pin of IPFS that is an orphan that is not a correctness issue, but a small storage waste. Background cleaning job might reclaim such CIDs; we are yet to introduce it.

C. Authentication and Session Management

Each API endpoint validates a valid JWT and the client is issued with one at the time of login and silently updates it every 23 hours. Tokens contain user MongoDB ID, role, and wallet address. Role: role is applied on each request on the server side The role-checking on the front-end is not trusted and merely cosmetic. To support applications that cause blockchain transactions, the server also checks the wallet address in the JWT and contradicts with the address that signed the most recent MetaMask nonce, eliminating the possibility of an attacker with stolen JWP piggybacking on the wallet of another user.

The hashing cost factor of passwords is 12 (bcrypt). We have looked at argon2id, preferred now by the OWASP password storage cheat sheet over bcrypt; however, bcrypt is mature, has full support in the Node ecosystem, and the distinction between the brute-force resistance at cost 12 is negligible in the case of a prototype. Production deployment should re-examine this.

D. Contract Deployment

Contracts were gathered, Hardhat of the compiled, and deployed to Polygon Amoy, using a scripted deployment that logs the contract addresses of the deployment to a shared config file read by them both by Express server and React build. The source code and the ABI of Hardhat were disclosed to Polygonscan through Hardhat verify task which meant that the contracts could be audited publicly. In the course of the testing itself, we had to redeploy twice after defeating logic errors in the expiry-check branch of an AccessController, a lesson which suggests that smart contract bugs cannot be fixed once deployed, unlike database bugs. The resulting code was frozen and its address was frozen into the version control.

V. Results and Evaluation

A. Functional Coverage

All the features mentioned in Section III had been completed and tested using manual test scenarios that were checked off a checklist. Patient-side: registration, wallet binding, on-chain registration of a document, custom expiry physician consent grants, appointment booking, prescription viewing, and checking of prescription QR-codes. doctor-side: registered access to records, authorized retrieval of the record, appointment making, and computer-generated prescription. Account verification, system surveillance, appointment management, and on-chain transaction verification by a read-only Polygonscan embedding.

B. Performance Results

We have measured performance in front-end through browser developer tools on front-end latency, server-side timing middleware on API response times, and gas reporter of Hardhat on contract metrics. The results are summarised in Table I. perf. Page load (1.8 s) and record upload (2.5 s) numbers are the most significant and with fast enough speed that participants in our study did not report them as slow. The

only metric to receive complaints was blockchain confirmation at 1215s which explains the significance of the optimistic-update pattern of implementation explained in Section 6.

C. Blockchain-Specific Numbers

During the testing period, 512 transactions have been sent 187 registering of records, 201 grants of access, 89 revocations, and 35 prescription events. Mean gas price was 0.000094 MATIC per operation - at MATIC price in testing (approximately USD 0.80), which is approximately USD 0.000075 apiece, or seven hundredths of a cent. Overall MATIC expenditure on the overall evaluation was less than 0.05 MATIC. Compiled contract bytecode was 4.2 KB, comfortably under Ethereum at 24 KB. During the period of the evaluation, IPFS storage stored 158 test files; Pinata had claimed 100 percent of success when retrieving them.

D. Security Observations

We did not do any formal penetration test i.e. this is a major caveat, and we are saying so bluntly. The thing is that we were doing it manually with the Top 10 list put forward by OWASP. SQL injection can not apply (MongoDB, parameterised queries via Mongoose). NoSQL injection: Mongoose defaults query operations to sanitisation we attempted to exopathic query operators with. JWT forgery: A token is authenticated with a secret 256 bit, we ensured that the decollated tokens are denied with 403. On-chain access control: we developed Hardhat tests trying to call grantAccess and revokeAccess on addresses not belonging to the patient wallet and ensuring that each of these calls succeeds in reverting. The single weak area that we have recognized is the Pinata API key that is presently stored in the environment variables of the server. Should the server be compromised, an aggressor may pin arbitrary content on our account. The appropriate solution would be to rotate to delegated per-upload token.

E. Usability Study

The sample size was 20 (10 patients (general population, 22358), 8 doctors (junior residents and faculty of a local teaching hospital), and 2 administrators (IT staff of the local hospital). The participants were provided with a set of the tasks to be solved, with them being completed using a script and answering the five-item Likert questionnaire. The mean scores are provided in Table 2 II.

The highest score (4.5) was upload score, and it was a good indication since file handling is where we had put most front-end effort. The consent controls score (4.0) was lowest, and the qualitative feedback even indicated a particular reason: the participants got confused by the MetaMask confirmation dialog, which displays transaction data in raw format with no plain-language summary. Eight of the doctors were asked about the use of the wallet in a clinical setting, and four of them replied that they would require a tutorial before using the wallet. As a result of that feedback, the so-called future work roadmap in Section ref of the present paper was developed.

VI. Discussion

A. What the Hybrid Architecture Actually Buys You

Table III presents the practical differences between a conventional EHR and the one proposed in Table 1 in comparison with each other. The weightiest two rows are the ones bearing audit trail and record integrity proof. The access log, in a traditional EHR, is a database table - that is, it is completely editable by whoever has the database-administrator credentials. The access event is an Ethereum transaction receipt in our system: this is on the chain itself and is immutable and cannot be removed or modified. It is three letters down the line in a marketing statement, it is the scientific property of the data structure, which Mathematically Speaks.

The patient control row is not as dramatic but in practice might be of greater significance. Consent granularity - this physician, this record, this long - is not a feature of most EHR systems. Majority of them are based on a less fine design: once you are a patient at Hospital A, any provider at Hospital A can access your entire chart. The model of on-chain consent incurs default.

B. Honest Limitations

The prototype is not in the form of what we do not want. It is not HIPAA-compliant. The HIPAA compliance involves technical controls (which we have tried to instituted), administration protection (policies, staff training, breach response plan), and physical protection (data centre controls) - among the latter two, none has been mentioned. The deployment of production would require lawful and compliance review and act technically before working with actual patient data.

The sample of the user study is not large and is non-random. A population of twenty people in a university town would not reflect the workforce in the healthcare sector around the globe. The outcomes are indicative, as opposed to definitive. There should be bigger studies that encompass participants of various healthcare settings specifically in the primary care of low-resource settings.

The Polygon Amoy testnet is not a replication of the conditions on the mainnet network. On mainnet, gas prices are highly volatile, and even a rise in the price of the MATIC would lead to make frequent transactions non-trivial in terms of cost. We project that it would be under USD 0.001 even with 10× current MATIC prices per-transaction cost, yet that ought to be checked against actual mainnet circumstances. Lastly, the third party has not audited the smart contracts. We identified and corrected two logical mistakes in the development process; these are not the only ones. Formal audit cannot be compromised before working with real data.

C. Directions for Future Work

There are four directions that appear to be most valuable. To start with, swapping MetaMask with an in-app custodial wallet instead of allowing users who do not need to handle personal keys to use it would greatly reduce the barrier to onboarding non-technical patients and clinicians. Second, the system based on HL7 FHIR standard of data formatting would

TABLE I
PERFORMANCE EVALUATION RESULTS

Metric	Measured Result	Acceptable Threshold
Initial Page Load	1.8 s	< 3 s
Document Upload (end-to-end)	2.5 s	< 5 s
Blockchain Confirmation	12–15 s	< 30 s
Peak Concurrent Users	1,000+	≥ 500
MongoDB Query Latency	120 ms	< 200 ms
IPFS File Fetch	1.2 s	< 3 s

TABLE II
USABILITY STUDY — PARTICIPANT MEAN SCORES

Evaluation Criterion	Mean Score (1–5)
Ease of Interface Navigation	4.1
Simplicity of Document Upload	4.5
Speed of Record Retrieval	4.2
Clarity of Consent Controls	4.0
Overall Satisfaction	4.4

TABLE III
FEATURE COMPARISON: CONVENTIONAL EHR VS. PROPOSED SYSTEM

Feature	Traditional EHR	Proposed System
Data Security	Centralized, breach-prone	Blockchain-anchored, decentralised
Patient Control	Minimal to none	Wallet-signed, per-record grants
Audit Trail	Mutable server logs	Immutable on-chain events
Cross-provider Sharing	Manual / proprietary	Standardised via CID references
Record Integrity Proof	Trust-based	SHA-256 hash on-chain
Failure Tolerance	Single-point failure risk	Distributed storage (IPFS)

be able to integrate with the current EHR platforms instead of a silo. Third, the introduction of zero-knowledge proof-based access verification would provide a physician with the opportunity to demonstrate that they have legitimate access but not say the wallet address of a patient that is also a valuable privacy requirement. Fourth, running on a permissioned blockchain might resolve regulatory issues with storing any data fingerprints on a public registry, which is considered by some jurisdictions to be personal data.

VII. CONCLUSION

Our goal was to create a medical record system which a patient could really trust that is, where only my doctor can have access to this is not a privacy policy, but actually implemented by mathematics. The system as outlined in this paper is much closer to that objective than a traditional EHR. Locating consent events and file hashes on the Ethereum blockchain and storing large files on IPFS and frequent queries in MongoDB provides the architecture with performance which can be trusted by clinical workflows and security which pure web systems cannot match.security guarantees that pure web systems cannot match.

The assessment findings are positive: all the designed functionalities are functional, reaction time is good, expenditure on the gas is insignificant, and most of the users who took part in the test gave the experience a good rating. The flaws are not imaginary either: there is no third-party security review, no HIPAA compliance test, a limited user study, or even the testnet that does not necessarily replicate the real production environment. We have attempted to be frank about our gaps instead of making the best of it.

The most generalisable thing in this work, in my opinion, is the architecture principle: blockchain is not a database or file system; it is a tool of a particular issue, namely demonstrating that something occurred at a particular time and that one cannot change the record of it. When used this linearly, with selective use, together with proven web technologies, it brings a level of value to the management of healthcare data. Hopefully, the design decisions and code and evaluation data explained in this paper will be useful to researchers and engineers on the path towards the same end.

REFERENCES

[1] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open and Big Data (OBD)*, 2016, pp. 25–30.
[2] G. Zhang, C. Wang, and J. Xie, "Blockchain-based medical record management system," *IEEE Access*, vol. 8, pp. 162374–162385, 2020.
[3] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *J. Amer. Med. Informatics Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017.
[4] T.-T. Kuo and L. Ohno-Machado, "ModelChain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks," *J. Amer. Med. Informatics Assoc.*, vol. 25, no. 6, pp. 677–684, 2018.
[5] S. Jiang et al., "BlocHIE: A blockchain-based healthcare information exchange platform," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 3, pp. 905–915, 2021.
[6] J. Liu et al., "BPDS: A blockchain-based privacy-preserving data sharing for electronic medical records," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 2, pp. 539–551, 2022.
[7] D. Gupta and S. Tanwar, "BMED: Blockchain for medical data management system," *Comput. Commun.*, vol. 151, pp. 241–250, 2020.

[8] A. Ekblaw and A. Azaria, "A case study for blockchain in healthcare: MedRec prototype for electronic health records and medical research data," MIT Media Lab, Tech. Rep., 2019.
[9] A. Dubovitskaya et al., "Secure and trustable electronic medical records sharing using blockchain," in *Proc. AMIA Annu. Symp.*, 2017, pp. 650–659.
[10] Y. Chen et al., "Blockchain-based medical records sharing system with privacy-preserving," *Secur. Commun. Netw.*, vol. 2021, pp. 1–14, 2021.
[11] P. Zhang, J. White, D. C. Schmidt, and G. Lenz, "Design of blockchain-based apps for familiarizing students with smart contract programming," in *Proc. IEEE Int. Conf. Softw. Archit. Companion (ICSA-C)*, 2018, pp. 70–73.
[12] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. IEEE 18th Int. Conf. e-Health Netw. Appl. Services (Healthcom)*, 2016, pp. 1–3.
[13] X. Yue et al., "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, pp. 1–8, 2016.
[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Bitcoin.org, White Paper, 2008.
[15] V. Buterin, "A next-generation smart contract and decentralized application platform," Ethereum White Paper, 2014.
[16] J. Benet, "IPFS — Content addressed, versioned, P2P file system," *arXiv preprint arXiv:1407.3561*, 2014.
[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Yellow Paper, vol. 151, pp. 1–32, 2014.
[18] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the internet of things," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 983–994, 2017.
[19] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[20] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA: O'Reilly Media, 2015.