

Patient-Centric Secure Medical Record Sharing on Ethereum: A Proof-of-Concept Study of Smart Contract-Based Access Control

Jianhao Yuan

*Guangzhou Minxin school, Guangzhou, China
minxin20249457@minxinnansha.org.cn*

Abstract. Fragmented electronic medical records (EMRs) across healthcare institutions hinder coordinated care and increase cybersecurity risks for sensitive patient data. While blockchain has been proposed for medical data governance, existing architectures remain often overly complex and lack minimal, verifiable implementations. This paper presents a proof-of-concept (PoC) using a minimal Ethereum smart contract to enforce patient-centric authorization and provide tamper-evident audit logs for EMR references without storing clinical data chain. Through analysis of prior systems (MedRec, FHIRChain, ACTION-EHR) and a Solidity implementation, it is demonstrated that deterministic on-chain execution can effectively manage access control lists (ACLs) and log access events. The PoC implements three core functions—setRecord, grantAccess, and getRecord—and validates them against core requirements: authorized access, denial for unauthorized users, and immutable event logging. Evaluation confirms the contract delivers transparent governance but exposes key limitations: poor scalability, weak privacy engineering, smart contract security risks, and missing real-world capabilities like access revocation and emergency override. Findings show that Ethereum smart contracts can serve as a complementary authorization and audit layer for EMR sharing, but not as replacement for interoperability standards (e.g., HL7 FHIR) or conventional security controls.

Keywords: Blockchain, Electronic Medical Records, Smart Contract, Patient-Centric Access Control, Healthcare Data Sharing

1. Introduction

Electronic Medical Records (EMRs) provide longitudinal clinical histories that can improve patient outcomes and healthcare delivery. However, patient data remains distributed across multiple hospitals, clinics, and vendor systems, creating information silos that result in repeated intake processes, delayed cross-provider coordination, and incomplete clinical information. This fragmentation creates significant cybersecurity and compliance risks, as healthcare organizations must satisfy strict requirements for access control, data integrity, authentication, and privacy protection [1, 2]. Blockchain technology has emerged as a potential solution to these governance challenges. By providing deterministic execution and tamper-evident logging, blockchain-based

systems offer mechanisms to enforce explicit and auditable access policies across organizational boundaries.

While numerous studies have explored blockchain-based healthcare data sharing (e.g., MedRec FHIRChain, ACTION-EHR), many existing proposals either present high-level architectures or introduce complex systems that obscure fundamental access-control mechanisms [3-5]. Therefore, a minimal and verifiable prototype is needed to evaluate whether blockchain-based permission management can serve as a cybersecurity-oriented governance layer for EMR sharing.

To address this gap, this study focuses on authorization for retrieving EMR references and investigates the following research question: Can a minimal Ethereum smart contract enforce patient-centric authorization for EMR access and produce evidence of compliant access decisions while respecting the limitations of on-chain computation and storage?

This paper makes three primary contributions: (1) we review representative healthcare blockchain architectures and identify the dominant hybrid pattern where clinical data remains off-chain while permissions and hashes are stored on-chain. (2) We design and implement a minimal Solidity smart contract that realizes patient-centric access control through three core functions. (3) We assess the proof-of-concept (PoC) from a cybersecurity perspective, discussing its security, privacy, cost, and compliance, and practical deployment challenges.

2. Background and related work

2.1. Foundational concepts

Blockchain and Smart Contracts: Blockchain, introduced by Nakamoto, is a decentralized, tamper-evident ledger [6]. Ethereum extends this concept with smart contracts—self-executing programs that run deterministically on the blockchain [7, 8]. These contracts enable "proactively enforced rules" without requiring a trusted third party [9].

EMR Interoperability Standards: The HL7 FHIR (Fast Healthcare Interoperability Resources) standard defines a framework for exchanging clinical data between systems [10]. It provides RESTful APIs and resource definitions but does not inherently solve cross-institutional authorization governance.

Regulatory Context: HIPAA technical safeguards (45 CFR 164.312) require access controls, audit controls, and integrity mechanisms for ePHI [1]. Similarly, GDPR mandates data protection by design and accountability for health data processing [2].

2.2. Related healthcare blockchain systems

Several influential projects have explored blockchain-based medical record governance, as summarized in Table 1.

Table 1. Comparison of representative blockchain-based EMR systems

Project	Purpose	Clinical Data	Access Control	Strengths	Limitations
MedRec	EMR access	Permissions, logs	Smart contracts	Auditability, patient control	No on-chain records
FHIRchain	FHIR sharing	Resource references	Backchain authorization	HL7 Interoperability	Complex integration

Table 1. (continued)

Action-GHR	Fine-grained sharing	Metadata	Patient-specified policies	Flexible Permissions	Scalability issues
HealthChain	EHR management	Encrypted indices	Permissioned access	High throughput	governance overhead
HealthCare Gateway	Patient-controlled sharing	Health records	Privacy-preserving policies	Strong privacy	High computation

2.3. Research gap

Existing studies generally adopt a hybrid architecture in which clinical data remain off-chain, while hashes, references access policies, and audit events are maintained on-chain [3-5, 11]. Although this design improves privacy and scalability, most systems tightly integrate identity management, storage, authorization, and interoperability, making it difficult to isolate and validate the core governance mechanism. Recent reviews also identify performance, governance, and deployment costs as persistent barriers [12, 13]. To address these limitations, this study implements a minimal smart contract that enforces a patient-centric Access Control List (ACL) and logs access events, deliberately avoiding scope creep into storage or interoperability.

3. Research design and system requirements

3.1. Research methodology

We adopt a Design Science/PoC research approach. Rather than developing a production-ready system, our goal is to validate the feasibility of a minimal smart contract as an authorization and audit layer. We first analyze representative blockchain-based healthcare systems (Section 2) to derive key identify design requirements. We then implement a PoC in Solidity (Section 5) and evaluate it against the defined requirements (Section 6).

3.2. Threat model

We consider the following practical threats:

External adversaries: Unauthorized accounts attempting to retrieve record references.

Insider misuse: Authorized users (e.g., hospital staff) attempting unauthorized access.

Authorization Disputes: Disagreements over whether access authorization existed at a specific point in time.

3.3. System requirements

Based on the literature review and threat model, the proposed PoC satisfies the following requirements:

R1—Patient Record Registration: Patients register an on-chain reference (hash/pointer) to their EHR.

R2—Patient-Centric Authentication: Patients explicitly grant access to designated doctors.

R3—Authorized Access: Only authorized doctors can retrieve the EHR reference.

- R4—Event logging: Authorization activities generate blockchain events for external monitoring.
- R5—Audibility: Authorization grants and access attempts are recorded on-chain.
- R6—Off-Chain Clinical Data: Clinical data remain off-chain; only references/hashes are kept.
- R7—Cost Avoidance: The contract minimizes gas consumption.

4. Proposed patient-centric blockchain architecture

4.1. System components

The proposed architecture consists of five core components:

Patient: Owns the EMR and controls access permissions.

Doctor: Requests access to patient records.

Smart Contract (Governance Layer): Stores record hashes, permission mappings, and audit events.

Off-Chain EHR Storage: Stores encrypted clinical data.

Blockchain Network: Executes the smart contract and maintains the immutable ledger.

4.2. Data flow and operational workflow

The system operates through the following workflow:

Step 1. Record Storage (Off-chain): The patient (or healthcare institution) stores the EMR secure off-chain storage and generates a cryptographic hash of the record.

Step 2. Record Registration (On-chain): The patient calls the `setRecord` function to store the record hash on-chain.

Step 3. Authorization Grant (On-chain): The patient calls `grantAccess` to authorize a doctor's Ethereum address. The contract updates the ACL and emits an audit event.

Step 4. Access Request (On-chain): The doctor calls `getRecord` with the patient's address.

Step 5. Access Enforcement (On-chain): The contract checks the ACL. If authorization exists, the record hash is returned; otherwise, the transaction reverts with "Access Denied."

Step 6. Off-chain Retrieval: The doctor retrieves the corresponding clinical record from off-chain storage using the returned reference and appropriate decryption credentials.

The smart contract governs authorization only and does not handle decryption keys or transmit clinical data.

5. Smart contract proof-of-concept implementation

5.1. Contract overview

The PoC is implemented in Solidity ^0.8.0 and contains approximately 30 lines of core logic. The smart contract serves as an authorization and audit layer rather than a clinical data storage system. It maintains EMR references, manages patient-defined permissions, and records access-related events.

5.2. Data structures

The contract uses two primary data structures:

```
mapping(address => string) private patientRecords;  
mapping(address => mapping(address => bool)) public doctorAccess;
```

patientRecords: stores the hash/reference of each patient's EMR. The private mapping prevents direct external access and allows retrieval only through the access-controlled function.

doctorAccess: Maintains a patient-doctor permission relationship. Patients can modify their own permissions through function-level access controls based on msg.sender.

5.3. Core functions

5.3.1. Record registration

```
function setRecord(string memory _recordHash) public {  
    patientRecords[msg.sender] = _recordHash;  
    emit RecordUpdated(msg.sender);  
}
```

The function allows patients to store or update their EMR reference hash. The use of msg.sender ensures that only the record owner can modify their own record, satisfying R1.

5.3.2. Access authorization

```
function grantAccess(address _doctor) public {  
    doctorAccess[msg.sender][_doctor] = true;  
    emit AccessGranted(msg.sender, _doctor);  
}
```

This function enables patients to authorize doctors to access their EMR reference. Only the patient can grant permission through msg.sender, implementing patient-centric authorization and satisfying R2.

5.3.3. Access verification

```
function getRecord(address _patient)  
    public view returns (string memory) {  
    require(  
        doctorAccess[_patient][msg.sender]  
        "Access Denied"  
    );  
    return patientRecords[_patient];  
}
```

This function verifies whether the requesting doctor exists in the patient's ACL. Unauthorized requests are rejected through the require statement. As a read-only function, it does not modify blockchain state and therefore reduces execution cost.

5.4. Security and design considerations

Identity Management: The PoC uses Ethereum addresses as identifiers. In practical healthcare scenarios, additional identity verification mechanisms (e.g., trusted authorities or Decentralized identity frameworks) are required.

Permission Revocation: The current implementation does not include a revokeAccess function. Granted permissions remain active until a revocation mechanism is introduced.

Transparency and Privacy: Although blockchain events are publicly visible, they contain only addresses and hashes, not clinical content, following the hybrid architecture principle.

6. Evaluation and test results

6.1. Functional testing

The PoC was evaluated through six functional test cases corresponding to the defined system requirements (R1-R6). The results in Table 2 demonstrate that all core authorization and audit functions operate as expected.

Table 2. Functional verification results

Requirement	Function Tested	Results
R1	Record registration	Pass
R2	Access authorization	Pass
R3	Authorized retrieval	Pass
R4	Unauthorized rejection	Pass
R5	Audit logging	Pass
R6	Off-chain data storage	Pass

6.2. Security and compliance evaluation

The proposed PoC is further evaluated from security and compliance perspectives. Table 3 summarizes how the smart contract addresses key requirements and the remaining limitations.

Table 3. Security and compliance evaluation of the PoC

Category	Current Implementation	Limitations
Access Control	Patient-centric ACL through smart contract permissions	Limited role-based control and fine-grained policies
Auditability	Blockchain events record authorization activities	Failed access attempts are not fully logged
Data Integrity	Hash-based references ensure record integrity verification	Cannot prevent malicious data sharing; depends on off-chain verification
Transaction Security	Blockchain provides secure transaction execution	Identity authentication mechanisms are out of scope
GDPR Accountability	Transparent and auditable access decisions	Immutable storage challenges the right to erasure

6.3. Gas cost analysis

The PoC uses minimal state variables. Estimated costs:
 setRecord: ~50,000–70,000 gas (depending on string length).
 grantAccess: ~45,000 gas (mapping update).

getRecord: ~2,000 gas (view function, low cost).

While these are feasible for low-frequency use, systematic reviews note that high-frequency workflows (e.g., daily multiple accesses per patient) would render public-chain costs impractical [12]. Permissioned blockchains like Hyperledger Fabric offer alternatives with lower transaction fees but different trust assumptions [14].

Summary: The PoC successfully fulfills R1–R6. It demonstrates that a minimal smart contract can provide deterministic authorization enforcement and auditable governance. However, it falls short of addressing revocation, delegation, break-glass access, and formal security verification, which are critical for production systems.

7. Discussion

7.1. Blockchain as a governance layer, not a panacea

The findings confirm that Ethereum smart contracts are well-suited as a cybersecurity governance layer for EMR sharing. They provide a shared, transparent state that multiple institutions can trust without a central authority. However, blockchain should not be considered a complete replacement for existing healthcare infrastructures. First, interoperability standards such as HL7 FHIR remain essential because smart contract only manage access permissions, while clinical data exchange still relies on standardized data formats and APIs [4, 10, 15]. Second, conventional security mechanisms remain necessary. Identity proofing, key management, secure off-chain storage, and encryption remain mandatory. The contract verifies Ethereum addresses rather than real-world identities. Third, privacy protection requires additional engineering solutions. Although the proposed design avoids storing clinical data on-chain, metadata such as addresses and timestamps may still reveal sensitive information, requiring further privacy-preserving mechanisms [16].

7.2. Public vs. permissioned blockchains

The PoC uses Ethereum (public). Public chains offer openness and censorship resistance but suffer from high latency, variable gas fees, and transparency that conflicts with medical privacy. Permissioned/consortium blockchains (e.g., Hyperledger Fabric, HealthChain) offer higher throughput, controlled membership, and privacy-preserving channels, but require governance agreements and reduce decentralization [14, 17]. Future deployments should weigh these trade-offs contextually.

7.3. Critical governance edge cases

The literature and our analysis highlight several gaps between prototypes and deployable systems [12, 13]:

- Access Revocation: Once granted, access is permanent. A revokeAccess function is essential.

- Time-bounded Permissions: Access should expire (e.g., valid for 30 days).

- Delegation: Patients may need to delegate management to proxies.

- Emergency (Break-Glass) Access: In critical situations, providers must access records without prior consent, with post-hoc auditability.

7.4. Smart contract security risks

As Luu et al. emphasize, smart contracts are software and contain vulnerabilities (e.g., reentrancy, overflows) [18]. Our contract is simple, but real-world extensions must undergo rigorous formal verification and automated analysis. Immutability of deployed contracts means bugs are costly to fix.

8. Conclusion

This paper investigated a minimal Ethereum smart contract can support patient-centric authorization and tamper-evident auditability for EMR sharing. Through a topic analysis of existing blockchain-based healthcare systems and the development of a Solidity-based proof-of-concept, the study demonstrated that blockchain smart contracts can effectively implement transparent access control mechanisms and provide immutable records of authorization activities. The proposed contract successfully realizes core functions including record registration, access permission management, and access event logging, showing the potential of blockchain as a governance layer for improving trust and accountability in EMR sharing.

However, the results also indicate that blockchain should be considered a complementary infrastructure rather than a complete replacement for existing healthcare information systems. Practical deployment still requires integration with interoperability frameworks, stronger identity management mechanisms, privacy protection technologies, and comprehensive regulatory compliance strategies. Although the proposed PoC verifies the feasibility of the core authorization mechanism, challenges related to scalability, privacy preservation, smart contract security, and real-world workflow adaptation remain unresolved.

Future research can further improve the proposed framework by incorporating advanced access management functions, such as permission revocation, temporary authorization, and emergency access mechanisms. In addition, future studies should explore stronger identity verification approaches, enhanced smart contract security validation, and performance optimization across different blockchain architectures to evaluate its feasibility in large-scale healthcare environments.

References

- [1] 45 CFR 164.312 - Technical safeguards. *eCFR*. URL <https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-C/section-164.312>. Accessed 2026-02-24.
- [2] Regulation (EU) 2016/679 (General Data Protection Regulation). *European Union*, 2016. URL <https://www.legislation.gov.uk/eur/2016/679/contents>. Accessed 2026-02-24.
- [3] Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)* (pp. 25-30). IEEE.
- [4] Zhang, P., Schmidt, D. C., White, J., & Lenz, G. (2018). FHIRChain: Applying blockchain to securely and scalably share clinical data. *Computational and Structural Biotechnology Journal*, 16, 267-278.
- [5] Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., & Wang, F. (2020). ACTION-EHR: Patient-centric blockchain-based electronic health record data sharing system. *Journal of Medical Internet Research*, 22(8), e13598.
- [6] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *White paper*.
- [7] Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. White paper.
- [8] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger (Yellow paper). Technical report.
- [9] Szabo, N. (1996). Smart contracts: Building blocks for digital markets. Online manuscript.
- [10] HL7 FHIR overview. HL7 FHIR Specification. URL <https://www.hl7.org/fhir/overview.html>. Accessed 2026-02-24.

- [11] Kuo, T.-T., Kim, H.-E., & Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 24(6), 1211-1220. doi: 10.1093/jamia/ocx068.
- [12] Tandon, A., Dhir, A., Islam, A. K. M. N., & Mantymaki, M. (2020). Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda. *Computers in Industry*, 122, 103290. doi: 10.1016/j.compind.2020.103290.
- [13] Fang, H. S. A., et al. (2021). Blockchain personal health records: Systematic review. *Journal of Medical Internet Research*.
- [14] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., et al. (2018). Hyperledger fabric: A distributed operating system for permissioned blockchains. *arXiv preprint arXiv:1801.10228*.
- [15] Ayaz, M., Pasha, M. F., Alzahrani, M., Budiarto, R., & Stiawan, D. (2021). The fast health interoperability resources (FHIR) standard: Systematic literature review of implementations, applications, challenges and opportunities. *JMIR Medical Informatics*, 9(7), e21929. doi: 10.2196/21929.
- [16] Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *Journal of Medical Systems*, 40(10), 218. doi: 10.1007/s10916-016-0574-6.
- [17] Xiao, Y., Xu, B., Jiang, W., & Wu, Y. (2021). The HealthChain blockchain for electronic health records: Development study. *Journal of Medical Internet Research*, 23(1), e13556. doi: 10.2196/13556.
- [18] Luu, L., Chu, D.-H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM. doi: 10.1145/2976749.2978309.