Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Standardized Blockchain Framework for Scalable Healthcare Data Management and Access Control

Prince Arora

princearorabca@gmail.com

Lovely Professional University

Research Article

Keywords: Blockchain, Electronic Health Records, Access Control, Data Interoperability, Decentralized Identifiers, Verifiable Credentials, Smart Contracts, Multi-Domain Systems, Healthcare Data Management

Posted Date: September 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10465718/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Healthcare organizations increasingly generate data across heterogeneous electronic health record (EHR) systems, laboratory and imaging platforms, and connected medical devices. This fragmentation, combined with strict confidentiality requirements, makes secure and interoperable data exchange one of the persistent open problems in health information technology. This paper proposes a blockchain-based framework for standardized healthcare data management and access control that addresses two coupled challenges: the absence of a common data representation across disparate clinical sources, and the need for fine-grained, verifiable, and auditable access control across organizational boundaries. The proposed model introduces a data standardization layer that normalizes incoming records to a common exchange schema prior to on-chain reference, together with an access-control layer built on decentralized identifiers (DIDs) and verifiable credentials, in which permissions are resolved through smart contracts rather than a centralized policy server. We formalize the subject-object-policy access model, describe the layered system architecture, and analyze how the design mitigates risks associated with centralized access-control models such as DAC, MAC, ABAC, and RBAC, namely policy manipulation, single points of failure, identity and permission forgery, and privacy leakage in multi-domain environments. We further outline the evaluation methodology, including transaction throughput, latency, and scalability under increasing node counts, planned for the implementation phase of this work on a permissioned blockchain platform. As a framework-and-design contribution, this paper does not report empirical performance results; it establishes the architectural and formal basis on which the subsequent implementation and evaluation will be conducted. By strengthening the trustworthiness, interoperability, and auditability of health data exchange, the proposed framework supports Sustainable Development Goal 3 (Good Health and Well-being), by helping health systems deliver safer, better-coordinated care, and Sustainable Development Goal 9 (Industry, Innovation and Infrastructure), by contributing resilient, standardized digital infrastructure for the health sector.

1. Introduction

Healthcare data is generated and stored across many independently managed systems: hospital electronic health record (EHR) platforms, diagnostic laboratories, imaging centers, pharmacies, insurers, and, increasingly, patient-worn monitoring devices. Each of these sources typically adopts its own data model, coding scheme, and access policy, which fragments the overall clinical record and complicates the coordinated care that modern health delivery depends on [1]. When data must move between these systems, whether for a referral, a research collaboration, or a second opinion, two distinct problems arise together. The first is a representation problem: records generated under different schemas cannot be combined or compared without a translation step. The second is a governance problem: each source enforces access control under its own policy, and there is no common mechanism for verifying, across organizational boundaries, that a given requester is entitled to see a given record. These problems are not merely technical: fragmented and poorly governed health data slows diagnosis, referral, and research, and disproportionately affects under-resourced health systems, so resolving them contributes directly to Sustainable Development Goal 3 (Good Health and Well-being) by supporting timely,

coordinated, and safe care, and to Sustainable Development Goal 9 (Industry, Innovation and Infrastructure) by advancing the standardized, resilient digital infrastructure that health systems increasingly depend on.

Blockchain technology has been proposed as a partial answer to the governance problem because it replaces a single trusted intermediary with a distributed ledger validated by consensus among network participants [2]. Each transaction is recorded in a block that references the cryptographic hash of its predecessor, so that altering any previously committed record would require recomputing every subsequent hash, an operation that is computationally impractical on a properly distributed network. This property gives blockchain-based systems a form of tamper evidence that centralized databases do not have by default, and it is this property, rather than blockchain's association with cryptocurrency, that motivates its use in healthcare access control [3].

However, blockchain alone does not solve the representation problem, and most existing blockchain-for-healthcare proposals treat data standardization and access control as separate concerns, addressed by separate systems that must then be integrated after the fact. This paper argues that the two problems are better addressed jointly: an access-control layer is only as reliable as the data model it protects, and a standardization layer is only as trustworthy as the access-control mechanism that governs who may write to or read from it. Accordingly, this paper proposes a single layered framework that combines a data standardization stage with a blockchain-based access-control stage, and describes the formal access model, system architecture, and security rationale for the design.

2. Background

A blockchain is a distributed ledger maintained by a peer-to-peer network of nodes, each holding a copy of the chain, rather than by a single central authority [2], [3]. Three components are central to how it operates. First, cryptography: every participant holds a key pair, using the private key to sign transactions and the public key to allow others to verify that signature. Second, a peer-to-peer network topology, in which no single node is a point of failure, since the ledger is replicated across participants. Third, a consensus mechanism, such as proof-of-work or a permissioned ordering service, by which the network agrees on the validity and order of new transactions before they are committed [3].

In a permissioned setting, such as the one this framework targets, network membership is restricted to known, credentialed organizations (hospitals, laboratories, insurers), and consensus can be achieved through more efficient protocols than public proof-of-work, since Sybil resistance is provided by membership control rather than by computational cost. This is the setting typically used for platforms such as Hyperledger Fabric, which support private channels between subsets of participants and pluggable consensus, and which are consistent with the confidentiality requirements of multi-domain healthcare data sharing [1].

2.1 Formal Ledger Model and Tamper-Evidence

The tamper-evidence property invoked informally above can be stated more precisely. Model the ledger as an ordered sequence of blocks B1, B2, ..., BN, where each block Bi commits to a batch of transactions txi and a timestamp ti through a cryptographic hash chain:

hi = H(hi − 1 ∥ txi ∥ ti), i = 1, ..., N (2)

with h0 a fixed genesis value agreed at network bootstrap and H a collision-resistant hash function. Because block hi is computed over hi − 1, altering any previously committed transaction txj (j < i) changes hj and therefore every hk for k > j, so an adversary who wishes to rewrite history undetected must recompute the entire suffix of the chain and have every replicated copy of the ledger accept the revised suffix through consensus. For a hash function H that is collision-resistant with advantage bounded by ε, the probability that a computationally bounded adversary can produce an alternative block Bi' ≠ Bi with H(Bi') = H(Bi) is bounded by ε, taken to be negligible in H's security parameter. This is the formal basis for the informal claim in Section 1 that altering a committed record is computationally impractical on a properly distributed network; it depends jointly on the collision resistance of H and on no single party controlling a quorum of the consensus mechanism described in Section 6.3 and 6.4.

3. Related Work

Existing research on blockchain-based access control spans several access-control paradigms and several application domains. This section reviews that work in three groups: general and cross-domain access-control models, blockchain-specific electronic health record (EHR) access-control systems, and interoperability-focused frameworks that, like the present paper, combine blockchain with a data standardization layer.

3.1 General and Cross-Domain Access-Control Models

Liu et al. propose a blockchain-based cross-domain authentication management system aimed specifically at Internet of Things devices, addressing how identity can be verified across administrative domains without a shared central authority [1]. Bogaerts et al. introduce entity-based access control, an approach that generalizes traditional subject-object policy models to support more expressive access rules than role- or attribute-based systems typically allow [2]. Ghosh et al. address a related but distinct problem, secure and fair collaboration in Software-as-a-Service environments, via a heuristic role-mapping framework that resolves permission conflicts when users move between differently structured role systems [3]. Mahalle et al. propose a fuzzy trust-based access-control approach for the Internet of Things, which is relevant to healthcare settings that increasingly rely on connected monitoring devices with variable trust levels [4]. Zhang et al. examine distributed, privacy-preserving access control specifically within sensor networks, a setting that shares the resource-constrained, multi-party character of IoT-enabled healthcare monitoring [5]. Ferraiolo et al.'s proposed NIST standard for role-based access control remains the reference point against which newer, decentralized models are typically compared, including the model proposed in this paper [6]. Sandhu et al. formalize role-based administration more

generally [7], and Bertino et al. examine how authorization constraints can be specified and enforced within workflow management systems, a concern directly relevant to clinical workflows that span multiple departments or institutions [8].

3.2 Blockchain-Based Access Control for Electronic Health Records

A substantial body of work applies blockchain specifically to EHR access control. Guo et al. propose a hybrid blockchain-edge architecture for EHR access control that distributes policy enforcement between edge nodes and the chain to reduce latency [9]. Rajput et al. present EACMS, an emergency access-control management system for personal health records that extends normal permission rules to handle time-critical override scenarios [10]. Rouhani and Deters survey the state of the art in blockchain-based access-control systems generally, cataloguing open challenges around scalability and standardization that this paper's framework attempts to address directly [11]. Thwin and Vasupongayya propose a blockchain-based access-control model aimed specifically at preserving privacy in personal health record systems [12]. Nguyen et al. examine blockchain for secure EHR sharing in mobile cloud-based e-health systems, a deployment context that shares this paper's concern with heterogeneous, resource-constrained data sources [13]. Tang et al. propose an efficient authentication scheme purpose-built for blockchain-based EHRs, addressing the identity-verification component of the access-control problem in more depth than most general frameworks [14].

More recent work has moved toward finer-grained and dual-layer designs. Jakhar et al. present a blockchain-based privacy-preserving access-control framework for EHR management that, like the present paper, separates privacy preservation from raw access control as distinct architectural concerns [17]. Peng et al. propose patient-centric, fine-grained access control for electronic medical record sharing using a dual-blockchain design, in which one chain manages identity and the other manages data references, a separation conceptually related to this paper's division between the standardization layer and the blockchain network layer [18]. Kassab et al. take a more critical view, examining under what conditions blockchain is and is not an appropriate solution for EHR management, a caution this paper takes seriously in Section 8's discussion of what the proposed architecture does not eliminate [19].

3.3 Interoperability and Standardization

Closer to the joint standardization-and-access-control position taken in this paper, Zhang et al. propose FHIRChain, which applies blockchain to securely and scalably share clinical data by aligning on-chain references with the HL7 FHIR standard [15]. Azaria et al.'s MedRec system, an early and widely cited proposal, uses blockchain for medical data access and permission management, establishing much of the conceptual vocabulary, patients as owners of access decisions rather than passive subjects, that later EHR-specific frameworks, including the one proposed here, continue to build on [16]. Compared with the present framework, FHIRChain and MedRec integrate standardization and access control within a single chain design; the framework proposed here separates the two concerns into distinct layers

(Section 6) so that either the standardization mapping or the access-control policy can be revised independently, at the cost of an additional interface between them that must itself be secured.

Read together, this body of work establishes three things that motivate the present framework. First, decentralized and cross-domain identity verification [1], [4] is an active and unresolved problem, particularly for IoT-connected healthcare devices. Second, the access-control literature has moved well beyond simple role-based models toward more expressive, policy-driven, and increasingly fine-grained approaches [2], [8], [18]. Third, while several systems combine blockchain with a standardization step such as FHIR [15], [16], comparatively few treat the standardization and access-control layers as independently specified and jointly evaluated components, which is the specific gap this paper's architecture and Section 11 evaluation plan are designed to close. Notably, none of this prior work explicitly frames its contribution in terms of sustainable development, even though decentralized, standardization-aware access control speaks directly to Sustainable Development Goal 3 (Good Health and Well-being) and Sustainable Development Goal 9 (Industry, Innovation and Infrastructure); the present paper makes that connection explicit as part of its motivation.

4. Problem Statement

Centralized access-control models, including Discretionary Access Control (DAC), Mandatory Access Control (MAC), Attribute-Based Access Control (ABAC), and Role-Based Access Control (RBAC), share a structural dependency on centralized policy storage and evaluation. In a multi-domain healthcare setting, this dependency creates four concrete risks. First, manipulation of access-control policies: unauthorized access to the central policy store can expose or alter the rules governing every downstream access decision. Second, a single point of failure: any compromise or outage of the central system affects the entire network of dependent organizations simultaneously. Third, identity and permission forgery: credentials issued and verified through disparate, loosely coordinated systems are more vulnerable to tampering, since there is no shared, tamper-evident record of issuance. Fourth, privacy and resource leakage: as the number of users and data sources grows across domains, so does the attack surface for inadvertent disclosure of sensitive clinical information.

These risks are compounded in healthcare specifically because the record types involved (diagnoses, treatment histories, genetic information) carry both clinical and legal sensitivity, and because the number of distinct organizational domains that must interoperate for coordinated care, hospitals, laboratories, insurers, and increasingly research institutions, is larger than in most other sectors that rely on centralized access control. The problem this paper addresses is therefore twofold: how to represent healthcare data consistently enough to be governed by a single access-control model despite originating in heterogeneous source systems, and how to make that access-control model itself decentralized enough to avoid the four risks above.

5. Objectives and Contributions

This paper makes the following contributions toward the problem stated above:

1. A layered architecture that couples a data standardization stage with a blockchain-based access-control stage, so that the two are designed jointly rather than integrated after the fact.
2. A formal subject-object-policy access model built on decentralized identifiers and verifiable credentials, in which access decisions are resolved by smart contract rather than by a central policy server.
3. An analysis of how this design mitigates the four risks associated with centralized access control (policy manipulation, single points of failure, identity forgery, and privacy leakage), mapped explicitly to specific architectural components.
4. A defined evaluation methodology, covering transaction throughput, latency, and scalability under increasing participant counts, to be executed once the framework is implemented on a permissioned blockchain platform.

Consistent with contribution 4, this paper presents the framework and its formal basis; it does not report implementation results, which are left to a subsequent evaluation phase described in Section 11. Taken together, these contributions are aimed at supporting Sustainable Development Goal 3 (Good Health and Well-being) and Sustainable Development Goal 9 (Industry, Innovation and Infrastructure), by making cross-institutional health data governance more trustworthy, auditable, and standardized.

6. Proposed System Architecture

Figure 1 shows the proposed architecture as five layers. Data originates in heterogeneous source systems (hospital EHR platforms, laboratory and imaging systems, and IoT-connected monitoring devices), each retaining its native format. Before any record is referenced on-chain, it passes through a data standardization and interoperability layer, which performs schema mapping and format normalization into a common exchange model; a canonical clinical data standard such as HL7 FHIR is a natural candidate for this common model, since it is already designed to represent heterogeneous clinical resource types under a shared schema.

Normalized records are then referenced within the blockchain network layer, which comprises three cooperating modules. A DID registry issues and manages decentralized identifiers for every subject in the system (clinicians, patients, devices, and institutions), giving each a persistent, self-sovereign identity that is not controlled by any single domain. A verifiable-credential issuer attaches tamper-evident claims to those identifiers, for example, that a given clinician is licensed and currently employed by a given hospital, in a form that any relying party can cryptographically verify without contacting the issuing hospital directly. Access-control smart contracts encode the permission rules themselves and execute the access decision logic described formally in Section 7.

Requests are resolved by a consensus and policy-evaluation engine, which combines the network's consensus protocol with the smart-contract logic to reach a single, auditable access decision, before an authorized stakeholder (a clinician, researcher, patient, or insurer, depending on the resolved policy) is

granted access to the requested resource. Every stage of this pipeline, standardization, identity verification, credential verification, and policy evaluation, produces an entry on the distributed ledger, so the full chain of custody for any access decision is independently auditable after the fact.

6.1 Data Standardization Pipeline

Figure 2 details the standardization stage introduced above. An incoming record is first parsed to extract its fields regardless of source format, then mapped onto a canonical clinical data model; HL7 FHIR is a natural choice here since it already defines resource types (Patient, Observation, Condition, and related resources) that most source EHR, laboratory, and device systems can be mapped to with a bounded, source-specific transformation rather than a bespoke mapping per pair of systems, consistent with how FHIR-based approaches have been used elsewhere to align blockchain references with clinical data structure [15]. After validation, the framework does not place the full clinical record on-chain. Consistent with standard practice in blockchain-EHR designs, the normalized record is stored off-chain in an encrypted repository, and only a cryptographic hash reference, together with the access-control metadata needed to evaluate Eq. (1), is committed to the ledger. This keeps the ledger itself free of protected health information while still allowing any party with a valid access decision to retrieve and verify the corresponding off-chain record.

The standardization step itself can be stated as a mapping between schemas. For each source system k (an EHR platform, a laboratory information system, a device gateway), let Rk denote its native record space and let F denote the target canonical field space defined by the exchange schema (FHIR resource types in the present design). The standardization stage implements a per-source mapping

$$\varphi_k : R_k \to F,\ \varphi_k(r) = r' \text{ such that } \mathrm{Fields}(r') \subseteq F \wedge \mathrm{Required}(\mathrm{class}(r)) \subseteq \mathrm{Fields}(r') \quad (3)$$

where Required(class(r)) is the set of fields the exchange schema mandates for the resource class of r (for example, a FHIR Observation resource requires a subject reference, a code, and an effective time). A record is referenced on-chain only once φk(r) passes schema validation, i.e., once Required(class(r)) ⊆ Fields(φk(r)) holds; records that fail this check are rejected before reaching the access-control layer rather than admitted with missing fields the policy engine could not reliably evaluate. Eq. (3) formalizes, without yet implementing, the field-by-field mapping specification that Section 11.5 identifies as necessary future work for each source system the prototype will target.

6.2 Sequence of an Access-Control Decision

Figure 3 traces a single access request through the architecture end to end. A subject first requests access to an object; the DID registry resolves and verifies the subject's decentralized identifier; the credential issuer verifies that the subject's credential claims are sufficient for the requested object class; the access-control smart contract evaluates Eq. (1) and submits the resulting decision for consensus; the consensus engine commits the decision; and the outcome, grant or deny, together with its full verification trail, is written to the ledger before being returned to the subject. Because each intermediate

step (identity verification, credential verification, and policy evaluation) is itself logged rather than only the final decision, an auditor can later reconstruct not just what was decided but why.

6.3 Consensus Mechanism Considerations

The choice of consensus protocol affects both the throughput and the trust assumptions of the framework, and is left as a configuration decision rather than fixed in the architecture, since the appropriate choice depends on deployment scale. In a permissioned setting with a small consortium of trusted institutional nodes, a crash-fault-tolerant protocol such as Raft is typically sufficient and offers higher throughput than protocols designed to tolerate malicious nodes, since institutional participants (hospitals, laboratories, regulators) are identity-verified at network admission rather than anonymous. Where the consortium includes participants with a weaker trust relationship, for example research partners outside the core institutional network, a Byzantine-fault-tolerant protocol trades some throughput for resilience against a participant that behaves maliciously rather than merely failing. This tradeoff between throughput and fault-tolerance strength is deployment-specific and is one of the parameters the evaluation methodology in Section 11 is designed to measure directly rather than assume.

6.4 System Model: Fault Tolerance, Latency, and Complexity

The consensus tradeoff described qualitatively above can be stated as a bound on the number of participating nodes required to tolerate a given number of faults. Let n denote the number of ordering/consensus nodes and f the number of nodes the protocol must tolerate as faulty. For a crash-fault-tolerant protocol such as Raft, which assumes faulty nodes simply stop responding rather than behave arbitrarily, safety requires

$$n \geq 2f + 1 \quad \text{(Crash Fault Tolerance)} \quad (4)$$

For a Byzantine-fault-tolerant protocol, which must additionally tolerate nodes that respond arbitrarily or maliciously, the corresponding requirement is stricter:

$$n \geq 3f + 1 \quad \text{(Byzantine Fault Tolerance)} \quad (5)$$

Equations (4)–(5) formalize why the CFT/BFT choice in Section 6.3 is a genuine tradeoff rather than a matter of preference: for a fixed consortium size n, a BFT protocol tolerates at most $\lfloor (n-1)/3 \rfloor$ faulty nodes against $\lfloor (n-1)/2 \rfloor$ for a CFT protocol, at the cost of the additional communication rounds BFT protocols require to reach agreement in the presence of conflicting messages.

Decomposing end-to-end latency along the stage boundaries traced in Fig. 3, the expected time to resolve a single access decision is

$$E[T_{Grant}(s,o)] = T_{std} + T_{DID} + T_{VC} + T_{policy} + T_{consensus} \quad (6)$$

where each term is the expected latency of the correspondingly named stage. Because VerifyDID and VerifyCredential in Eq. (1) are logically independent predicates, TDID and TVC can in principle be evaluated in parallel rather than sequentially, in which case Eq. (6) reduces to E[TGrant(s,o)] = Tstd + max(TDID, TVC) + Tpolicy + Tconsensus; which of the two forms applies is an implementation choice for the Section 11.2 prototype rather than a property fixed by the architecture. Table 6's finding that credential-verification latency is itself load-sensitive (Section 11.1) means TVC in Eq. (6) should be treated as a load-dependent random variable rather than a constant during the evaluation planned in Section 11.3.

The consensus term Tconsensus in turn depends on the underlying commit pipeline. For an execute-order-validate architecture of the kind Hyperledger Fabric implements, sustained throughput can be approximated as

TPS ≈ B / (Tendorse + Torder + Tvalidate) (7)

where B is the number of transactions batched per block and Tendorse, Torder, and Tvalidate are the mean per-block times spent in endorsement, ordering, and validation/commit respectively. Eq. (7) is consistent with the pattern in Table 4: configurations that increase B or reduce per-stage latency (larger batches, tuned channel configuration) report higher throughput, while single-channel baseline configurations, which bound B more conservatively, report lower throughput. As the number of participating organizations n grows, Tendorse and Torder both increase because more organizations must co-sign and propagate each block, which is the structural reason the throughput figures in Table 5 fall as consortium size grows; the precise functional form of that decline is protocol- and deployment-specific and is left as an empirical question for Section 11.3 rather than asserted here.

Finally, because only a hash reference and fixed-size access-control metadata are committed on-chain per record (Section 6.1), ledger growth is bounded independently of raw record size:

|Ledger| = O(N · c), c constant (8)

where N is the number of standardized records referenced on-chain and c is the fixed per-record footprint (hash digest plus metadata) rather than the size of the underlying clinical record |r|, which is stored off-chain (Section 6.1). This bounds both the storage growth of the ledger and, since audit and traversal operations scan on-chain entries rather than off-chain payloads, the growth of the per-audit query cost referenced in Section 6.2, at O(N) rather than O(Σ|ri|).

7. Access Control Model

Let S = {s1, s2, ..., sn} denote the set of subjects in the system (users and devices), O = {o1, o2, ..., om} the set of objects (standardized clinical records or resources), and P the policy set governing which subjects may access which objects under which conditions. An access decision for a request (s, o) is defined as a function of three verification stages, given in Eq. (1).

$$Grant(s, o) = VerifyDID(s) \wedge VerifyCredential(s, o) \wedge EvaluatePolicy(s, o, P) \quad (1)$$

VerifyDID(s) confirms that the requesting subject's decentralized identifier resolves to a valid, non-revoked entry in the DID registry. VerifyCredential(s, o) confirms that the subject holds a verifiable credential whose claims are sufficient for the class of object o (for example, a treating-physician credential for a given patient's record, or a research-access credential for a de-identified dataset). EvaluatePolicy(s, o, P) applies the policy set, which may encode role, purpose-of-use, consent status, and jurisdictional constraints, and returns true only if every applicable rule is satisfied. Because all three predicates are evaluated within a smart contract executed as part of the network's consensus process, the decision itself, not merely its outcome, is committed to the ledger, which is what distinguishes this model from centralized ABAC or RBAC systems where policy evaluation occurs outside any tamper-evident record.

This structure also allows roles to shift dynamically, consistent with the subject-object interplay described in Section 4: a physician (subject) requesting a lab result (object) may, in a research-sharing workflow, become the object of a subsequent access request from a research coordinator. Because Grant(s, o) is evaluated fresh for each request rather than cached from a prior session, this dynamic role-switching does not require a separate mechanism.

Equation (1)'s three predicates can each be stated more precisely in terms of the underlying cryptographic primitives. Let δ denote the DID registry's resolution function, mapping a decentralized identifier to its currently bound public key and revocation status, $\delta(dids) = (pks, statuss)$. Then

$$VerifyDID(s) = 1 \text{ iff } \delta(dids) = (pks, statuss) \wedge statuss = active \quad (9)$$

Credential verification is likewise reducible to signature verification under the issuing authority's key, together with a claims-sufficiency and non-revocation check:

$$VerifyCredential(s,o) = 1 \text{ iff } Verify(pk_{issuer}, creds, \sigma s) = 1 \wedge Claims(creds) \supseteq Requirements(o) \wedge \neg Revoked(creds) \quad (10)$$

where Verify(·) is the verification algorithm of the credential issuer's digital-signature scheme, Claims(creds) is the set of attribute assertions the credential carries, and Requirements(o) is the minimum claim set the object class of o demands (Section 6.1's schema mapping determines o's class). Finally, EvaluatePolicy resolves as a conjunction over the individual policy predicates in P rather than as a monolithic check:

$$EvaluatePolicy(s,o,P) = \bigwedge k = 1..K \; pk(s,o), pk \in P \quad (11)$$

which is why Table 1's worked example can attribute a denial to a specific failed predicate pk (purpose-of-use, department match, active-relationship) rather than to an undifferentiated policy failure, and why the auditability claim in Section 6.2 extends to identifying which individual predicate, not merely which overall decision, determined an outcome.

7.1 Worked Example

To make Eq. (1) concrete, consider a single illustrative request. This is a worked example of the decision logic, not a measured system output. A clinician subject s holds a DID that resolves to a non-revoked registry entry (VerifyDID(s) = 1) and a verifiable credential asserting "attending physician, Cardiology Department, Hospital A," valid through a stated expiry date. The requested object o is a cardiology lab result belonging to a patient currently admitted under that clinician's care. The applicable policy set P requires: (i) a credential claim matching the object's clinical department, (ii) an active treatment relationship between subject and patient, and (iii) purpose-of-use recorded as "treatment." Table 1 traces the evaluation.

Table 1
Worked evaluation of Eq. (1) for an illustrative treatment-context request (hypothetical values, not measured data).

Predicate	Evaluation	Result
VerifyDID(s)	DID resolves; registry entry not revoked	1
VerifyCredential(s,o)	Credential department (Cardiology) matches object's clinical department; credential unexpired	1
EvaluatePolicy(s,o,P)	Active treatment relationship confirmed; purpose-of-use = treatment; all three policy conditions satisfied	1
Grant(s,o)	Logical AND of the three predicates above	1 (access granted)

Now vary a single condition: if the same clinician instead requested a mental-health record for a patient outside their department and without an active treatment relationship, EvaluatePolicy(s,o,P) would evaluate to 0 regardless of a valid DID and credential, and Grant(s,o) would resolve to 0 (access denied), with the denial itself, including which policy condition failed, recorded on the ledger per Section 6.2. This second case illustrates why the model in Section 8 characterizes policy manipulation risk as mitigated: changing the outcome of this second request would require altering replicated smart-contract logic across consensus nodes, not a single centralized record.

7.2 Formal Security Argument

Equations (9)–(10) make it possible to state the unforgeability claim implicit in Section 8's mitigation of identity and permission forgery as a standard cryptographic reduction rather than an architectural assertion. Assume the credential issuer's signature scheme is existentially unforgeable under adaptive chosen-message attack (EUF-CMA), the standard security notion for signature schemes used in W3C verifiable-credential deployments. Then for any probabilistic polynomial-time adversary A that does not hold the issuer's private key,

$$Adv_{A}^{EUF\text{-}CMA}(\lambda) \leq \varepsilon(\lambda) \quad (12)$$

where ε(λ) is a negligible function of the security parameter λ, meaning the probability that A produces a credential passing Verify(pkissuer, ·, ·) = 1 for claims it was never issued is asymptotically vanishing. Combined with the collision-resistance bound on the ledger's hash function established in Eq. (2), the overall probability that Grant(s,o) is manipulated, either by forging a credential that satisfies Eq. (10) or by retroactively altering a committed decision that satisfies Eq. (2), is bounded by the sum of two independently negligible advantages. This bound formalizes, rather than replaces, the qualitative mitigation claim in Table 2: it holds only insofar as the issuer's signing key remains uncompromised and the underlying hash and signature primitives are correctly instantiated, both operational assumptions the security evaluation in Section 11.4 is designed to probe rather than guarantees the architecture can provide on its own.

8. Mitigation of Centralized Access-Control Risks

Section 4 identified four risks inherent to centralized access-control models. Table 2 maps each risk to the architectural component intended to mitigate it.

Centralized-Model Risk	Mitigating Component	Mechanism
Policy manipulation	Access-control smart contracts	Policy logic executes identically across all consensus nodes; a unilateral change by one party cannot alter a committed decision.
Single point of failure	Distributed ledger / peer-to-peer network	The ledger is replicated across participating nodes, so compromise of one node does not compromise policy evaluation network-wide.
Identity and permission forgery	DID registry and verifiable credentials	Credentials are cryptographically signed and independently verifiable, removing reliance on a single issuing authority's availability.
Privacy and resource leakage	Data standardization layer and purpose-bound policy evaluation	Normalization to a common schema allows purpose-of-use constraints to be applied consistently, and only resolved decisions, not raw records, are recorded on-chain.

It is worth noting explicitly what this architecture does not eliminate: consensus latency introduces a floor on transaction throughput that centralized databases do not have, and the standardization layer itself becomes a component whose correctness the rest of the system depends on. Both trade-offs are the subject of the evaluation methodology in Section 11.

9. Qualitative Comparison with Existing Access-Control Paradigms

Figure 4 contrasts the structural difference between centralized access control and the proposed model at the level of where policy decisions are stored and evaluated. In the centralized case, every user's request is routed through a single policy server backed by a single database; the server is both the sole source of truth and the sole point of failure. In the proposed model, policy logic is replicated as a smart

contract across participating nodes, and a decision is only committed once consensus is reached, so no single node's compromise or outage determines the outcome for the network as a whole.

Table 3
extends this comparison across the four established access-control paradigms reviewed in Section 3.1, based on the properties reported for each in the cited literature, alongside the properties targeted by the proposed model. This is a qualitative comparison of architectural properties, not a benchmarked performance comparison; the latter is deferred to the evaluation phase in Section 11.

Model	Policy Locus	Single Point of Failure	Cross-Domain Support	Identity Verification
DAC [6]	Object owner, per-resource	Yes (owner record)	Limited	Owner-defined, not independently verifiable
MAC [6]	Central security authority	Yes (central authority)	Limited	Centrally issued clearances
RBAC [6], [7]	Central role administrator	Yes (role store)	Requires role mapping across domains	Centrally assigned roles
ABAC / Entity-based [2]	Central policy engine	Yes (policy engine)	Requires shared attribute schema	Centrally managed attributes
Proposed model	Smart contract, replicated	No (replicated across nodes)	Native via shared ledger and DIDs	Cryptographically verifiable credentials

The comparison highlights a consistent pattern: DAC, MAC, RBAC, and ABAC all locate both policy storage and identity verification in a single administrative point, whether that point is an object owner, a security authority, or a policy engine [6], [7], [2]. The proposed model's principal structural difference is not any single feature but the removal of that single point across both dimensions simultaneously, which is also why Section 8 treats consensus latency and standardization-layer correctness, not any single component failure, as the framework's residual risks.

10. Planned Evaluation Methodology

This paper has presented a framework and its formal basis rather than an implemented and benchmarked system. No performance numbers from this system are reported in this paper for that reason; the evaluation phase planned for this work will proceed in five stages, described below, the first of which situates the framework's expected performance within figures already established in the published literature.

10.1 Expected Performance Envelope from Prior Literature

Although this framework has not itself been benchmarked, Hyperledger Fabric, the platform identified in Section 11.2 as the implementation target, has been benchmarked extensively by others. Table 4

summarizes representative throughput figures reported in prior peer-reviewed and vendor-published studies, to give a literature-grounded sense of the order of magnitude this framework's prototype might expect, not a prediction specific to this system.

Equation (7) in Section 6.4 gives an analytical reason to expect exactly this ordering: throughput scales with batch size B and inversely with the sum of per-stage commit latencies, so configurations that differ mainly in channel count, batch size, or asset payload size, as the studies in Table 4 do, should be expected to differ in reported throughput even when running the same underlying protocol on comparable hardware.

Table 4

Representative Hyperledger Fabric throughput benchmarks reported in prior literature (not results from this paper's framework).

Study	Configuration	Reported Peak Throughput	Latency
Androulaki et al. [20]	Fabric v1.x, UTXO-style workload	~ 3,500 TPS	Sub-second
Thakkar et al. [21]	Single channel, baseline config	~ 140 TPS (saturation)	Not specified
Thakkar et al. [21]	16 channels, tuned config	~ 850 TPS	Not specified
Kelsey / Hyperledger Fdn. [22]	Fabric 2.5, 2 peer orgs, Raft, single channel, 100-byte asset	~ 3,000 TPS (blind write)	Low, non-saturating
Kelsey / Hyperledger Fdn. [22]	Same config, read/write, 1000-byte asset	~ 1,530 TPS	Low, non-saturating
Zulkarnain et al. [23]	Heterogeneous SBC + laptop nodes, IoT-oriented	~ 1,148 TPS (high-perf. nodes)	Not specified

More directly relevant to this paper's multi-domain healthcare setting, Barbaria et al. develop a theoretical performance model for a Fabric-based healthcare consortium and project throughput as a function of the number of participating organizations, based on established Fabric scaling characteristics [24]. Table 5 reproduces their projections, which are of the same character as, and a useful literature anchor for, the scalability analysis this paper's own prototype will need to conduct under Section 11.3.

Table 5

Throughput projected against consortium size for a Fabric-based healthcare data-sharing network, as reported by Barbaria et al. [24] *(their projection, not a result of this paper).*

Consortium Size	Projected Throughput [24]	Illustrative Use Case
10 organizations	150–300 TPS	Regional research consortium
50 organizations	80–180 TPS	Statewide collaboration
100 organizations	30–100 TPS	International research data aggregation

Tables 4 and 5 characterize transaction throughput at the consensus layer, but the architecture in Fig. 3 also depends on identity and credential verification (steps 2–3), which prior literature has benchmarked separately from raw ledger throughput. Table 6 summarizes two such studies.

Table 6

Reported decentralized-identity and verifiable-credential verification latency in prior literature (not results from this paper's framework).

Study	Measurement	Reported Result
Prajapati and Rahman [25]	p95 credential verification latency, decentralized biometric identity framework, concurrent multi-node conditions	~ 820 ms
Dunphy [26]	Trust Registry read/write transaction latency, Hyperledger Indy, empirical study of 45,000 + transactions on AWS	Latency variable and load-sensitive; write and revocation operations disproportionately slower under ambient load

Table 6 is a caution as much as a data point. Dunphy's finding that revocation and write operations are disproportionately slower under ambient transaction load [26] directly informs the limitation noted in Section 11.5: VerifyDID and VerifyCredential are treated as atomic, individually correct predicates in Eq. (1), but Table 6 indicates their real-world latency, and by extension their contribution to end-to-end Grant(s,o) resolution time in Fig. 3, is neither constant nor negligible under concurrent load. The 820 ms p95 figure from Prajapati and Rahman [25], while measured on a biometric-specific identity framework rather than this paper's DID/VC design, is a useful literature-grounded upper bound for what "sub-second" credential verification means in practice for the evaluation planned in Section 11.3.

Two further observations follow from Tables 4 and 5. First, throughput and consortium size trade off inversely across every study in Tables 4 and 5, which is consistent with this paper's own Section 6.3 discussion of consensus overhead scaling with node count; any prototype of the present framework should expect the same qualitative trend, with the specific numbers depending on endorsement policy, channel configuration, and hardware, as Table 4 illustrates. Second, the 10-organization projection in Table 5 (150–300 TPS) is the most directly comparable literature figure to this framework's likely initial deployment scale (a small consortium of hospitals, laboratories, and a research partner, as illustrated in

Fig. 1), and is therefore the figure the evaluation in Section 11.3 will treat as the closest published benchmark to validate against, not merely the largest reported number in Table 4.

10.2 Prototype Implementation

The framework will be implemented on a permissioned blockchain platform, with Hyperledger Fabric identified as the primary candidate given its support for private channels between subsets of participating organizations and its pluggable consensus mechanism, both of which align with the multi-domain confidentiality requirements described in Section 4 and with prior Fabric-based healthcare deployments reviewed in Section 3 [9], [17]. Access-control logic will be implemented as chaincode, and the DID registry and credential issuance flow will follow the W3C decentralized identifier and verifiable credentials specifications to allow interoperability with identity tooling outside the prototype network.

10.3 Performance Metrics

The prototype will be evaluated against three categories of metric. Transaction throughput, measured in transactions per second, under representative access-request loads, benchmarked against the Table 4 and Table 5 figures above. Latency, measured as end-to-end time from access request to resolved decision, decomposed into standardization, credential verification, and consensus components (following the stage boundaries in Fig. 3) so that the source of any bottleneck can be identified. Scalability, measured as throughput and latency as the number of participating organizational nodes increases, since a design that performs well at small scale but degrades sharply as domains are added would not meet the framework's stated goal of multi-domain scalability.

11. Conclusion

This paper has proposed a layered framework for healthcare data management that treats data standardization and blockchain-based access control as a single, jointly designed problem rather than two separately integrated systems. The architecture combines a standardization layer, intended to reconcile the heterogeneous formats in which clinical data originates, with a decentralized identifier and verifiable-credential-based access-control layer, in which permission decisions are resolved by smart contract and recorded on a distributed ledger rather than evaluated by a central policy server. We have formalized the resulting access model, mapped its architectural components to the specific risks associated with centralized access control, and outlined an evaluation methodology covering throughput, latency, scalability, and security testing against a centralized baseline. The framework's practical value depends on that evaluation, which is the immediate next stage of this work; the contribution of this paper is the architectural and formal foundation on which it will be conducted. More broadly, the proposed framework is intended as a contribution toward Sustainable Development Goal 3 (Good Health and Well-being), by making cross-institutional health data exchange more trustworthy and auditable in support of timely and coordinated care, and toward Sustainable Development Goal 9 (Industry, Innovation and Infrastructure), by proposing standardized, resilient digital infrastructure that health systems can build on. Realizing this potential, however, depends on the implementation and

evaluation work outlined above, including validation in resource-constrained health system settings where the sustainable development impact is likely to be greatest.

Declarations

Consent to Publish declaration

Not applicable. This manuscript does not contain any individual person's data, images, or other identifiable personal information in any form.

Funding declaration

No funding was received for the conduct of this research or the preparation of this manuscript.

Author Contribution

P.A. is the sole author of this manuscript and is responsible for all aspects of the work. P.A. conceived the study, developed the theoretical framework and system architecture, conducted the literature review and comparative analysis, formalized the access-control model and system equations, designed the evaluation methodology, and wrote and revised the manuscript. The author read and approved the final manuscript.

Data Availability Statement

No datasets were generated or analysed during the current study, as this manuscript presents a conceptual architecture and formal access-control model rather than an empirical implementation; data generation and analysis are planned for the subsequent evaluation phase described in Section 10.

Ethics and Consent to Participate declarations

Not applicable. This manuscript presents a conceptual architecture and formal model and did not involve human participants, patient data, or animal subjects, so no ethics approval or consent to participate was required.

References

1. Liu Y, Liu A, Xia Y, Hu B, Liu J, Wu Q, Tiwari P. A Blockchain-Based Cross-Domain Authentication Management System for IoT Devices, DiVA Portal, 2024. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2

2. Bogaerts J, Decat M, Lagaisse B. Entity-Based Access Control: Supporting More Expressive Access Control Policies, in Proc. ACM Symp. Access Control Models and Technologies, 2015, 10.1145/2818000.2818009
3. Ghosh N, Chatterjee D, Ghosh SK. An Efficient Heuristic-Based Role Mapping Framework for Secure and Fair Collaboration in SaaS, 2014.
4. Mahalle PN, Thakre PA, Prasad NR, Prasad R. A Fuzzy Approach to Trust Based Access Control in Internet of Things, 2013.
5. Zhang R, Zhang Y, Ren K. Distributed Privacy-Preserving Access Control in Sensor Networks, 2012.
6. Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R. Proposed NIST Standard for Role-Based Access Control. ACM Trans Inf Syst Secur. 2001. 10.1145/501978.501980.
7. Sandhu RS, Bhamidipati V, Munawer Q. The ARBAC97 Model for Role-Based Administration of Roles. ACM Trans Inf Syst Secur. 1999;2(1):105–35. 10.1145/300830.300839.
8. Bertino E, Ferrari E, Atluri V. The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. ACM Trans Inf Syst Secur. 1999. 10.1145/300830.300837.
9. Guo H, Li W, Nejad M, Shen C-C. Access Control for Electronic Health Records with Hybrid Blockchain-Edge Architecture, in Proc. 2019 IEEE Int. Conf. Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 44–51. 10.1109/Blockchain.2019.00015
10. Rajput AR, Li Q, Ahvanooey MT, Masood I. EACMS: Emergency Access Control Management System for Personal Health Record Based on Blockchain, IEEE Access, vol. 7, pp. 84304–84317, 2019.
11. Rouhani S, Deters R. Blockchain Based Access Control Systems: State of the Art and Challenges, in Proc. IEEE/WIC/ACM Int. Conf. Web Intelligence (WI '19), Thessaloniki, Greece, 2019, pp. 423–428. 10.1145/3350546.3352561
12. Thwin TT, Vasupongayya S. Blockchain-Based Access Control Model to Preserve Privacy for Personal Health Record Systems, Security and Communication Networks, vol. 2019, Art. no. 8315614, 2019.
13. Nguyen DC, Pathirana PN, Ding M, Seneviratne A. Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems, IEEE Access, vol. 7, pp. 66792–66806, 2019.
14. Tang F, Ma S, Xiang Y, Lin C. An Efficient Authentication Scheme for Blockchain-Based Electronic Health Records. IEEE Access. 2019;7:41678–89. 10.1109/ACCESS.2019.2904300.
15. Zhang P, White J, Schmidt DC, Lenz G, Rosenbloom ST. Comput Struct Biotechnol J. 2018;16:267–78. FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data,.
16. Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: Using Blockchain for Medical Data Access and Permission Management, in Proc. 2016 2nd Int. Conf. Open and Big Data (OBD), Vienna, Austria, 2016, pp. 25–30.
17. Jakhar AK, Singh M, Sharma R, Viriyasitavat W, Dhiman G, Goel S. A Blockchain-Based Privacy-Preserving and Access-Control Framework for Electronic Health Records Management. Multimedia Tools Appl. 2024. 10.1007/s11042-024-18827-3.

18. Peng G, Zhang A, Lin X. Patient-Centric Fine-Grained Access Control for Electronic Medical Record Sharing with Security via Dual-Blockchain. IEEE Trans Netw Sci Eng. 2023;10(6):3908–21.
19. Kassab M, DeFranco J, Malas T, Graciano Neto VV, Destefanis G. Blockchain: A Panacea for Electronic Health Records? in Proc. 2019 IEEE/ACM 1st Int. Workshop Software Engineering for Healthcare (SEH), Montreal, QC, Canada, 2019.
20. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, Enyeart D, Ferris C, Laventman G, Manevich Y, Muralidharan S, Murthy C, Nguyen B, Sethi M, Singh G, Smith K, Sorniotti A, Stathakopoulou C, Vukolić M, Cocco SW, Yellick J. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains, in Proc. 13th EuroSys Conf. (EuroSys '18), Porto, Portugal, 2018, Art. no. 30. 10.1145/3190508.3190538
21. Thakkar P, Nathan S, Viswanathan B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform, in Proc. 2018 IEEE 26th Int. Symp. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 2018, pp. 264–276. 10.1109/MASCOTS.2018.00034
22. Kelsey D. Benchmarking Hyperledger Fabric 2.5 Performance, LF Decentralized Trust / Hyperledger Foundation Blog, Feb. 16, 2023. [Online]. Available: https://www.lfdecentralizedtrust.org/blog/2023/02/16/benchmarking-hyperledger-fabric-2-5-performance
23. Zulkarnain MM, Ramli N, Nordin AN. Performance Benchmarking of Hyperledger Fabric on Heterogeneous Hardware for IoT Applications. IIUM Eng J. 2025;26(3):156–70. 10.31436/iiumej.v26i3.3610.
24. Barbaria S, Jemai A, Ceylan Hİ, Muntean RI, Dergaa I, Boussi Rahmouni H. Advancing Compliance with HIPAA and GDPR in Healthcare: A Blockchain-Based Strategy for Secure Data Exchange in Clinical Research Involving Private Health Information, Healthcare. Art no 2594. 2025;13(20). 10.3390/healthcare13202594.
25. Prajapati AK, Memon S, Rahman MM, Al-Nemrat A. Ciphera: A Decentralised Biometric Identity Framework, arXiv:2605.29868, 2026.
26. Dunphy P. A Note on the Blockchain Trilemma for Decentralized Identity: Learning from Experiments with Hyperledger Indy, arXiv:2204.05784, 2022.

Figures

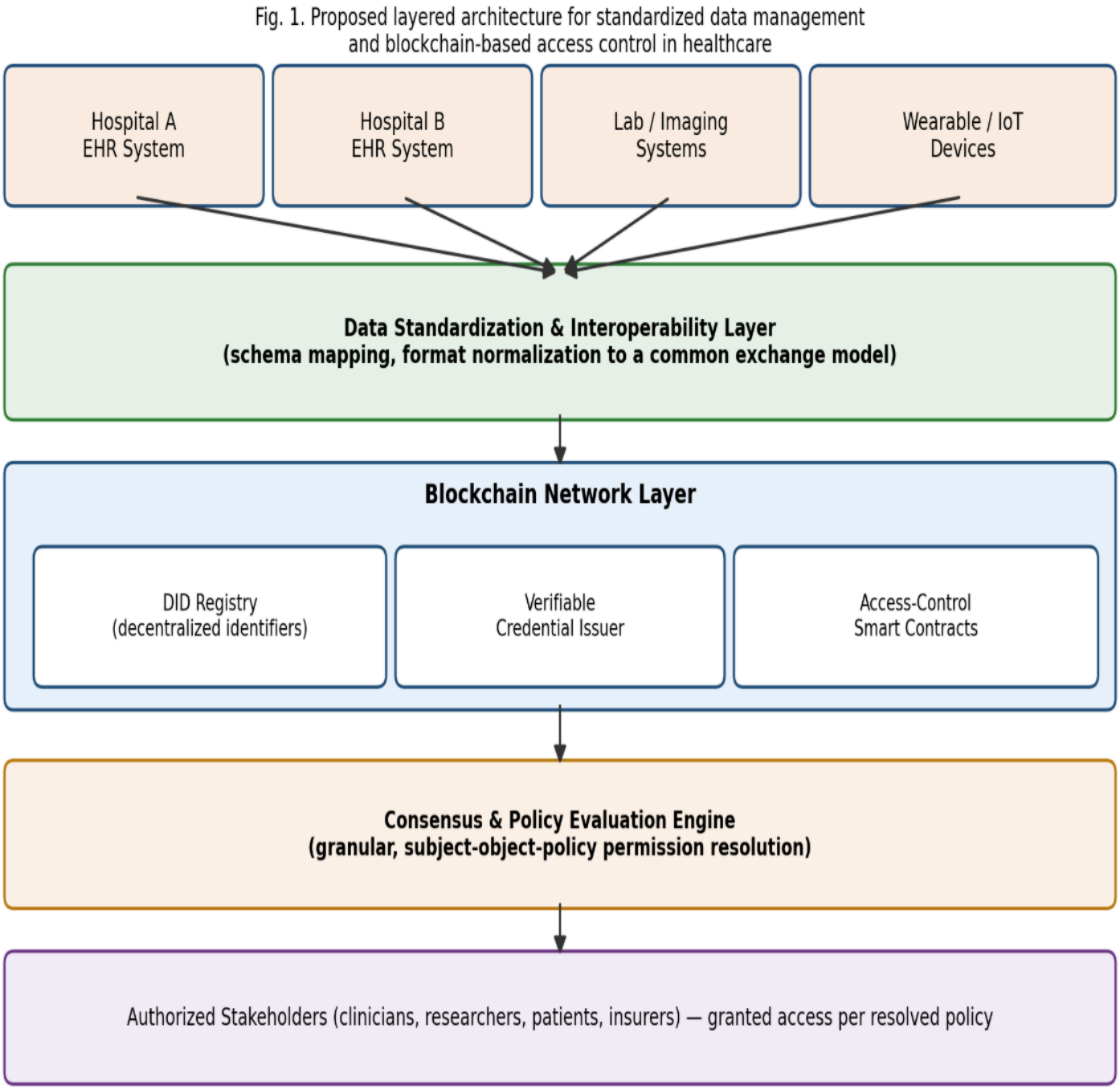


Figure 1

See image above for figure legend.

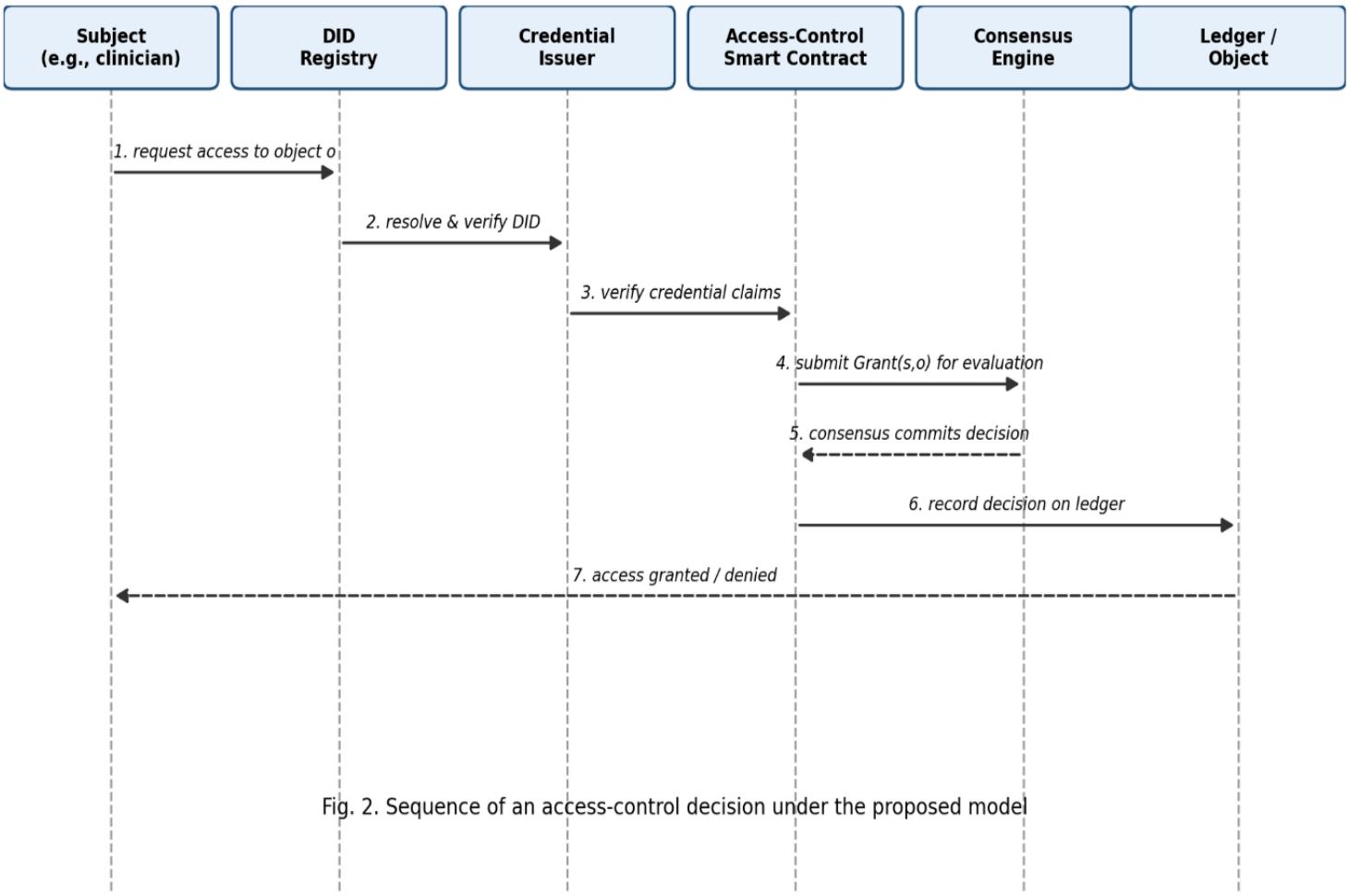


Fig. 2. Sequence of an access-control decision under the proposed model

Figure 2

See image above for figure legend.

Fig. 3. Data standardization pipeline preceding on-chain reference

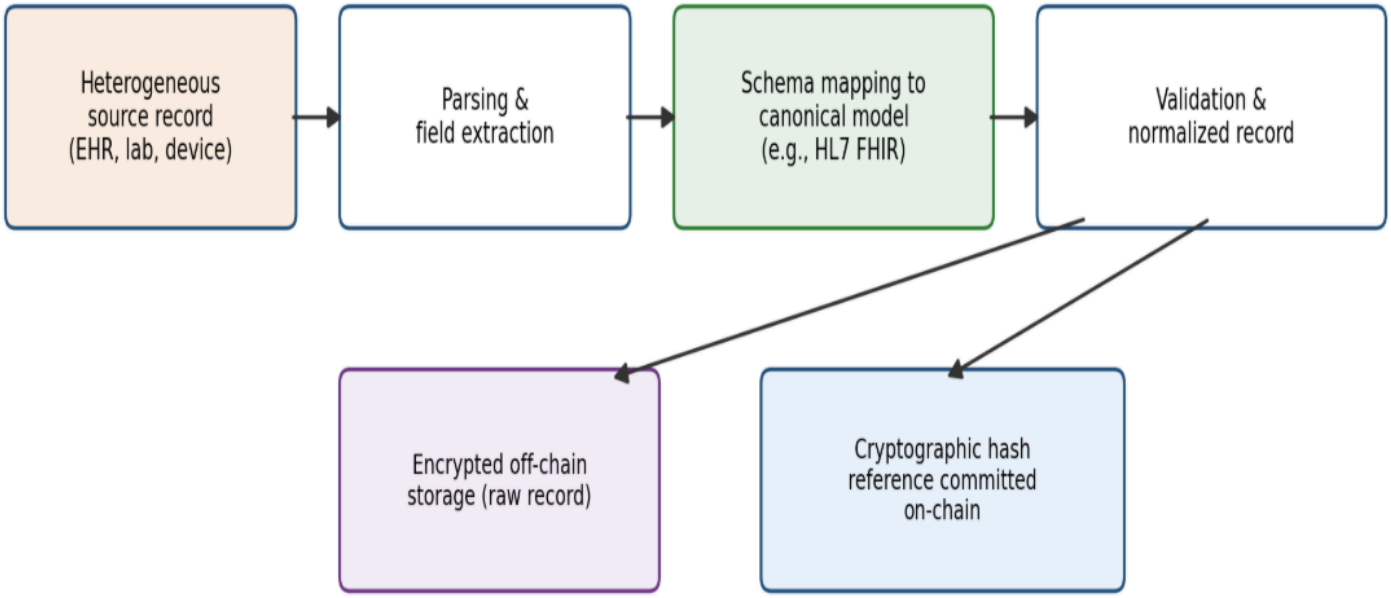


Figure 3

See image above for figure legend.

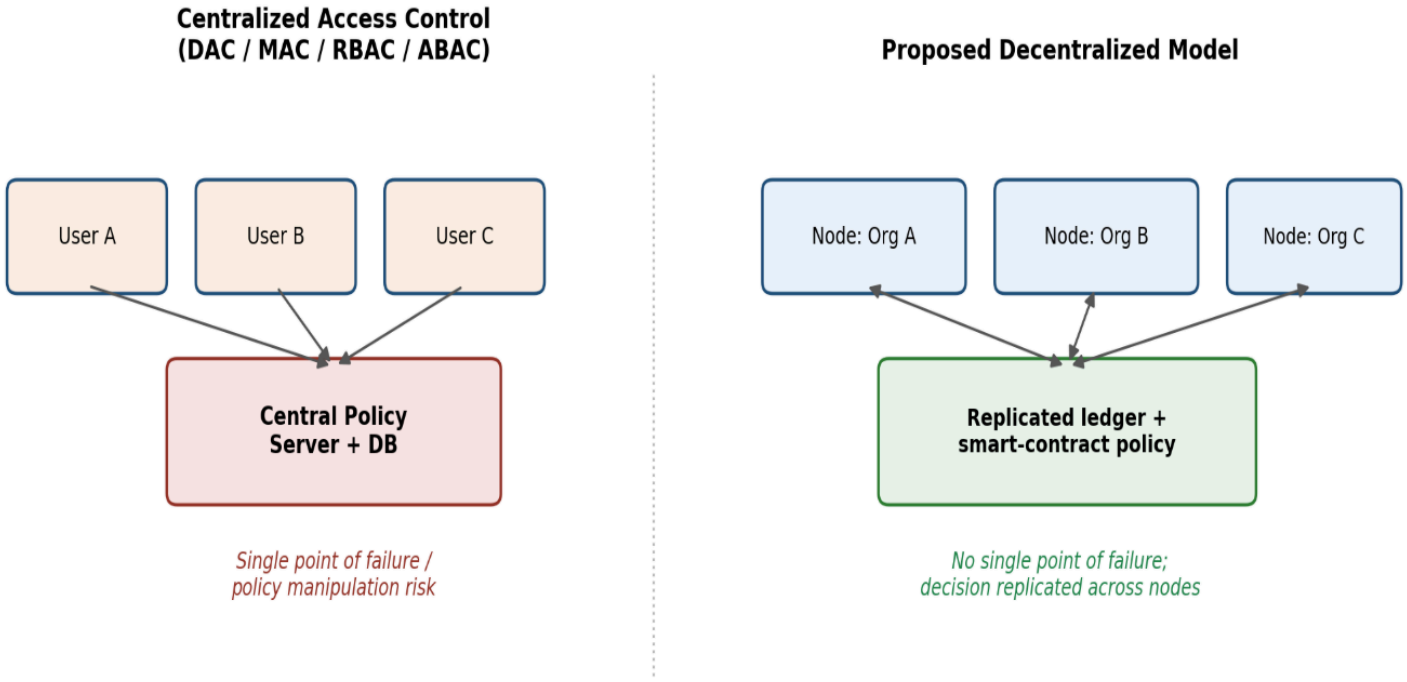


Fig. 4. Structural comparison of centralized and proposed decentralized access-control models

Figure 4

See image above for figure legend.