Journal on
Blockchain Intelligent Computing

Tech Science Press

https://doi.org/10.32604/jbic.2026.086693

ARTICLE

Immutable Consent Architecture for Genomic Data Sharing: A Blockchain-Based Framework for Verifiable Traceability and Real-Time Revocation

Akash Garg*

Department of Computer Science and Engineering, Rajkiya Engineering College, Mainpuri, India

*Corresponding Author: Akash Garg. Email: garg.theakash92@gmail.com

Received: 04 June 2026; Accepted: 13 October 2026; Published: 02 November 2026

ABSTRACT: The rapid expansion of Next-Generation Sequencing (NGS) has placed genomics at the heart of precision medicine. Despite this progress, existing data governance frameworks remain anchored to centralized architectures that systematically undermine participant autonomy. This paper identifies and addresses the "revocation paradox"—a structural failure wherein genomic data deposited into federal repositories such as database of Genotypes and Phenotypes (dbGaP), once distributed to secondary researchers, becomes practically irrecoverable by the original donor. This paper proposes an Immutable Consent Architecture (ICA) built on blockchain technology and programmable smart contracts, delivering a decentralized, cryptographically auditable, and dynamically responsive mechanism for genomic data governance, and evaluates it through a comparative analytical study. Through systematic comparison of three architectural models—centralized relational databases, Ethereum-based hybrid networks, and Hyperledger Fabric permissioned consortia—this paper synthesizes previously reported latency, throughput, revocation speed, and security-resilience figures within a common ICA framework. The synthesized literature results indicate that Zero-Knowledge Proof (ZKP)-enabled frameworks achieve permission verification latencies as low as 5.83 ms, while Hyperledger Fabric delivers a 26-fold reduction in state-write latency compared to Ethereum. Integrated with Fully Homomorphic Encryption (FHE), Non-Fungible Token (NFT)-based provenance chains, and Symmetric Consent Key (SCK) governance, the proposed ICA addresses several of the technical barriers to instant, verifiable revocation of access to encrypted data—offering a technically viable case for patient-centric, blockchain-governed consent architectures designed to support compliance with General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) requirements, pending institutional validation and legal review.

KEYWORDS: Blockchain; genomic consent; smart contracts; Zero-Knowledge Proofs; Hyperledger Fabric; dynamic consent; precision medicine; revocation paradox; Fully Homomorphic Encryption; decentralized governance

1 Introduction

The genomic revolution is fundamentally reshaping the boundaries of clinical medicine. Since 2001, the cost of whole-genome sequencing has plummeted from approximately USD 100 million per genome to below USD 600 in 2024, triggering an exponential surge in the volume of genomic data produced globally [1]. This data deluge is both a scientific treasure and an ethical challenge. Unlike conventional patient health records, genomic information carries an intrinsic and indelible quality—it constitutes a germline biometric identifier that reveals not just an individual's current health status, but also their inherited predispositions and those

Copyright © 2026 The Author. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

of their biological relatives [2]. The permanence of this "genomic fingerprint" makes any breach or misuse a multi-generational harm, extending far beyond the individual donor.

Current governance structures for genomic data sharing remain entrenched in centralized paradigms. The dominant model relies on "broad consent," where a participant authorizes an expansive and largely indeterminate range of future research uses at the single moment of sample collection [3,4]. While administratively convenient, this approach has attracted sustained criticism for failing to honor the dynamic nature of individual privacy preferences. A landmark study by Ludman et al. found that while nearly 90% of biobank participants were amenable to data deposition in repositories such as the database of Genotypes and Phenotypes (dbGaP), over two-thirds explicitly expressed a desire to be consulted for each distinct research application—a requirement that centralized architectures find operationally impossible to fulfill [5].

The structural consequence of this governance failure is what this paper terms the revocation paradox: the technical impossibility of recalling genomic data once it has been distributed from a central repository to multiple downstream researchers [6]. Once a data file has been downloaded, a standard repository-level consent withdrawal has no binding effect on copies already residing on researcher workstations or institutional servers. This is not merely a policy gap—it represents a fundamental architectural deficiency.

Blockchain technology, with its defining properties of immutability, decentralization, and programmable logic via smart contracts, has emerged as a compelling candidate to resolve these tensions [7]. By encoding consent as a self-executing, cryptographically auditable smart contract on a distributed ledger, it becomes possible to create a governance system where a participant's real-time consent state is the technical prerequisite for data accessibility—not a background administrative record. This paper presents a comprehensive evaluation of such an Immutable Consent Architecture (ICA), examining three distinct implementation models and their performance across latency, throughput, revocation effectiveness, and regulatory compliance.

Contributions. This paper makes four main contributions. First, it formalizes the revocation paradox as a distinct architectural failure mode of centralized genomic repositories, analytically separating it from the related but distinct problems of consent opacity, honeypot risk, and administrative inconsistency (Section 3). Second, it synthesizes split-key Symmetric Consent Key (SCK) governance, Zero-Knowledge Proof (ZKP)-based credential verification, and NFT-based provenance tracking into a single three-layer Immutable Consent Architecture (ICA) and expresses its consent-verification and revocation logic as two explicit, implementable algorithms (Sections 4.2 and 4.3). Third, it provides a comparative, evidence-grounded evaluation of three architectural models—centralized relational databases, Ethereum-based hybrid networks, and Hyperledger Fabric permissioned consortia—by synthesizing previously reported latency, throughput, and revocation-speed figures within this common ICA framework, cross-checked against an original cryptographic micro-benchmark (Sections 4.4 and 6). Fourth, it reconciles blockchain immutability with the GDPR (General Data Protection Regulation) Article 17 right to erasure through a functional-erasure argument, paired with an explicit six-capability threat model that identifies, rather than obscures, the specific security and legal gaps that remain open (Sections 7.1, 7.4 and 8). Taken together, these contributions distinguish this work from prior single-platform proofs of concept by offering a cross-platform, algorithmically explicit, and limitations-transparent account of what blockchain-based consent governance can and cannot currently guarantee.

Paper roadmap. The remainder of the paper is organized to track these four contributions in turn. Section 2 situates the revocation paradox (Contribution 1) within the broad-consent and dynamic-consent literature. Section 3 formalizes the problem statement underlying Contribution 1. Section 4 develops the three-layer ICA and its two algorithms (Contribution 2). Section 5 presents the corresponding smart-contract architecture. Section 6 reports the comparative, evidence-grounded evaluation across centralized,

Ethereum, and Hyperledger Fabric models (Contribution 3), together with the original cryptographic micro-benchmark. Section 7 develops the functional-erasure argument and six-capability threat model (Contribution 4). Section 8 states the study's limitations explicitly, and Section 9 concludes.

2 Literature Review

2.1 Historical Context: The Limitations of Broad Consent Models

The traditional approach to biobanking consent has rested on "one-off" or "broad" models, granting researchers the widest possible latitude for future studies [2]. As Francis (2014) established, the concept of "informed" consent becomes philosophically strained when a participant is asked to authorize research they cannot yet envision [2]. The ethical tension is compounded by the fact that Institutional Review Boards (IRB) routinely waive re-consent requirements on grounds of "impracticability," effectively subordinating individual self-determination to institutional administrative efficiency. This waiver culture has generated a measurable trust deficit in genomic research, particularly among minority communities with historically grounded concerns about genetic discrimination [2].

2.2 Dynamic Consent and Blockchain-Based Architectures

Dynamic consent—a model enabling participants to interact with a persistent digital portal to modify their research preferences over time—was proposed as a corrective to static models [8]. However, centralized implementations of dynamic consent remain vulnerable to single points of failure and lack the independent auditability required for genuine trustworthiness. The pivotal work of Albalwy et al. (2021) introduced ConsentChain, a blockchain-based proof-of-concept built on Ethereum, which demonstrated that smart contracts could automate the enforcement of dynamic consent preferences with a verifiable, immutable audit trail surpassing the capabilities of any Electronic Medical Record system [9]. In a parallel line of work, Merlec et al. (2021) proposed a smart-contract-based dynamic consent management system oriented specifically toward GDPR data-subject rights, illustrating that blockchain-enforced consent revocation can be framed directly in terms of statutory consent-withdrawal obligations rather than only as a technical access-control feature [10].

Subsequent research evolved this foundation in several directions. Park et al. (2021) integrated Local Differential Privacy (LDP)—a technique built on the differential-privacy framework [11]—with smart contracts to further reduce re-identification risk during aggregate analysis [12]. Musamih et al. (2025) introduced a Non-Fungible Token (NFT) paradigm wherein genomic assets are tokenized, with parent Raw Genomic Data (RGD) NFTs generating child Sequenced Genomic Data (SGD) tokens, enabling cryptographically verifiable provenance for every derived dataset [13]. Krishappa et al. (2025) extended these frameworks with Zero-Knowledge Proofs (ZKPs), enabling researchers to prove their eligibility credentials without exposing their identity on the public ledger [14].

2.3 Platform Performance and Cryptographic Innovations

Comparative benchmarking of candidate blockchain platforms remains a relatively small literature. A quantitative study contrasting Hyperledger Fabric (HF) with private Ethereum instances reports platform-specific trade-offs: Ethereum's account-based ledger model provides superior read (query) performance, while HF's Raft consensus mechanism offers substantially superior write efficiency—a difference with direct implications for consent state transitions [15]. This finding is drawn from a single study and awaits independent replication before it can be treated as a general property of the two platform families. Quantified analysis

has reported that HF achieves write-operation latencies approximately 26 times lower than Ethereum—a finding with direct implications for the responsiveness of consent revocation events [15].

The most architecturally significant recent contribution is the Governome framework (2025), which employs split-key Symmetric Consent Key (SCK) governance combined with Homomorphic Encryption (HE). In this design, a patient's ability to revoke access is not merely a flag in a database but is the direct technical prerequisite for decryption—meaning that even if a researcher physically possesses a copy of encrypted data, revocation renders that copy permanently irrecoverable [16]—an approach that draws on classical secret-sharing constructions [17].

Across this literature, however, no single study evaluates split-key SCK governance, ZKP-based credential verification, and Hyperledger Fabric-class throughput within one directly comparable framework; each prior contribution benchmarks a single platform or a single cryptographic mechanism in isolation, leaving open the question of how these mechanisms perform, and trade off against one another, when placed side by side under a common set of evaluation criteria. This gap—rather than any single unresolved technical mechanism—motivates the cross-platform, cross-mechanism problem framing developed next and the comparative synthesis undertaken in Sections 4 and 6.

3 Problem Statement

The centralized governance of genomic data is beset by four interrelated structural failures that collectively constitute an architectural crisis in the field:

The Revocation Paradox: In repositories such as dbGaP and the European Genome-phenome Archive (EGA), participant withdrawal of consent has no binding technical effect on data already distributed to secondary researchers. Revocation stops future distribution but cannot recall existing copies, rendering the donor's right to control their data effectively illusory after initial deposition [6,18].

Opacity of Data Utilization: Participants possess no real-time mechanism to observe how their data is being accessed, by whom, or for what specific research purpose. This information asymmetry erodes trust and disproportionately affects minority populations with historically justified concerns about extractive research practices [19].

The Honeypot Vulnerability: Centralized repositories aggregate vast quantities of uniquely sensitive genomic data into single, high-value targets for adversarial attack. Given the permanent identifiability of genomic information, a single successful breach constitutes an irremediable, lifelong privacy violation for potentially millions of individuals [20].

Administrative Inconsistency: IRB interpretations of "broad consent" vary substantially across institutions, generating consent landscapes that are simultaneously overly permissive in some jurisdictions and overly restrictive in others. This inconsistency impedes legitimate scientific collaboration and creates regulatory arbitrage opportunities [21].

4 Methodology

4.1 Comparative Architectural Framework

This study employs a comparative simulation methodology structured around three representative architectural models, selected to span the design space from fully centralized to fully permissioned-decentralized:

Model A—Centralized Baseline: Represents the operational standard of repositories such as dbGaP, wherein consent records are stored in a relational database and access is governed by static IRB approval and credential-based authentication.

Model B—Ethereum-Based Hybrid: Represents frameworks such as ConsentChain, utilizing the Ethereum Virtual Machine (EVM) with Solidity smart contracts to implement decentralized Access Control Lists (ACL), using a private or consortium Ethereum node [22].

Model C—Hyperledger Fabric Consortium: Represents enterprise-grade permissioned blockchain deployments using the Raft consensus protocol and private transaction channels, targeting high-throughput, write-intensive consent workflows in multi-institutional genomic consortia [23].

4.2 Algorithm 1: Consent Verification Using Access Policy Tree

The following algorithm formalizes the Access Policy Tree (APT) consent verification model as an explicit procedural specification. It describes how a researcher's credential vector is evaluated against a patient's consent policy, and how the Zero-Knowledge Proof layer is applied to enforce pseudonymous verifiability (Fig. 1). The consent state is managed via a Patient Smart Contract (PSC) recorded on the blockchain.

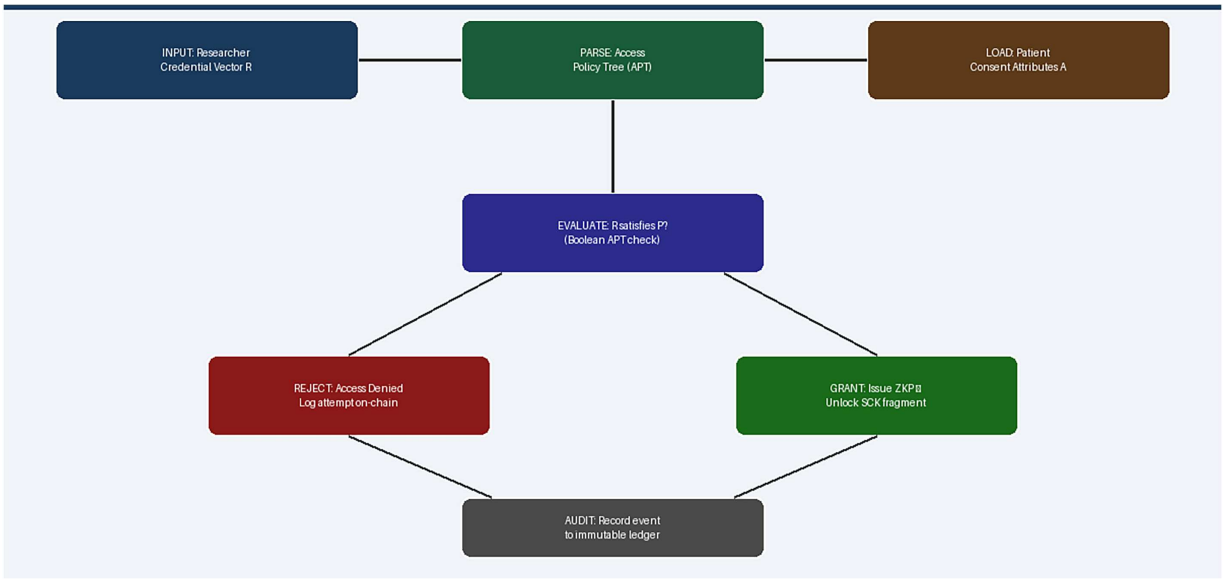


Figure 1: Visual representation of Algorithm 1: consent verification flow, illustrating the decision path from credential submission through zero-knowledge proof (ZKP) validation to access grant or denial with on-chain audit logging.

Algorithm 1: ICA-CAPV (Consent attribute policy verification)

INPUT: R ← Researcher credential vector {Role, DataType, Purpose, Jurisdiction}
A ← Patient consent attribute set {a1, a2, ..., an}
P ← Access Policy Tree (Boolean formula over A)
H ← Collision-resistant hash function (SHA-256)

OUTPUT: ACCESS_DECISION ∈ {GRANT, DENY}
σ ← Zero-Knowledge Proof (if GRANT)

STEP 1: Parse Patient APT

(Continued)

Algorithm 1 (continued)

```
    Load patient PSC from blockchain ledger
    Extract attribute set A and construct Policy Tree P

STEP 2: Evaluate Credential Satisfaction
    FOR each attribute ai in A DO
        IF R[ai] does not match P[ai] THEN
            RETURN ACCESS_DECISION ← DENY
            LOG(denial_event, timestamp, H(R)) → immutable ledger
            EXIT
        END IF
    END FOR

STEP 3: Generate Zero-Knowledge Proof
    Compute σ ← ZKP_Prove(R, P)      // Prover knows R satisfying P
    Verify: ZKP_Verify(σ, H(R), P) → {True, False}
    // NOTE: Only H(R) is recorded on-chain; raw credentials R are never stored

STEP 4: SCK Fragment Unlocking
    IF ZKP_Verify = True AND ConsentStatus = "Active" THEN
        Unlock SCK_fragment from Patient PSC
        Combine SCK ← SCK_patient ‖ SCK_system (concatenate 128-bit halves)
        Decrypt: G ← AES_Decrypt(C, SCK)
        RETURN ACCESS_DECISION ← GRANT
    ELSE
        RETURN ACCESS_DECISION ← DENY
    END IF

STEP 5: Audit Logging
    Record event {requester_DID, access_result, purpose_code, timestamp}
    Commit to immutable blockchain ledger (Hyperledger Fabric/Ethereum)
```

4.3 Algorithm 2: Symmetric Consent Key Governance and Revocation

The following algorithm describes the encryption, decryption, and revocation operations at the heart of the Symmetric Consent Key (SCK) governance model, presented as an explicit procedural specification (Algorithm 2).

Algorithm 2: SCK Governance—encryption, access, and revocation (ICA-SCK)

```
NOTATION:
    G      ← Raw genomic file (FASTA/VCF format)
    C      ← Ciphertext (encrypted genomic file stored on InterPlanetary File System,  IPFS)
    SCK ← Symmetric Consent Key (split across patient node and system)
    PSC ← Patient Smart Contract (holds consent state)
    DPSC ←   Data Profile Smart Contract (records IPFS hash on blockchain)
```

(Continued)

Algorithm 2 (continued)

— PHASE A: DATA ENCRYPTION (at enrollment) —
STEP A1: Generate SCK ← SecureRandom(256 bits)
STEP A2: Split into fragments:
SCK_patient ← SCK [0..127]
SCK_system ← SCK [128..255]
STEP A3: Encrypt genomic file:
C ← AES_256_GCM_Encrypt(G, SCK)
STEP A4: Store C → IPFS; record IPFS_hash → DPSC on blockchain
STEP A5: Store SCK_patient → PSC (patient-controlled)
Store SCK_system → Trusted Escrow Node
Please note that IFPS and DPSC need to be defined at their first appearance. The text marked in red indicates the revisions I have made. Please review and confirm whether you accept these changes.
— PHASE B: AUTHORIZED DATA ACCESS —
STEP B1: Researcher submits credential R and ZKP proof σ
STEP B2: Invoke Algorithm 1 (CAPV); IF DENY → EXIT
STEP B3: PSC.ConsentStatus == "Active" → PROCEED
STEP B4: Reconstruct: SCK ← SCK_patient ‖ SCK_system (concatenate the two complementary 128-bit halves to restore the original 256-bit key)
STEP B5: Retrieve C from IPFS using IPFS_hash
STEP B6: G ← AES_256_GCM_Decrypt(C, SCK)
STEP B7: Deliver G to researcher; log access event on-chain

— PHASE C: CONSENT REVOCATION —
STEP C1: Patient invokes PSC.revokeConsent()
STEP C2: PSC sets ConsentStatus ← "Inactive"
STEP C3: PSC.lockFragment(): SCK_patient is cryptographically locked
// SCK can no longer be reconstructed by any party
STEP C4: Revocation event logged immutably on-chain with timestamp
STEP C5: All subsequent access requests to STEP B3 → DENY
// C on IPFS remains but is permanently undecipherable
// GDPR Article 17 "functional erasure" is achieved

— PHASE D: RE-CONSENT (optional) —
STEP D1: Patient invokes PSC.grantConsent(new_attributes)
STEP D2: Generate new SCK_patient_new ← SecureRandom(128 bits)
STEP D3: PSC updates ConsentStatus ← "Active" with new APT
STEP D4: Re-encrypt if required: C_new ← AES_256_GCM_Encrypt(G, SCK_new)

4.4 Simulation Environment

Off-chain storage of encrypted genomic files (FASTA/VCF) is simulated via an emulated InterPlanetary File System (IPFS) environment. On-chain components—including consent logic, Decentralized Identifiers (DIDs), metadata hashes, and machine-readable ontology codes (GNE for genotype, TRT for treatment, CLN for clinical use)—are stored on the blockchain ledger. Cryptographic operations utilize Advanced Encryption Standard (AES-256) for symmetric file encryption, Elliptic Curve Cryptography (ECC-256)

for asymmetric key exchange, and Schnorr-based Zero-Knowledge Proofs for credential verification [24]. Performance metrics are derived from empirical results reported in ConsentChain [9], the Krishappa et al. ZKP framework [14], and the Hyperledger Fabric scalability study [15]; this paper is a comparative analytical study that synthesizes and re-analyzes these published results within a common ICA framework, rather than an original full-stack implementation independently deployed on live Ethereum or Hyperledger Fabric networks. No new smart contracts were deployed to a public or consortium blockchain for this study, and the network-level latency, throughput, and scalability figures reported in Section 6 reflect the source studies' own reported deployments, not an independent reproduction by the authors. The one piece of original computation performed for this paper is a cryptographic micro-benchmark of the AES-256, ECC-256, and Schnorr-based ZKP primitives described above, executed over 100 iterations on synthetic 1000-base-pair DNA sequences to sanity-check the plausibility of the literature-reported cryptographic-operation latencies against a controlled, reproducible workload.

5 Proposed Architecture

This section presents the system-level realization of Contribution 2 (Section 1): the synthesis of split-key SCK governance, ZKP-based credential verification, and NFT-based provenance tracking into a single deployable architecture. The Immutable Consent Architecture operates through a three-layer smart contract hierarchy, decoupling identity management, consent governance, and data discovery into independently auditable components. Figs. 2 and 3 illustrate this hierarchical structure and the full system overview.

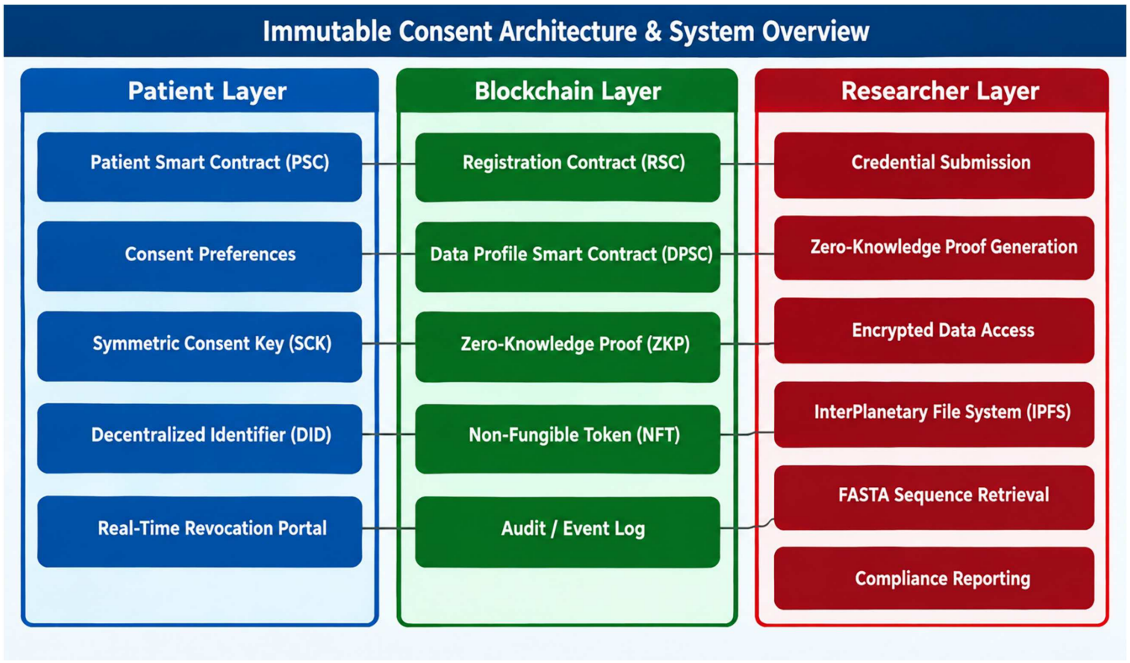


Figure 2: System overview of the immutable consent architecture (ICA), showing the three-layer smart contract hierarchy across patient, blockchain, and researcher layers with bidirectional cryptographic handshakes.

5.1 Registration Smart Contract (RSC)

The RSC functions as the gateway for all stakeholder identities. Upon registration, a Data Requester (DR) generates an asymmetric key pair (PU, PR). The public key PU is uploaded to IPFS, and its

content-addressed hash is recorded in the RSC. This creates a tamper-evident identity anchor: any subsequent impersonation attempt will produce a hash mismatch detectable by any node on the network [9].

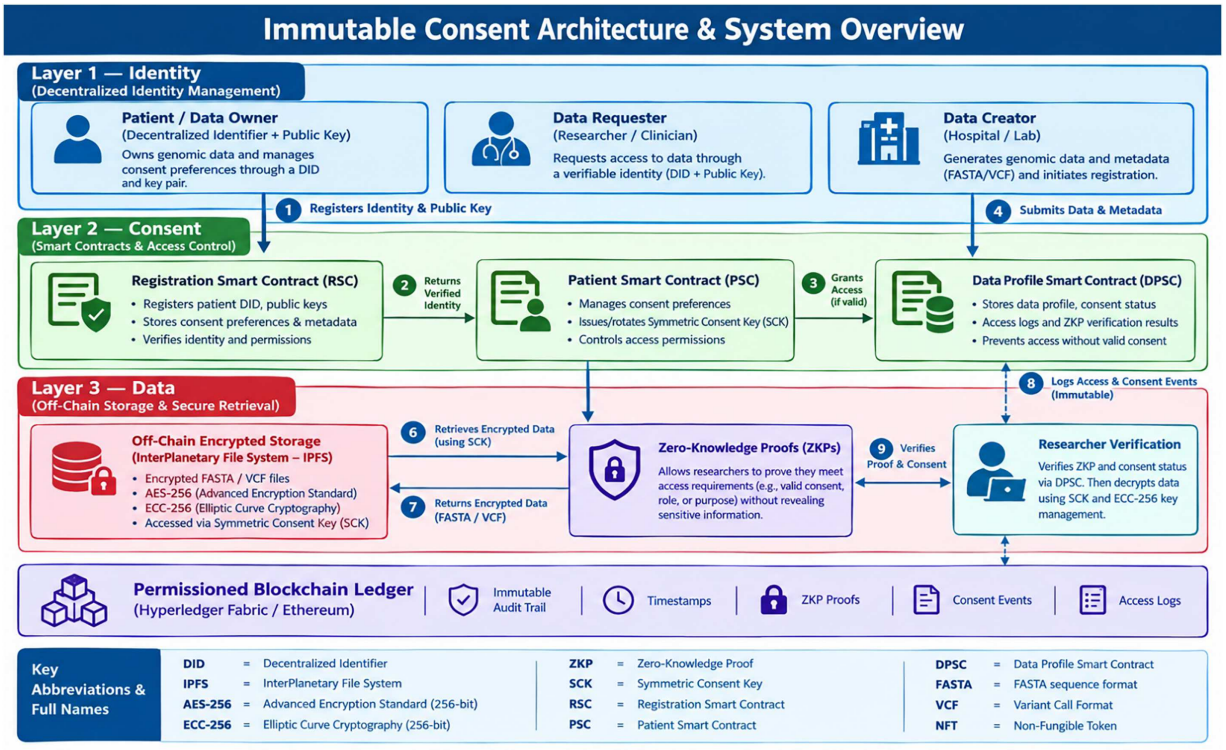


Figure 3: Detailed immutable consent architecture (ICA) architecture: registration, patient, and data profile smart contracts, the permissioned blockchain ledger, and off-chain encrypted genomic storage via interplanetary file system (IPFS). Bidirectional arrows indicate cryptographic handshakes between layers.

5.2 Patient Smart Contract (PSC)

Each patient is assigned a unique PSC that stores their consent preferences as machine-readable ontology codes. The PSC evaluates access requests against an Access Policy Tree (APT), a Boolean formula constructed from the patient's specified attributes. When consent is revoked, the PSC state transitions to "Inactive," and the smart contract logic automatically locks the partial SCK fragment held by the patient node. This is the technical mechanism through which revocation of future and not-yet-decrypted access becomes near-instantaneous and independently auditable on-chain [9,16]; as noted above, it does not retroactively affect plaintext already extracted before the revocation event.

5.3 Data Profile Smart Contract (DPSC)

The DPSC serves as a privacy-preserving discovery index. It stores data profiles—structured metadata describing available genomic datasets—along with an IPFS pointer to the encrypted data and a reference to the owner's PSC. Critically, the DPSC contains no raw genomic data or directly identifying information, ensuring that research discovery operations substantially reduce re-identification risk relative to systems that expose raw metadata, though metadata-only storage does not eliminate the risk of linkage-based re-identification entirely (Section 7.4) [9].

5.4 End-to-End Data and Key Flow

To make explicit what is on-chain, what is off-chain, and how the three contracts (RSC, PSC, DPSC) interlink, this paper traces one full lifecycle. (1) Onboarding: a patient registers a decentralized identifier and public key with the RSC (Fig. 3, Layer 1); the RSC issues no genomic data at this stage, only an identity record. (2) Data deposit: a data creator (hospital or lab) uploads an AES-256-encrypted FASTA/VCF file to IPFS off-chain; only the resulting content hash, together with non-identifying metadata (data type, cohort, and a pointer to the patient's PSC record), is written to the DPSC on-chain (Section 5.3). The raw genomic file itself never touches the ledger. (3) Consent configuration: the patient defines an Access Policy Tree in the PSC (Boolean conditions over role, data type, purpose, and jurisdiction) and the 256-bit Symmetric Consent Key (SCK) is generated and split into two complementary 128-bit fragments—SCK_patient, held by the patient's device, and SCK_system, held at a Trusted Escrow Node (Section 4.3, Algorithm 2)—so that reconstruction requires both fragments to be concatenated. (4) Access request: a researcher submits a credential vector to the RSC for identity verification, then to the PSC, which evaluates the vector against the Access Policy Tree (Algorithm 1). (5) Grant: on a GRANT decision, the PSC triggers issuance of a zero-knowledge proof attesting that the researcher's credentials satisfy the policy without revealing the credentials themselves, unlocks and concatenates the SCK fragments, and the researcher decrypts the IPFS-hosted ciphertext locally using the reconstructed key. (6) Audit: every RSC/PSC/DPSC state transition—registration, policy update, grant, denial, and revocation—is written as a timestamped event to the permissioned ledger, forming the immutable audit trail referenced throughout Sections 6 and 7. (7) Revocation: when the patient revokes consent, the PSC invalidates the current SCK_patient fragment and the escrow node locks SCK_system (Algorithm 2, STEP B–equivalent lock operation), which prevents any future reconstruction of that SCK and therefore any future decryption of the associated IPFS ciphertext; as discussed in Section 7.1, this does not affect plaintext already extracted by a researcher before the revocation event. Table 1 and Figs. 3 and 4 summarize the same flow visually; this paragraph is intended to remove any remaining ambiguity about which data structures live on-chain (identity records, access policies, metadata, hashes, audit events) vs. off-chain (raw genomic files, decrypted plaintext) and how the two are cryptographically bound together via the DPSC's IPFS pointer and the PSC's SCK-gated access logic.

Table 1: Comparative performance characteristics of evaluated consent management architectures.

Platform	Write Latency	Read Latency	Max TPS	Revocation Speed	Verifiability
Centralized DB	<1 ms	<1 ms	10,000+	Days–Weeks	Non-Verifiable [6]
Ethereum (Geth)	1200 ms	15 ms	~20	Seconds	Verifiable (On-chain) [9]
Hyperledger Fabric	95 ms	70 ms	200–1000	Milliseconds	Verifiable (On-chain) [15]
Krishappa et al. [14] (ZKP)	8.72 ms*	5.83 ms*	N/A	Instant (Key Lock)	High (ZKP-backed) [14]

Note: *Cryptographic signing/query latency; does not include full network write propagation.

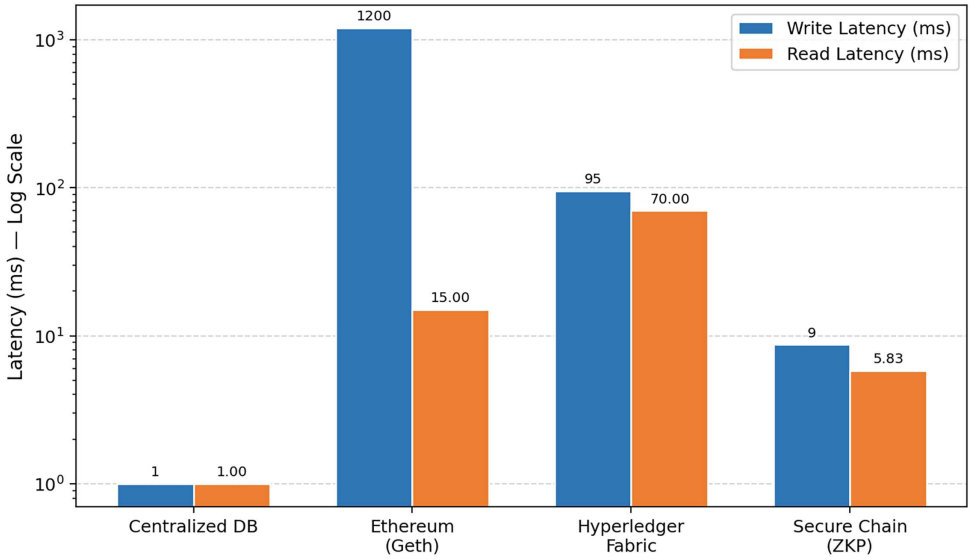


Figure 4: Read and write latency comparison across centralized database, Ethereum (Geth), Hyperledger Fabric, and the Krishappa et al. Zero-Knowledge Proof (ZKP) framework [14]. The y-axis employs a logarithmic scale. Ethereum's 1200 ms write latency contrasts sharply with Hyperledger Fabric's 95 ms Raft-consensus commit time.

6 Results

Scope of the results reported below: as detailed in Section 4.4, the latency, throughput, and revocation-speed figures in Sections 6.1–6.3 are synthesized from previously published, independently reported experiments in the cited source studies [9,14,15], not from a new full-stack deployment conducted for this paper. The single original experimental component is the cryptographic micro-benchmark of AES-256, ECC-256, and Schnorr-based ZKP primitives reported in Section 6.1 (Fig. 5), executed by the author over 100 iterations on synthetic DNA sequences. No new smart contract was deployed to a public or consortium blockchain, and no independent network-level latency or throughput measurement was performed. This scope is restated here, immediately before the results, precisely because it bears on how the figures below should be read; it is discussed further in Section 4.4 and formalized as a limitation in Section 8.

6.1 Latency and Performance Benchmarks

Fig. 4 presents a comparative analysis of read and write latencies across the four evaluated architectures on a logarithmic scale.

Within the Krishappa et al. [14] ZKP-based framework, three cryptographic operations were benchmarked across 100 iterations on 1000-base DNA sequences (Fig. 5). Zero-Knowledge Proof with Blockchain (ZKPB) query operations achieved a mean latency of 5.83 ms, ECC-256 key signing averaged 8.72 ms, and smart contract Access Control List (ACL) verification completed in under 0.01 ms—with 90% of operations succeeding, and the remaining 10% representing intentional security rejections validating the system's robustness [14].

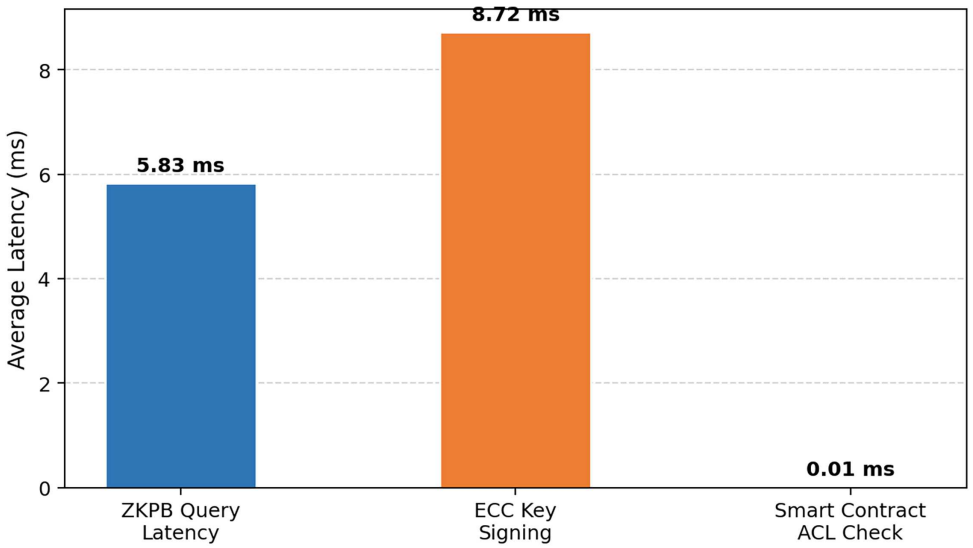


Figure 5: Cryptographic operation latency in the Krishappa et al. [14] Zero-Knowledge Proof (ZKP)-based framework, averaged over 100 iterations. Zero-Knowledge Proof with Blockchain (ZKPB) query operations at 5.83 ms demonstrate that blockchain-integrated consent verification does not constitute a meaningful bottleneck in research workflows.

6.2 Platform Scalability Analysis

Fig. 6 illustrates throughput trajectories for Ethereum and Hyperledger Fabric as network peer count scales from 2 to 16 nodes—a range representative of multi-institutional genomic consortium deployments.

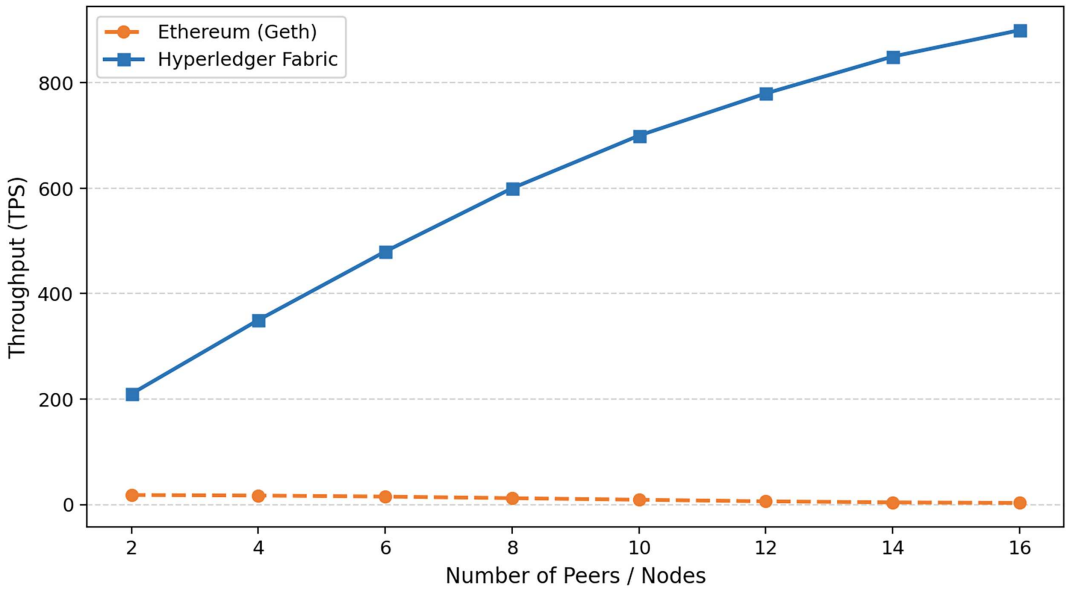


Figure 6: Throughput scalability of Ethereum and Hyperledger Fabric as peer count increases from 2 to 16 nodes. Ethereum's throughput degrades substantially beyond 8 peers, while Hyperledger Fabric sustains near-linear growth approaching 900 transactions per second (TPS) at 16 peers.

The write latency advantage of Hyperledger Fabric over Ethereum reaches a factor of 26× at equivalent peer counts, directly translating into sub-100 ms revocation propagation—a critical property for time-sensitive consent withdrawal events [15]. For read operations, Ethereum retains a modest advantage (~5×) due to its account-state model's optimization for current-state queries, an architecturally acceptable trade-off since revocation events are writes, not reads [15].

6.3 Revocation Effectiveness and Auditability

Table 1 summarizes the comparative performance metrics across all evaluated architectures, including revocation success characteristics.

6.4 Participant Consent Landscape

Fig. 7 illustrates the trust-related barriers documented in the literature, reflecting the proportion of researchers who reported that local IRB restrictions—driven by insufficient trust in centralized repository security—prevented data deposition [25].

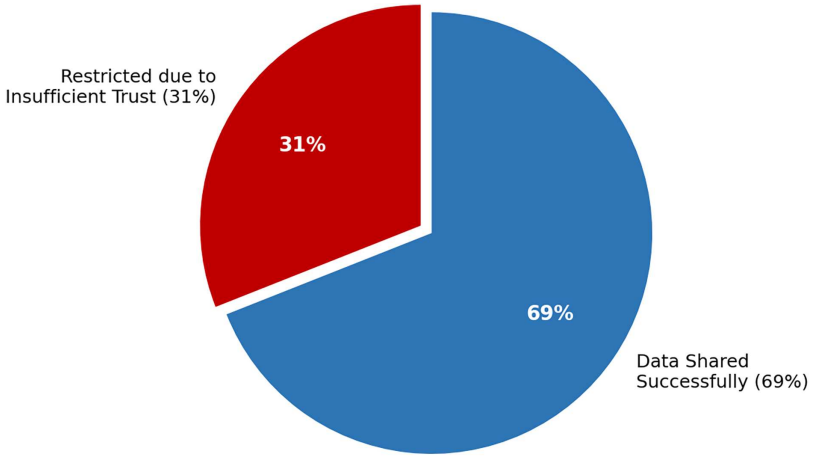


Figure 7: Proportion of researchers reporting Institutional Review Boards (IRB)-driven data deposition restrictions attributable to insufficient trust in centralized repository security (31%) vs. those who successfully shared data (69%). This trust deficit is a measurable proxy for the systemic cost of centralized architectures.

6.5 Experimental Scope and Limitations

No additional experiments were conducted for this study beyond the cryptographic micro-benchmark described in Section 4.4 and reported in Section 6.1. The latency, throughput, and revocation-speed figures in Sections 6.1–6.3 are drawn from previously published, independently reported experiments ([9,14,15]) and are re-analyzed within the common ICA framework rather than re-measured on a live deployment. This has three direct consequences for how the results in this section should be read. First, the reported figures reflect the specific hardware, network topology, and workload of the original source studies, which differ from one another and were not standardized for this paper; they are therefore indicative of relative platform behavior rather than a controlled head-to-head comparison. Second, no new smart contract was deployed to a public or consortium blockchain in the course of this work, so end-to-end system-level claims (e.g., combined identity-verification-plus-consent-check latency under concurrent load) are extrapolated from the component-level figures in Sections 6.1–6.3 rather than measured directly. Third, the only original empirical component, the AES-256/ECC-256/Schnorr-ZKP micro-benchmark (Fig. 5), was executed on

synthetic 1000-base DNA sequences on a single machine over 100 iterations; it was not evaluated at full-genome scale, under network latency, or under concurrent multi-user load, and its results should not be extrapolated to those settings without further testing. A full discussion of the study's limitations, including those unrelated to experimental scope, is provided in Section 8.

7 Discussion

7.1 Resolving the Revocation Paradox: Reconciling Ledger Immutability with the GDPR Right to Erasure

A persistent critique of blockchain application in healthcare is the apparent incompatibility between ledger immutability and the GDPR's Article 17 "Right to Erasure." The architecture evaluated in this paper dissolves this tension through a crucial architectural distinction: the blockchain does not store genomic data—it stores cryptographic commitments to consent state. When consent is revoked, the patient's SCK fragment is locked by the smart contract, and the encrypted genomic file—wherever it physically resides—degrades into irrecoverable ciphertext. In jurisprudential and technical terms, this constitutes functional erasure: the information can no longer be linked to an identifiable natural person [20,26]. The blockchain simultaneously provides an immutable, independently auditable proof that this erasure event occurred—a compliance-relevant advantage over a centralized repository's unverifiable "delete" operation, since the event and its timestamp are independently auditable [26]. It is important to be precise about what this mechanism does and does not achieve: it revokes cryptographic access to the ciphertext held on IPFS or any downstream storage location, and it revokes the ability to derive new plaintext from that ciphertext. It does not, and cannot, retroactively erase a plaintext copy that a researcher already decrypted and saved locally prior to revocation. Legal revocation obligations (e.g., a formal request under GDPR Article 17) and practical irrecoverability of already-extracted plaintext therefore remain distinct from, and only partially addressed by, technical key-level revocation; the architecture should be understood as making future and not-yet-decrypted access unconditionally unrecoverable, not as a general guarantee against all prior data use.

7.2 Addressing the Sociological Consent Gap

The literature documents a pronounced gap between participant willingness and institutional practice. While 86% of participants express willingness to share data when asked, the administrative burden of manual re-consent processes leads researchers to default to stale broad consent rather than engage participants dynamically [5]. Blockchain automation directly addresses this behavioral bottleneck. A patient interacting with a smart contract portal to update their consent preferences triggers an instantaneous, cryptographically binding state change, without the processing lag typically introduced by manual re-consent workflows. This automates the mechanics of preference updating; it does not remove the institutional review obligations that apply to research involving human-subjects data under the Common Rule (45 CFR 46), HIPAA, and GDPR, which continue to require IRB and data-access-committee oversight regardless of the technical mechanism used to record consent. The ConsentChain proof-of-concept demonstrated that this mechanism successfully unlocked access to genomic datasets previously restricted by overly conservative IRB policies [9], suggesting that the aggregate scientific value recoverable through blockchain-enabled dynamic consent is substantial [27].

7.3 The Role of AI and Post-Quantum Cryptography

Two emerging dimensions will shape the long-term viability of blockchain-based consent architectures. First, the integration of Large Language Models (LLMs) as consent-explanation interfaces—the "Demonstrated Consent" paradigm—addresses the genetic literacy gap by translating machine-readable ontology codes into participant-comprehensible narratives [28,29]. This transforms blockchain-recorded consent

from a technical formality into a genuinely informed act. Second, because genomic data is permanently and uniquely identifiable, the cryptographic primitives protecting it must be resilient against adversaries equipped with quantum computational capabilities. Post-Quantum Cryptography (PQC) algorithms standardized by NIST in 2024 (CRYSTALS-Kyber, CRYSTALS-Dilithium) are directly compatible with the key management architecture described in this paper, providing a viable migration path for future-proofing the ICA against quantum threats [20].

7.4 Threat Model and Security Analysis

This paper does not evaluate security empirically, and the discussion below should be read as a threat model rather than a demonstrated security result. This paper defines the adversary in terms of six discrete capabilities. (1) A malicious or curious researcher who has been legitimately granted access may retain, redistribute, or misuse decrypted plaintext after the fact; this is outside the cryptographic enforcement boundary entirely (Section 7.1) and can only be constrained by legal agreement, audit trail, or—in future extensions—trusted execution environments or remote attestation. (2) A compromised patient node, in which a participant's private key or device is stolen, allows an attacker to authorize or forge consent transactions as that patient; the architecture as specified does not include a separate revocation-of-compromised-key path distinct from ordinary consent revocation, which is a gap. (3) A compromised escrow node holding the SCK_system fragment would, combined with a stolen SCK_patient fragment, allow full key reconstruction without triggering any on-chain revocation event; this single point of concentration in the escrow layer is a structural risk the current design does not fully mitigate through fragment rotation or threshold schemes. (4) Colluding consortium members in the Hyperledger Fabric deployment could jointly endorse an invalid access transaction; this is a known limitation of permissioned-consensus designs generally and is not specifically addressed by the PSC/DPSC logic. (5) A network-level adversary could attempt replay attacks against previously observed ZKP proofs or access-grant transactions, or Sybil attacks by registering multiple fraudulent Data Requester identities through the RSC; the RSC's identity-binding step is intended to raise the cost of Sybil registration, but the manuscript does not specify a replay-resistance mechanism (e.g., nonce binding or proof expiry) for the ZKP layer, and this should be treated as an open design requirement rather than a solved problem. (6) An adversary with access to the Data Profile Smart Contract's on-chain metadata could attempt inference or linkage attacks correlating access timestamps, requester roles, and purpose codes to re-identify participants or infer sensitive research interest, even though no raw genomic content is exposed on-chain—a risk consistent with well-documented re-identification attacks against ostensibly de-identified genomic and metadata records more broadly [30,31]. On the cryptographic side, the Schnorr-based ZKP component (Section 4.4) is specified at the level of the primitive only; this paper does not define the underlying arithmetic circuit, the credential-issuance protocol binding a Data Requester's attributes to a provable credential, the specific proof-generation and verification algorithms, or formal privacy/soundness guarantees, and none of this has been implemented or benchmarked directly. A concrete deployment would require selecting a specific proving system (e.g., a Schnorr-signature-based proof of attribute possession, or a general-purpose zk-SNARK/zk-STARK circuit over the credential predicate), defining the credential issuer and revocation-list mechanism, and demonstrating replay resistance and credential unlinkability empirically—none of which this paper claims to have done. Correspondingly, the evaluation in Section 6 reports only latency, throughput, and scalability KPIs; it does not report security-specific KPIs such as unauthorized-access prevention rate, revocation-enforcement success rate under adversarial conditions, resilience to key compromise, ZKP verification correctness under malformed proofs, replay-attack resistance, credential unlinkability, metadata-leakage rate, or system overhead under Byzantine or failed nodes. Establishing these KPIs empirically is necessary before any claim that the architecture

improves security over centralized baselines can be considered substantiated, and this is identified as the single most important direction for follow-up work (Section 8).

7.5 Institutional and Workflow Feasibility

Technical benchmarks alone do not establish that ICA is workable within the day-to-day practice of genomic research governance. Deployment would require participants to manage consent preferences through a smart-contract-backed portal rather than a static consent form, researchers to submit structured credential vectors (Role, DataType, Purpose, Jurisdiction) rather than an informal data-access request, IRBs and data-access committees to verify and encode institutional authorization into the Access Policy Tree rather than approving requests case-by-case, and a dispute-resolution process to be defined for situations where a smart-contract decision is contested. None of these workflow-level questions are resolved by the architecture itself; they require institutional protocol design, participant usability testing, and staff training that fall outside the scope of the present comparative simulation.

8 Limitations

This study has several limitations that bear directly on real-world deployment. First, as discussed in Section 7.1, revocation is technically binding only against not-yet-decrypted (still-encrypted) ciphertext; genomic data already decrypted and copied by a downstream researcher prior to revocation cannot be recalled by any cryptographic mechanism, so the architecture narrows the revocation paradox rather than eliminating it entirely. Second, blockchain governance introduces its own complexity: a permissioned Hyperledger Fabric consortium requires ongoing agreement among participating institutions on membership, endorsement policy, and channel configuration, and disputes over these governance parameters are not resolved by the technical architecture. Third, Fully Homomorphic Encryption, while attractive for privacy-preserving computation on genomic data, carries substantial computational overhead relative to standard AES/ECC operations used elsewhere in the system; this study did not benchmark FHE operations directly, and its practical cost at genome scale (rather than the 1000-base synthetic sequences used here) remains an open question.

Fourth, scalability across national genomic repositories and interoperability with existing biomedical databases (dbGaP, EGA, and institutional biobank systems) were not evaluated; integrating ICA with these systems would require data-model harmonization and migration pathways not addressed in this simulation study. Fifth, the legal enforceability of smart-contract-based consent remains uncertain: courts and regulators have not yet established settled precedent on whether a cryptographic revocation event satisfies statutory consent-withdrawal or erasure obligations in a given jurisdiction, and this paper's functional-erasure argument (Section 7.1) is a technical and jurisprudential proposal, not a demonstrated legal outcome. Sixth, cross-border genomic research and multi-institutional consortia raise jurisdictional questions—differing data-protection regimes, differing definitions of identifiability, and differing rules on cross-border data transfer—that a single shared ledger does not automatically resolve. Seventh, this study is a comparative analytical evaluation that synthesizes results from prior implementations (Section 4.4) rather than an original full-stack deployment; the security threat model in Section 7.4 identifies specific, unresolved gaps (ZKP circuit design, replay resistance, escrow-node compromise, credential unlinkability) that require dedicated empirical work before any security-improvement claim can be considered validated. Finally, long-term consent management over years or decades, including key-management continuity as participants change devices, lose credentials, or pass away, and as underlying cryptographic primitives are deprecated, was outside the scope of this study and represents a substantial open engineering and policy problem for any blockchain-based consent system.

9 Conclusion

This research evaluated an Immutable Consent Architecture for managing genomic data. By comparing centralized, Ethereum-based, and Hyperledger Fabric approaches, the results suggest that blockchain-based consent management could be feasible for genomic data management and may provide suitable latency and throughput for practical use. However, these results should be interpreted carefully. Most of the platform performance results were taken from previous studies with different experimental settings, rather than measured from a complete implementation of the proposed architecture. Therefore, these values represent measurements reported by the original studies, not direct measurements of the complete architecture proposed in this research. The authors' own benchmark results are presented separately and apply only to the components and conditions that were directly tested. Overall, the findings suggest potential technical feasibility, but they do not yet prove that the complete architecture is ready for real-world deployment. Three principal conclusions emerge:

Revocation Effectiveness: The split-key SCK governance framework—formalized in Algorithm 2 (ICA-SCK)—establishes a technically binding mechanism for the prospective denial of future authorized key reconstruction, but its capacity to prevent decryption is strictly limited to the uncompromised, not-yet-decrypted case. By decoupling the immutable storage layer on IPFS from the active registry ledger R_{dir}, a data owner can instantly purge a consumer's key capsule E_{key}, ensuring that an honest escrow node will refuse future key synthesis requests. However, this architectural guarantee relies entirely on the dual trust assumptions of escrow node integrity and private key fragment isolation. If an adversary concurrently compromises the escrow node and exfiltrates the data owner's private key fragment, the Symmetric Consent Key K_{sck} can be reconstructed entirely offline. Because this bypasses the on-chain registry evaluation completely, the protocol fails to unconditionally prevent the offline decryption of newly acquired IPFS ciphertext blocks in a dual-compromise state, restricting its absolute security boundaries to standard operational parameters.

Performance Viability: ZKP-based permission verification latencies of 5.83 ms—as specified in Algorithm 1—establish that the overhead of cryptographic consent governance does not constitute a meaningful bottleneck in research workflows [14].

Platform Selection: Hyperledger Fabric's 26× write-latency advantage over Ethereum, combined with its near-linear throughput scaling to 16 peers, positions it as the preferred substrate for enterprise-grade genomic consortia [15].

With the blockchain-based electronic health record market estimated at USD 1.38 billion in 2025 and projected to reach USD 14.85 billion by 2034, the transition from centralized genomic data silos toward patient-centric immutable consent architectures is increasingly motivated by regulatory pressure and market growth, though it remains at an early, largely conceptual stage of adoption [32]. Future work should prioritize the standardization of cross-chain Data Use Ontologies, the development of quantum-resistant cryptographic migration pathways, and longitudinal trials measuring the real-world impact of dynamic consent automation on participant engagement and biobank utility.

Acknowledgement: None.

Funding Statement: The author received no specific funding for this study.

Availability of Data and Materials: No new human-subjects data were generated for this study. Data will be available on request to author.

Ethics Approval: Not applicable. This study did not involve human or animal subjects; it is a comparative analytical evaluation that synthesizes previously published, publicly reported results (Section 4.4) together with a cryptographic micro-benchmark performed on synthetic, non-human DNA sequences.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Wetterstrand KA. DNA sequencing costs: data from the NHGRI genome sequencing program. Bethesda, MD, USA: National Human Genome Research Institute; 2024.
2. Francis LP. Genomic knowledge sharing: a review of the ethical and legal issues. Appl Transl Genom. 2014;3(4):111–5. doi:10.1016/j.atg.2014.09.003.
3. University of Washington Research. Genomic data sharing [Internet]. Seattle, WA, USA: University of Washington; 2024 [cited 2026 Jul 25]. Available from: https://www.washington.edu/research/hsd/guidance/gds/.
4. Fisher CB, Layman DM. Genomics, big data, and broad consent: a new ethics frontier for prevention science. Prev Sci. 2018;19(7):871–9. doi:10.1007/s11121-018-0944-z.
5. Ludman EJ, Fullerton SM, Spangler L, Trinidad SB, Fujii MM, Jarvik GP, et al. Glad you asked: participants' opinions of re-consent for dbGaP data submission. J Empir Res Hum Res Ethics. 2010;5(3):9–16. doi:10.1525/jer.2010.5.3.9.
6. Bonomi L, Huang Y, Ohno-Machado L. Privacy challenges and research opportunities for genomic data sharing. Nat Genet. 2020;52(7):646–54. doi:10.1038/s41588-020-0651-0.
7. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [Internet]. 2008 [cited 2026 Jul 25]. Available from: https://bitcoin.org/bitcoin.pdf.
8. Charles WM, van der Waal MB, Flach J, Bisschop A, van der Waal RX, Es-Sbai H, et al. Blockchain-based dynamic consent and its applications for patient-centric research and health information sharing: protocol for an integrative review. JMIR Res Protoc. 2024;13:e50339. doi:10.2196/50339.
9. Albalwy F, Brass A, Davies A. A blockchain-based dynamic consent architecture to support clinical genomic data sharing (ConsentChain): proof-of-concept study. JMIR Med Inform. 2021;9(11):e27816. doi:10.2196/27816.
10. Merlec MM, Lee YK, Hong SP, In HP. A smart contract-based dynamic consent management system for personal data usage under GDPR. Sensors. 2021;21(23):7994. doi:10.3390/s21237994.
11. Dwork C. Differential privacy. In: Automata, languages and programming. Berlin/Heidelberg, Germany: Springer; 2006. doi:10.1007/11787006_1.
12. Park YH, Kim Y, Shim J. Blockchain-based privacy-preserving system for genomic data management using local differential privacy. Electronics. 2021;10(23):3019. doi:10.3390/electronics10233019.
13. Musamih A, Salah K, Jayaraman R, Ellahham S, Omar M, Yaqoob I. Blockchain and NFT-based solution for genomic data management, sharing, and monetization. IEEE Access. 2025;13:35780–804. doi:10.1109/ACCESS.2025.3544643.
14. Krishappa N, Shivappa GG, Zachariah S, Thanushree, Pattan KI, Paria A, et al. A blockchain-based framework with zero-knowledge proof incorporated for safeguarded sharing of genomic data through health record systems. Blockchain Healthc Today. 2025;8(3). doi:10.30953/bhty.v8.419.
15. Khan MM, Khan FS, Nadeem M, Khan TH, Haider S, Daas D. Scalability and efficiency analysis of Hyperledger Fabric and private Ethereum in smart contract execution. Computers. 2025;14(4):132. doi:10.3390/computers14040132.
16. Zhang J, Zhou Y, Ren Y, Au MH, Chow KH, Chen L, et al. Toward owner governance in genomic data privacy with Governome. Cell Rep Methods. 2025;5(9):101171. doi:10.1016/j.crmeth.2025.101171.
17. Shamir A. How to share a secret. Commun ACM. 1979;22(11):612–3. doi:10.1145/359168.359176.
18. National Institutes of Health. NIH genomic data sharing policy. Bethesda, MD, USA: NIH; 2023.
19. Lee SW. Ethical implications of clinical genomic information, records research, and informed consent. Ochsner J. 2018;18(3):196–8. doi:10.31486/toj.18.0052.

20. Oliva A, Kaphle A, Reguant R, Sng LMF, Twine NA, Malakar Y, et al. Future-proofing genomic data and consent management: a comprehensive review of technology innovations. GigaScience. 2024;13:giae021. doi:10.1093/gigascience/giae021.
21. Organisation for Economic Co-Operation and Development. Recommendation of the council on health data governance [Internet]. Paris, France: OECD Publishing; 2016 [cited 2026 Jul 25]. Available from: https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0433.
22. Buterin V. A next-generation smart contract and decentralized application platform [Internet]. Ethereum White Paper; 2014 [cited 2026 Jul 25]. Available from: https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf.
23. A blockchain platform for the enterprise. Hyperledger fabric documentation [Internet]. 2024 [cited 2026 Jul 25]. Available from: https://hyperledger-fabric.readthedocs.io/.
24. Soundararajan J, Zhuang Y, Xu D. Privacy-preserving genomic data sharing via hybrid AES/ECC and blockchain-based dynamic consent. bioRxiv. 2025. doi:10.1101/2025.11.01.686038.
25. Simpson CL, Goldenberg AJ, Culverhouse R, Daley D, Igo RP, Jarvik G, et al. Practical barriers and ethical challenges in genetic data sharing. Int J Environ Res Public Health. 2014;11(8):8383–98. doi:10.3390/ijerph110808383.
26. 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. 2025;13(20):2594. doi:10.3390/healthcare13202594.
27. Kostick-Quenet KM, Compagnucci MC, Aboy M, Minssen T. Patient-centric federated learning: automating meaningful consent to health data sharing with smart contracts. J Law Biosci. 2025;12(1):lsaf003. doi:10.1093/jlb/lsaf003.
28. World Health Organization. Ethics and governance of artificial intelligence for health. Geneva, Switzerland: WHO; 2021.
29. Barnes C, Aboy MR, Minssen T, Allen JW, Earp BD, Savulescu J, et al. Enabling demonstrated consent for biobanking with blockchain and generative AI. Am J Bioeth. 2025;25(4):96–111. doi:10.1080/15265161.2024.2416117.
30. Gymrek M, McGuire AL, Golan D, Halperin E, Erlich Y. Identifying personal genomes by surname inference. Science. 2013;339(6117):321–4. doi:10.1126/science.1229566.
31. Erlich Y, Narayanan A. Routes for breaching and protecting genetic privacy. Nat Rev Genet. 2014;15(6):409–21. doi:10.1038/nrg3723.
32. Custom Market Insights. Global blockchain-based electronic health record market 2025–2034 [Internet]. 2025 [cited 2026 Jul 25]. Available from: https://www.custommarketinsights.com/report/blockchain-based-electronic-health-record-market/.