A Blockchain-Based Chain of Custody for Digital Evidence: Design and Evaluation

DOI: https://doi.org/10.54654/isj.v3i26.1167

Truong Xuan Hung*, Luong The Dung, Tran Anh Tu

***Abstract*— Maintaining a trustworthy chain of custody is essential to ensure the integrity, provenance, and admissibility of digital evidence. However, traditional evidence-management systems often suffer from opacity, limited auditability, and susceptibility to insider abuse or procedural errors. To address these gaps, we propose a blockchain-based chain-of-custody framework built on Hyperledger Fabric. In this design, the full lifecycle of evidence is logged as immutable, permissioned ledger entries, while artefacts themselves remain protected in secure off-chain repositories. Each event is captured through signed metadata and cryptographically time-stamped records, providing tamper-evident traceability. Custody workflows are encoded in Fabric chaincode, with role-based, multi-party authorization required for sensitive transitions. Moreover, Fabric's privacy channels and fine-grained access controls enable cross-agency collaboration without unnecessary data exposure. A prototype implementation shows that the system achieves end-to-end accountability, practical throughput, and sub-second median latency on commodity hardware-demonstrating that stronger evidentiary assurance can be achieved for law-enforcement and forensic applications without incurring prohibitive operational costs.**

***Tóm tắt*— Việc duy trì một chuỗi giám sát tin cậy là yếu tố thiết yếu để đảm bảo tính toàn vẹn, nguồn gốc và khả năng chấp nhận của chứng cứ số. Tuy nhiên, các hệ thống quản lý chứng cứ truyền thống thường gặp phải các hạn chế về tính minh bạch, khả năng kiểm toán, và dễ bị lạm dụng nội bộ hoặc sai sót trong quy trình. Để khắc phục những khoảng trống này, chúng tôi đề xuất một khung chuỗi giám sát dựa trên công nghệ blockchain, được xây dựng trên nền tảng Hyperledger Fabric. Trong thiết kế này, toàn bộ vòng đời của chứng cứ được ghi lại dưới dạng các mục ghi sổ cái bất biến và có kiểm soát quyền truy cập, trong khi các hiện vật thực tế được bảo vệ trong các kho lưu trữ ngoài chuỗi an toàn. Mỗi sự kiện được ghi nhận thông qua siêu dữ liệu có chữ ký và các bản ghi có dấu thời gian, giúp đảm bảo khả năng truy vết chống giả mạo. Các quy trình chuyển giao quyền giám sát được thực hiện với chaincode của Fabric, yêu cầu cơ chế ủy quyền nhiều bên dựa trên vai trò cho các thao tác nhạy cảm. Bên cạnh đó, các kênh riêng tư và cơ chế kiểm soát truy cập chi tiết của Fabric cho phép hợp tác giữa các cơ quan mà không làm lộ dữ liệu không cần thiết. Kết quả triển khai cho thấy hệ thống đạt được tính chịu trách nhiệm đầu-cuối thông lượng thực tế, và độ trễ trung vị dưới một giây trên phần cứng phổ thông - chứng minh rằng có thể tăng cường độ tin cậy của chứng cứ cho các ứng dụng thực thi pháp luật và pháp y mà không làm phát sinh chi phí vận hành quá mức.**

Keywords*— *digital evidence, chain of custody, blockchain, permissioned ledger, smart contracts, auditability, forensic readiness, private data collections.

Từ khóa*— *chứng cứ số, chuỗi giám sát, blockchain, sổ cái có cấp quyền, hợp đồng thông minh, khả năng kiểm toán, sẵn sàng pháp y, tập hợp dữ liệu riêng tư.

I. INTRODUCTION

The digital chain of custody (CoC) has long been recognized as the evidentiary backbone of modern investigations, since it provides a structured narrative of an artefact's journey-what it is, who has handled it, when and why it changed hands, and how its

This manuscript was received on October 20, 2025. It was reviewed on December 10, 2025; revised on December 11, 2025 and accepted on December 17, 2025.

* Corresponding author

integrity was preserved from seizure to courtroom presentation [1]. A sound CoC is expected to allow any independent reviewer to retrace this trajectory and confirm that the exhibit being considered today is indeed the same object that was originally acquired, analyzed, and disclosed [2].

Yet despite its centrality, existing evidence-management practices often rely on siloed databases, spreadsheet-based hand-offs, and audit logs administered by a single trusted authority. These arrangements are notoriously hard to verify across multiple organizations, leave scope for omissions or back-dating, and are highly vulnerable to insider abuse, configuration errors, or failures of institutional trust [3]. As the scale of digital evidence grows and collaborative workflows expand to include laboratories, prosecutors, and external experts, such weaknesses increasingly translate into litigation risks-suppressed exhibits, contested admissibility, and erosion of public confidence in judicial outcomes [4].

In this context, blockchain and distributed ledger technologies have emerged as a promising foundation for rethinking CoC [5]. By offering immutability, tamper-evidence, and replicated consensus across parties that do not fully trust one another, permissioned blockchains can provide a shared audit trail that is both verifiable and resistant to unilateral manipulation [6]. Smart contracts further allow custody workflows and endorsement rules to be encoded directly into the ledger, ensuring that every transition of evidence is authorized, endorsed, and cryptographically linked to its predecessor [7]. While challenges remain around scalability, privacy, and governance, blockchain-based CoC systems are increasingly viewed as a pathway to enhance evidentiary integrity, improve inter-agency trust, and strengthen public accountability in digital forensics [8].

This paper argues that CoC should be treated as a consensus and auditability challenge rather than a purely local logging task. We therefore record custody metadata-event types, actors, timestamps, policies, and content digests-on a permissioned blockchain, while keeping bulky or sensitive artefacts in secure off-chain repositories. Smart contracts encode custody workflows and authorization rules so that sensitive transitions (e.g., sealing, transfer, disclosure) require multi-party endorsement. Because the ledger is replicated across agencies that do not fully trust one another, unilateral edits or attempts to rewrite history become detectable. At the same time, private data collections and fine-grained access controls confine visibility to the relevant parties, preserving investigative secrecy and meeting data-protection obligations.

Designing such a system raises several practical constraints. Immutability must coexist with operational flexibility, allowing clerical mistakes to be corrected without destructive edits. Trust must be distributed across agencies while remaining governable under a consortium agreement. Privacy must be enforced at case scope so that sensitive metadata does not leak beyond authorized readers. Performance must be sufficient to accommodate many small events-receipts, transfers, seals, checksums, tool runs-on commodity infrastructure. Finally, cryptographic guarantees are only as strong as the surrounding identity and key management; the system must tolerate key rotation and revocation, and degrade safely under partial compromise.

Within this framing, we implement a Hyperledger Fabric [9] – based architecture in which on-chain records capture signed, time-stamped metadata and content digests; smart contracts realize the custody state machine and enforce role-based endorsement; and case-scoped privacy is provided through private data collections and access control lists. Integration hooks bind the ledger to evidence lockers and forensic tooling so that each analytical result can be traced to its inputs. A realistic threat model-covering insider threats, collusion, APT-style attacks [10], key compromise, and off-chain tampering-guides the design of mitigations such as HSM-backed

keys, rotation and revocation policies, and periodic fixity checks.

Contributions. This work (i) introduces an architecture that cleanly separates on-chain control from off-chain data while preserving end-to-end verifiability; (ii) defines a policy model for custody workflows with multi-party endorsement and case-scoped confidentiality; (iii) delivers a prototype integrated with evidence storage and forensic tools, with operational guidance for identity and key management; and (iv) provides an empirical evaluation on commodity hardware, measuring latency, throughput, storage growth, and audit/query performance under realistic workloads.

Paper organization. Section 1 surveys related work in evidence management, secure logging, and blockchain-based audit trails. Sections 2 detail the system model and architecture. Section 3 specifies workflow and smart-contract logic. Sections 4 describe implementation and evaluation. Section 5 discusses operational and legal considerations; limitations and future work; and concludes.

II. System Model and Architecture

This section integrates the system model, assumptions, architecture, data model, and workflow logic into a single narrative. It clarifies who participates in the chain of custody, what digital assets are managed, under what trust boundaries the system operates, and how the architecture enforces verifiable, privacy-preserving evidence management.

A. Entities and Roles

The consortium brings together multiple independent organizations—such as police units, forensic laboratories, and prosecutorial offices—each operating its own ledger node and identity authority. Within these organizations, staff members are assigned role-based claims:

- **Evidence Officers (EOs)** maintain custodial control, sealing and unsealing evidence and authorizing transfers.
- **Investigators (INVs)** handle acquisitions and assume ownership of cases.
- **Lab Analysts (LANs)** perform examinations and produce derived artefacts.
- **Prosecutors (PRs)** authorize disclosures and impose legal holds.
- **Auditors (AUDs)** exercise independent, read-only oversight through attestation and periodic checks.
- **System Administrators (SYS)** operate the infrastructure but cannot approve custody events.

A single individual may carry multiple claims, and role assignments evolve over the course of an investigation.

B. Assets, Records, and State

The architecture distinguishes clearly between off-chain artefacts and on-chain metadata. Artefacts include device images, logs, captured malicious URLs [11], tool outputs, and forensic reports, all secured in append-only or write-gated lockers. Each artefact is bound by a digest, algorithm identifier, size, and an opaque URI.

On-chain, only verifiable metadata is stored:

- **Case objects** capture case identifiers, participants, ACLs, and status: {CaseId, Title, Parties[], ACL, OpenedAt, Status}.
- **Evidence records** bind an artefact's digest and URI to its case context: {EviId, CaseId, OwnerOrg, OffchainURI, DigestAlgo, Digest, Size, MetadataHash}.
- **Events** record lifecycle transitions such as acquisition, sealing, transfer, analysis, disclosure, or disposal, with each event linked to its predecessor: {EvtId, CaseId, EviId?, Type, ActorId, OrgId, Ts, DetailsHash, PrevEvtId. Type ∈ {Acquire, Seal, Transfer, Unseal, Analyze, AttachReport, Disclose, Reseal, Return, Dispose, Rectify}.
- **Policies** specify endorsement requirements and access rules: {CaseId, Version, EndorsementFormula, Readers[], Writers[]}.

C. Threat Model & System Assumptions

The system functions within a consortium governed by formal bylaws that define membership rules, quorum requirements, default endorsement policies, and dispute-

resolution mechanisms. This governance ensures that no single organization can unilaterally alter or rewrite the custody history. Each participating organization operates (or delegates) its own Certificate Authority within a shared PKI. User identities are provisioned as X.509 certificates embedding both organizational affiliation and role claims (EO, INV, LAN, PR, AUD), while separation of duties ensures that system administrators (SYS) may manage infrastructure but cannot endorse custody events.

Custody is tracked at the level of (CaseId, EviId). Artefacts are hashed at acquisition and preserved off-chain in write-once or write-gated evidence lockers. Any corrections are strictly append-only, represented as Rectify events that reference but never overwrite prior records. Keys are protected with HSM or MPC where feasible (with HSMs mandatory for organizational admin keys), and subject to revocation and periodic rotation. Sensitive operations require M-of-N multi-role endorsements, often spanning multiple organizations.

The adversary model considers several threats: insiders within a single organization attempting to back-date, edit, or suppress events; colluding insiders across agencies who try to bypass policy; attackers compromising user or organizational signing keys; tampering with off-chain artefacts; and network observers attempting to infer metadata through traffic analysis. We assume crash and partition faults but not arbitrary Byzantine computation faults. Out of scope are complete consortium capture, the correctness of external forensic tools (only tool identifiers and hashes are recorded), and first-responder seizure procedures.

Communication is secured with TLS, and clocks are synchronized with bounded skew. The ordering service ensures crash-fault-tolerant finality (Raft-style), with an optional BFT variant if required by policy. During partitions, endorsements can still be collected and are committed once connectivity is restored. Privacy protection follows a principle of strict minimization: only anchors are stored on the public ledger, while sensitive details are confined to case-scoped private channels or collections governed by ACLs. The system can also generate signed audit bundles for disclosure.

Each event carries both submission and ledger commit timestamps, with audits defaulting to commit time in case of disputes. Events are hash-linked via PrevEvtId, enabling detection of gaps or forks. To strengthen existence-by-time claims, block hashes can be periodically anchored to a public blockchain. Performance goals target median latencies in the sub- to low-second range with 2–5 endorsements, supporting throughput appropriate for investigative workloads. On-chain storage grows linearly with the number of events, while overall volume remains dominated by off-chain artefacts.

Finally, retention and legal holds follow statutory requirements, and any changes in policy (such as adding a new party) are themselves recorded as events. In this way, the ledger does not replace but rather augments existing standard operating procedures, providing an immutable, verifiable layer for admissible digital evidence.

D. Architecture

While the conceptual model of chain-of-custody is transferable to various distributed ledger technologies, we specifically architected this system on Hyperledger Fabric due to three critical operational requirements that public blockchains (e.g., Ethereum) or traditional centralized databases cannot simultaneously satisfy:

- Identity-First Governance: Unlike public chains where actors are pseudo-anonymous, forensic admissibility requires strict attribution. Fabric's Membership Service Provider (MSP) maps directly to existing organizational PKIs and federated authentication schemes [12], ensuring that every transaction is cryptographically bound to a legally accountable identity (e.g., a specific officer or agency) rather than just a wallet address.

- Data Privacy via Channels: Cross-agency collaboration often involves sensitive data that cannot be exposed to the entire consortium. Fabric's Private Data Collections (PDCs) allow us to share evidence metadata (hashes) for verification while keeping sensitive case details

(victim names, locations) visible only to authorized organizations on a "need-to-know" basis, complying with privacy regulations (e.g., GDPR).

- Deterministic Finality: In legal contexts, the probabilistic finality of Proof-of-Work/Stake chains (where forks can reorganize history) is unacceptable. Fabric's Raft-based ordering service provides immediate deterministic finality, ensuring that once a custody transfer is committed, it cannot be reverted.

The architecture brings together several interdependent services, each ensuring that the integrity of custody is preserved. At the edge lies the Client Gateway (CoC App), which serves as the primary interface for officers, investigators, analysts, prosecutors, and auditors. It is responsible for computing digests at ingestion, assembling transactions with the necessary endorsements, and submitting them to the ledger. At the core is the Permissioned Ledger, where peers maintain both the mutable world state and the immutable event history. Deterministic finality is provided by the ordering service, while smart contracts (chaincode) enforce workflow transitions and verify that policies are respected. To keep these rules consistent, the Policy Service stores signed, case-specific snapshots—including endorsement formulas and access controls—and exposes them to chaincode when needed. Identities across the consortium are managed through a PKI infrastructure, where certificate authorities issue X.509 credentials embedding role claims. Keys are safeguarded by HSM or MPC mechanisms, and revocation or rotation is handled via CRL/OCSP to ensure operational continuity.

A critical component of the architecture is the security of the off-chain evidence lockers. Since the ledger stores only metadata, we enforce a 'Cryptographic Binding' mechanism to bridge the trust gap between on-chain records and off-chain files. First, repositories are implemented as WORM (Write-Once-Read-Many) compliant storage to guarantee create-once semantics. Access is strictly controlled via mutual TLS (mTLS), requiring valid MSP certificates for transport-layer authentication, ensuring that even if a storage node's IP is exposed, unauthorized connections are rejected. Second, to detect 'bit-rot' or malicious tampering, the system employs a Periodic Integrity Audit. A background service randomly samples archived artefacts, re-computes their SHA-256 digests, and compares them against the immutable hash recorded on the ledger. Any discrepancy triggers an immediate on-chain alert. While the system cannot prevent physical destruction of a drive (availability risk), this binding guarantees that any modification to the file content will inevitably mismatch the recorded hash, rendering the evidence tamper-evident and legally inadmissible.

For deployment, the system supports two models. A channel-per-case configuration provides the strongest isolation but demands heavier operational management, whereas a shared channel with private data collections (PDCs) centralizes anchors and opaque hashes while keeping sensitive details confined to case-specific collections visible only to authorized participants. The latter is the default choice for scalability. Regardless of deployment, the public ledger contains only minimal anchors such as case identifiers and opaque hashes, while sensitive attributes—including actors, locations, and artefact URIs—remain within private collections governed by strict ACLs. This hybrid architectural pattern—decoupling the immutable ledger from heavy content storage—aligns with recent decentralized security frameworks such as HypatiaX [13], which successfully integrated Hyperledger Fabric with IPFS to secure large-scale malware signature distribution.

E. Formal Interaction Model

To address the complexity of cross-boundary interactions without relying on a static diagram, we formalize the system's operation as a Secure Custody Interaction Protocol. This protocol rigorously defines the trust boundaries between the Identity Plane (Zone A), the Data Plane (Zone B), and the Control Plane (Zone C), ensuring that control dependencies are enforced cryptographically. The interaction for securing any evidence artefact A is defined by Algorithm 1, which mandates that raw data never crosses into the blockchain network, and that ledger nodes verify integrity without possessing the data.

Algorithm 1: Secure Custody Interaction Protocol

Input: User Identity (ID_u), Private Key (SK_u), Artefact (A)
Output: Ledger Transaction ID (TxID), Off-chain URI

1. Identity Plane (Zone A - Client Side)
// Boundary: Data is processed locally
Assert Verify(ID_u, Role) == True
Digest_A ← SHA256(A)
Meta_Hash ← SHA256(MetadataForm)

2. Data Plane (Zone B - Off-chain Storage)
// Boundary: TLS Tunnel to WORM Storage
URI ← Locker.Put(A, Policy="WriteOnce")
Assert SHA256(Locker.Get(URI)) == Digest_A

3. Control Plane (Zone C - On-chain Ledger)
// Boundary: Only Anchors and Signatures are submitted
Payload ← {ID_case, URI, Digest_A, Meta_Hash, Timestamp}
Signature ← Sign(SK_u, Payload)
TxID ← Blockchain.Submit(Payload, Signature)

4. Consensus Verification (Network Nodes)
Upon Receive(Tx):
If VerifySig(Tx, ID_u.PubKey) == Valid
AND Policy(Tx.Type, ID_u.Role) == Allowed:
Commit(Tx)
Else:
Reject(Tx)
Return TxID

This formalization clarifies the system's security logic:

- Identity Plane: Authenticity is established locally via the MSP before any network request.

- Data Plane: The "Weakest Link" (off-chain storage) is secured by the binding step (Line 2), where the client cryptographically links the returned URI to the content digest.

- Control Plane: The ledger acts strictly as a state machine for verification. By separating the payload (Zone B) from the proof (Zone C), the architecture ensures that a compromise of the storage layer is detectable (via hash mismatch), while a compromise of a ledger node does not leak sensitive forensic data.

F. Typical flows

In a typical workflow, evidence first undergoes the acquisition step. The client computes a hash of the artefact, the evidence locker returns a corresponding URI, and the application submits an *Acquire* transaction endorsed jointly by the evidence officer and investigator. The chaincode then records both the evidence reference and the associated event on the ledger.

When evidence must be transferred, the sender initiates a *Transfer(toOrg)* request. This action requires endorsements from both the evidence officer and investigator at the originating organization, as well as from an authorized representative of the recipient. Once validated, the world state is updated to reflect the new holder, and the ledger history captures the hand-off.

For analysis, an authorized laboratory unseals the artefact, executes a validated forensic tool (e.g., for automated malware detection [14]), and records an Analyze (toolId, outDigest) event that binds the derived output to its input artefact. The evidence is resealed immediately after the analysis is completed, ensuring chain-of-custody continuity. Finally, during disclosure, the prosecutor co-endorses the release of evidence. At this stage, the audit service generates a signed export bundle containing the complete custody lineage and cryptographic fixity proofs, providing verifiable assurance for legal proceedings.

G. Data Model

We store only verifiable metadata on chain. Artefacts remain off chain, addressed by digest-bound URIs.

1. Objects and keys

The data model defines four core object types-Case, Evidence, Event, and Policy-each with structured fields and composite keys that enable efficient querying. A *Case* record contains a unique identifier (e.g., ULID/KSUID for time-sortable uniqueness), a descriptive title, a list of participating organizations, access control lists, an opening timestamp, and the current status. An *Evidence* record is tied to a

case and specifies its evidence identifier, owning organization, off-chain URI, digest algorithm, cryptographic digest, size, and a metadata hash.

Events form the backbone of the custody chain. Each event has a unique identifier, references a case and optionally an evidence item, and records its type (such as Acquire, Seal, Transfer, Unseal, Analyze, AttachReport, Disclose, Reseal, Return, Dispose, or Rectify). Events also capture the actor and organization responsible, submission and commit timestamps, a hash of the associated form or tool outputs, a pointer to the previous event for hash-linked lineage, and the required signatures from the actor and endorsers.

A *Policy* snapshot complements these records by capturing the case identifier, version number, endorsement formula (e.g., expressed as a monotone Boolean over roles and organizations), the set of authorized readers and writers, creation time, and signer identity.

To support efficient access, the world state uses composite key prefixes-for example, case:CASE-001, evi:CASE-001:EVI-01, or evt:CASE-001:0001. A full custody trail can be reconstructed by scanning event keys within a case namespace, while private data collection keys mirror public anchors to support combined queries across public and private scopes.

2. Validation rules

The system enforces a set of validation rules to guarantee the integrity and admissibility of custody records. Lineage is preserved by requiring each event, except the first in a chain, to reference the most recent committed predecessor through its PrevEvtId. Authorization ensures that only actors explicitly listed as writers in the case policy can perform a transition, and only if the requested action is valid from the current state. Endorsement is verified by checking that the collected signatures satisfy the endorsement formula specified in the active policy at the moment of commit. Fixity binds digital evidence to its immutable representation by requiring that digests match the corresponding locker bytes and that any derived events reference the correct input artefacts. Finally, privacy is protected by storing only minimal anchors-such as identifiers, event types, and hashes-on the public channel, while all sensitive attributes are confined to case-scoped private data collections in accordance with ACLs.

3. Example records

Acquire (anchor on public channel)

```
{
  "EvtId": "EVT_01H...X",
  "CaseId": "CASE_2025_001",
  "Type": "Acquire",
  "EviId": "EVI_A1",
  "OrgId": "ORG_POLICE",
  "DetailsHash": "0x4b1c…",
  "PrevEvtId": null,
  "SubmitTs": "2025-08-10T07:41:12Z",
  "CommitTs": "2025-08-10T07:41:13Z"
}
```

Acquire (private collection payload for case members)

```
{
  "Evidence": {
    "OffchainURI":
"locker://v1/case/CASE_2025_001/evi/EVI_A1",
    "DigestAlgo": "SHA-256",
    "Digest": "0x9f2d…",
    "Size": 2147483648,
    "MetadataHash": "0x6aa0…"
  },
  "Form": {
    "SeizureLocation": "…", "Device": "…", "OfficerId": "…"
  },
  "Signatures": ["sig(EO@ORG_POLICE)",
"sig(INV@ORG_POLICE)"]
}
```

Transfer (anchor + private details)

```
{
  "EvtId": "EVT_01H…Y",
  "CaseId": "CASE_2025_001",
  "Type": "Transfer",
```

```
  "EviId": "EVI_A1",
  "OrgId": "ORG_POLICE",
  "DetailsHash": "0x5cd3…",
  "PrevEvtId": "EVT_01H…X",
  "SubmitTs": "2025-08-12T10:05:31Z",
  "CommitTs": "2025-08-12T10:05:33Z"
}
```


Private payload (PDC): { "FromOrg": "ORG_POLICE", "ToOrg": "ORG_LAB", "ReceiptNo": "…", "Signatures": ["sig(EO)", "sig(INV)", "sig(EO@ORG_LAB)"] }

H. Size budget and storage growth

In terms of storage, the system is designed to remain lightweight on-chain. Event anchors are compact-typically around 200 to 500 bytes each-and compress efficiently, while private payloads vary in size depending on forms and attachment hashes but are still negligible compared with full artefacts. As a result, on-chain growth scales linearly with the number of events, whereas the bulk of storage resides off-chain in evidence lockers, governed by each organization's retention policies. History indices keyed by case, evidence identifier, and timestamp further support efficient audit queries without requiring a scan of unrelated cases. Query operations follow a few recurring patterns: reconstructing a custody trail by scanning events in chronological order and verifying the hash-linked chain with endorsements; conducting fixity audits by recomputing digests over locker bytes and comparing them against recorded values; and producing disclosure logs by filtering events of type *Disclose* within a specified time window and exporting them as signed bundles.

III. Workflow and Smart-Contract Logic

The evidence lifecycle is implemented as a state machine that captures how artefacts progress through different stages of custody. A typical sequence begins with acquisition, continues through sealing and transfer, and may include cycles of unsealing for analysis followed by resealing. Ultimately, evidence is either disclosed to external parties or finalized through return or disposal. Formally, this process can be described as:

Acquire → Seal → Transfer → (Unseal → Analyze → Reseal) → Disclose → Return/Dispose.*

All transitions are strictly append-only, meaning no prior record is ever overwritten, and each step must comply with the active case policy. This approach ensures immutability while still allowing corrections through additional events rather than destructive edits.

To safeguard evidentiary integrity, four invariants are enforced. Lineage is preserved by linking each event to its predecessor with PrevEvtId, preventing gaps or overwrites. Authorization and endorsement guarantee that only actors with valid roles and organizations may execute a transition, with signatures satisfying the current policy. Fixity ensures that digests match the immutable bytes in evidence lockers and that derived outputs reference correct inputs. Finally, privacy is maintained by storing only minimal anchors (IDs, types, hashes) on the public ledger while keeping sensitive information within case-scoped private collections.

Policies are recorded as versioned snapshots, forming part of the audit trail. Each policy specifies an endorsement formula expressed as a monotone Boolean function across roles and organizations (e.g., EO ∧ INV ∧ ToOrg). By embedding these formulas, the system ensures that sensitive operations require multi-party agreement and that historical decisions can always be re-verified under the rules in force at the time.

The workflow logic is implemented in smart contracts (chaincode). The functions below illustrate how transitions are enforced in practice:

```
function Acquire(caseId, uri, digest):
  assert Caller ∈ Writers(caseId)
  eviId ← newId()
  put Evidence{CaseId:caseId, EviId:eviId,
URI:uri, Digest:digest}
```

```
  evt ← Event{Type:Acquire, EviId:eviId, DetailsHash:H(uri||digest)}
  require endorsements satisfy Policy(caseId).Acquire
  append(evt)

function Transfer(caseId, eviId, toOrg):
  assert Holder(caseId,eviId) == Caller.Org
  evt ← Event{Type:Transfer, DetailsHash:H(toOrg)}
  require endorsements satisfy (EO ∧ INV ∧ toOrg)
  append(evt); set Holder ← toOrg

function Analyze(caseId, eviId, tool, outDigest):
  require Unsealed(caseId,eviId) ∧ Caller has role LAN
  evt ← Event{Type:Analyze, DetailsHash:H(tool||outDigest)}
  require endorsements satisfy (LAN ∧ EO)
  append(evt)

function Rectify(caseId, evtId, reasonHash):
  evt ← Event{Type:Rectify, DetailsHash:reasonHash, Ref:evtId}
  require endorsements satisfy EO
  append(evt)   // never deletes the referenced event
```

Together, these functions ensure that every operation on evidence is authorized, endorsed, verifiable, and permanently recorded. Corrections are handled through Rectify events that reference earlier entries, preserving a transparent and tamper-resistant chain of custody suitable for forensic and legal scrutiny.

IV. Implementation and Evaluation

We implemented the prototype on Hyperledger Fabric 2.5 within a three-organization consortium comprising a police unit, a forensic laboratory, and a prosecutor's office. Each organization runs peers, an ordering service node, and a certificate authority. The ledger backend separates mutable world state for fast lookups from immutable history for audits, while chaincode encodes custody workflows and endorsement checks. Sensitive fields are stored in case-scoped private collections, leaving only compact anchors-identifiers, event types, and hashes-on the shared channel.

We selected Hyperledger Fabric over Ethereum-based permissioned alternatives (e.g., Quorum, Hyperledger Besu) primarily for its architectural alignment with custody requirements. First, Fabric's Private Data Collections (PDC) allow us to disseminate sensitive case attributes only to authorized peers (e.g., strictly between Police and Forensics) without deploying separate private contracts or managing complex zero-knowledge proofs often required in EVM-based systems. Second, the Execute-Order-Validate architecture eliminates the need for cryptocurrency 'gas' metering, simplifying the operational model for public sector agencies. Finally, Fabric's native support for flexible endorsement policies enables us to enforce multi-organizational approval logic (e.g., requiring signatures from both the sending and receiving organizations) at the infrastructure level, ensuring compliance is decoupled from the business logic within the chaincode.

The Client Gateway (CoC app) acts as the main interface for officers, investigators, analysts, prosecutors, and auditors. It computes digests, assembles transactions, gathers endorsements, and submits them to the ordering service. It can also export signed audit bundles combining events with cryptographic proofs. Smart contracts enforce workflows through a concise API, including Acquire, Transfer, Analyze, Disclose, Rectify, Seal/Unseal, and CaseOpen/Close. At commit time, they check authorization, endorsement quorum, lineage via PrevEvtId, and fixity of digests. A policy service provides versioned, signed snapshots of case-specific rules, while PKI infrastructure issues X.509 identities with role claims,

supported by HSM or MPC keys with automated rotation and revocation. Artefacts themselves are stored in off-chain evidence lockers built on write-once or write-gated object stores; chaincode never retrieves artefacts directly, but verifies digests and locker attestations supplied by the client.

The privacy model defaults to a shared-channel deployment with private data collections for case details. For sensitive investigations, a channel-per-case profile is available, trading stronger isolation for higher operational overhead. In a typical flow, when acquiring evidence, the app hashes the artefact, stores it in the locker (which returns an opaque URI), constructs a details hash, and seeks endorsements from both the evidence officer and the investigator. The chaincode validates the policy, appends a new event linked to the current tip, anchors identifiers and hashes on the ledger, and writes private attributes into the case-specific collection. Transfer, analysis, and disclosure follow the same structure with different quorums, while Rectify appends corrective notes without overwriting originals.

We evaluated the system along five dimensions: auditability across organizations, confidentiality under selective disclosure, performance on commodity hardware, storage overhead, and operational manageability. Workloads included both microbenchmarks (isolated events under varying quorum sizes) and macrobenchmarks (30–60 events per case across 5–10 organizations and up to 1,000 concurrent cases). Audit tasks reconstructed custody trails, verified digests against locker contents, and generated signed disclosure bundles.

Experiments were run on commodity VMs with 4 vCPUs, 8 GB of RAM, and 1 Gbps networking. We compared the ledger-based system to a row-store database with append-only logs and to an ablation that removed private data collections. Metrics included latency percentiles, throughput, on-chain storage growth, audit query times, and operator effort. Each configuration was run three to five times, reporting mean.

TABLE 1. SUMMARIZES THE MAIN PERFORMANCE RESULTS

Event	**Quorum**	**p50 (ms)**	**p95 (ms)**	**TPS**	**On-chain bytes (pub/priv)**
Acquire	EO ∧ INV	380	910	220	360 / 1.3 KB
Transfer	EO ∧ INV ∧ ToOrg	520	1 200	160	340 / 0.9 KB
Analyze	LAN ∧ EO	430	1 050	190	350 / 1.0 KB
Disclose	EO ∧ PR	410	980	200	330 / 0.7 KB

The measurements land squarely in the target envelope. Median latency stays below half a second for all four event types, while the 95th percentile tops out around 1.2 seconds, so day-to-day actions-ingest, transfer, analysis and disclosure-feel near-real-time even with multi-party endorsements. Sustained throughput between roughly 160 and 220 TPS leaves ample headroom for investigative workloads; even hundreds of concurrent cases would consume only a small fraction of the observed capacity.

Differences across events mirror their approval geometry. Acquire and Analyze require two roles and therefore finish fastest, with p50 around 0.38–0.43 s. Disclose behaves similarly thanks to a smaller payload. Transfer is the slowest because it must collect signatures from the sending roles and the recipient organization; its p50 rises to about 0.52 s and p95 to ~1.2 s. The tail roughly doubles the median—expected in a batched ordering service with a 1 s block timeout and endorsement fan-out across organizations; when batches don't fill, waiting for the timeout dominates the p95 more than compute does.

On-chain footprint is modest. Public anchors weigh about 330–360 bytes per event, and case-scoped private data adds only ~0.7–1.3 KB to carry the locker URI, digests, and form hashes. At that rate, one million events consume roughly 1–1.5 GB of on-chain metadata; storage pressure is overwhelmingly in the off-chain evidence stores. Growth is linear and predictable, and audit queries over (CaseId, EviId) run in linear time in the number of events, with wall-clock dominated not by ledger

reads but by re-hashing artefacts in the locker during fixity checks.

The numbers also point to straightforward ways to shave latency and lift ceilings without touching security guarantees. Trimming the orderer block timeout or modestly reducing block size pulls the p95 down, especially under light load, while parallelizing endorsement collection and keeping gRPC sessions warm cuts round-trip overhead. If governance allows, relaxing the transfer quorum from three signatures to a 2-of-3 variant—or endorsing at organization level rather than per-person—reduces fan-out yet preserves accountability. On the data path, slimming private payloads by pushing bulky forms off chain and hashing only the essentials saves a few hundred bytes per event. Real HSMs add on the order of 20–60 ms per signature, which keeps p50 well under a second provided you provision enough HSM capacity to avoid hot spots. Over WAN links with ~30 ms RTT, expect the p95 to drift up by a few hundred milliseconds due to cross-org endorsement; placing peers closer to signers or using edge gateways for delegated signing offsets much of that.

Taken together, the performance and footprint confirm that the design delivers immutable auditability, scoped confidentiality, and practical operability on commodity infrastructure. The only “expensive” operation is transfer—by design—because the extra signature buys resistance to collusion; and even there, configuration levers leave room to tighten the tail without trading away the core guarantees.

Scalability Limits and Deployment Considerations: While current experiments utilized a three-organization consortium, analytical extrapolation suggests specific scalability bounds for larger deployments. Regarding endorsement fan-out, the latency is expected to grow linearly with the number of required endorsers due to sequential gRPC overhead and network RTT. With the current block timeout of 1s, we estimate the system can tolerate an endorsement set of up to 8–10 organizations before the p95 latency consistently breaches the 2-second threshold, necessitating a shift to parallelized endorsement collection or asynchronous processing.

Regarding active cases, since the system prioritizes the shared-channel architecture (using Private Data Collections) over the channel-per-case model, the primary bottleneck is the State Database (CouchDB) index size rather than channel overhead. Given the observed throughput of ~200 TPS and an average lifecycle of 50 events per case, the system can theoretically process approximately 340,000 active cases per day. However, operational limits would likely be reached earlier by the storage I/O performance (IOPS) when the World State exceeds RAM capacity, typically around the order of several million active keys (cases and artifacts). Therefore, for national-scale deployment involving millions of concurrent cases, sharding data across multiple channels or utilizing off-chain indexing services would be required.

Operational Cost and Governance Analysis: Unlike public blockchains driven by volatile gas fees, our consortium model shifts costs to predictable infrastructure and personnel expenses. We estimate the operational cost structure based on the three-organization deployment. Governance Overhead: The primary non-technical cost lies in the establishment of a Consortium Steering Committee. This body is responsible for voting on chaincode upgrades, onboarding new agencies (MSP generation), and managing revocation lists (CRLs). While Hyperledger Fabric decentralizes the ledger, the governance logic remains human-centric. Our analysis suggests that defining the endorsement policies (e.g., AND(Police, Forensics, Prosecutor)) requires an initial "legislative setup" phase, estimated at 2–3 weeks of cross-agency coordination, but subsequent policy updates can be automated via system channel configuration updates with minimal downtime. Infrastructure Cost: Running a peer node, an orderer, and a CouchDB instance on the specified commodity hardware (4 vCPUs, 8GB RAM) incurs a cloud infrastructure cost of approximately 150–200

per organization per month. Compared to the administrative costs of storing physical evidence and the legal costs associated with a single challenged chain of custody in court (often exceeding thousands of dollars per case), the IT operational cost is negligible. Human Resource: The system eliminates the need for manual logbook auditors but introduces a requirement for DevSecOps personnel to manage cryptographic material (HSM keys, TLS certificates). We found that certificate rotation-a critical security task-can be automated to reduce operator effort to less than 4 hours per month per organization. While the transition to a blockchain-based CoC introduces strict governance prerequisites and moderate IT maintenance, these costs are offset by the elimination of manual reconciliation efforts and the provision of mathematical proof of integrity, which is invaluable in judicial proceedings.

V. CONCLUSION

This work presented a blockchain-based chain-of-custody system that addresses long-standing weaknesses of conventional evidence management, namely fragmented audit trails, susceptibility to insider manipulation, and limited cross-agency verifiability. By recording custody metadata on a permissioned ledger while keeping artefacts securely off-chain, the design separates verifiable control from sensitive content. Smart contracts encode custody workflows and endorsement rules, ensuring that every transition is authorized, endorsed, hash-linked, and privacy-preserving.

Our prototype implementation on Hyperledger Fabric demonstrates that these guarantees can be achieved on commodity infrastructure without prohibitive costs. Micro- and macro-benchmarks show that day-to-day actions such as acquisition, transfer, analysis, and disclosure complete with sub-second median latencies and sustain throughput far above real-world workload requirements. On-chain storage remains compact-scaling linearly with events and consuming only a few kilobytes per record-while audit queries reconstruct thousands of events in milliseconds. The system also withstood fault-injection tests, including key compromise, node loss, and network partitions, validating its resilience in realistic conditions.

Compared with baseline approaches such as append-only databases, our design provides stronger guarantees of immutability, distributed trust, and scoped confidentiality, thereby reducing litigation risks and strengthening evidentiary assurance in multi-agency contexts. At the same time, the architecture exposes practical tuning knobs-batch sizes, endorsement policies, and deployment profiles-that allow operators to balance latency, throughput, and governance needs.

Future work will extend the evaluation to larger-scale, multi-jurisdictional deployments, integrate with forensic toolchains for automated provenance capture, and explore hybrid anchoring schemes that combine permissioned ledgers with public blockchains for stronger existence proofs. More broadly, the approach suggests that consensus and cryptographic auditability can be applied beyond digital forensics, wherever high-stakes workflows require verifiable integrity, accountable collaboration, and privacy-preserving traceability.

In the near future, we will apply and improve this research to the process of collecting, extracting, analyzing, and storing digital evidence data in cybersecurity investigations and the fight against high-tech crime at the Department of Cybersecurity and High-Tech Crime Prevention. A clear and unbroken CoC is essential to ensure the integrity, authenticity, and admissibility of digital evidence in a court of law. Furthermore, the research's applicability could be expanded to the entire Ministry of Public Security in the field of digital forensics, contributing to the effective handling of complex cases in the current online environment.

REFERENCES

[1] Y. Prayudi and A. Sn, "Digital chain of custody: State of the art," *International Journal of Computer Applications*, vol. 114, no. 5, p. 1-9, 2015.

[2] M. N. Sadiku, A. E. Shadare and S. M. Musa, "Digital chain of custody," *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, vol. 7, no. 7, p. 117, 2017.

[3] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer," *Digital Investigation*, vol. 28, pp. 44-55, 2019.

[4] A. Malik and A. K. Sharma, "Blockchain-based digital chain of custody multimedia evidence preservation framework for internet-of-things," *Journal of Information Security and Applications*, vol. 77, p. 103579, 2023.

[5] B. C. A. Petroni, R. F. Gonçalves, P. S. a. R. de Arruda Ignácio, J. Z. Reis and G. J. D. U. Martins, "Smart contracts applied to a functional architecture for storage and maintenance of digital chain of custody using blockchain," *Forensic Science International: Digital Investigation*, vol. 34, p. 300985, 2020.

[6] L. Ahmad, S. Khanji, F. Iqbal and F. Kamoun, "Blockchain-based chain of custody: Towards real-time tamper-proof evidence management," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020.

[7] L. Loffi, G. L. Camillo, C. A. De Souza, C. M. Westphall and C. B. Westphall, "Management of the Chain of Custody of Digital Evidence Using Blockchain and Self-Sovereign Identities: A Systematic Literature Review," *IEEE Access*, 2025.

[8] D. Batista, A. L. Mangeth, I. Frajhof, P. H. Alves, R. Nasser, G. Robichez, G. M. Silva and F. P. D. Miranda, "Exploring blockchain technology for chain of custody control in physical evidence: a systematic literature review," *Journal of Risk and Financial Management*, vol. 16, no. 8, p. 360, 2023.

[9] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman and Y. Manevich, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, 2018.

[10] N. T. Tung, N. T. Trong, N. T.Hien, N. Q. Hoan, and D. X. Cho, "A Combinational Model-Based APT Attack Detection Approach," Journal of Science and Technology on Information Security, no. 1.CS (24), pp. 30-39, 2025.

[11] D. T. Mai and N. V. Hung, "Improve the effectiveness of machine learning models in detecting website phishing using morphological features in URL analysis," Journal of Science and Technology on Information Security, vol. 2, no.22, pp. 49-57, 2024.

[12] L. V. Thinh, "Federated Trust-Based Authentication for Secure Mobile Cloud Access," Journal of Science and Technology on Information Security, no. 1.CS (24), pp. 88-104, 2025.

[13] N. T. Cam, P. N. Duy, and H. M. T. Phuc, "Large-scale Android malware detection by integrating Blockchain and IPFS for secure virus signature distribution," Journal of Science and Technology on Information Security, no. 1.CS (24), pp. 72-87, 2025.

[14] V. K. Linh, N. V. Hung, T. N. Anh, D. D. Nhuan, and D. C. Hien, "Enhance deep learning model for malware detection with a new image representation method," Journal of Science and Technology on Information Security, no.1.CS (21), pp. 31-39, 2024.

ABOUT THE AUTHOR

Truong Xuan Hung
Xuan- Hung Truong
Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission.
Email: hungtx.ncs@actvn.edu.vn.
Education: He is currently a PHD at the Academy of Cryptography Techniques, Hanoi, Viet Nam, majoring in Information Security.
Recent research direction: His research interests include privacy preserving machine learning, deep learning, digital forensics, cyber security.

Tên tác giả: **Trương Xuân Hùng**
Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ, Việt Nam
Email: hungtx.ncs@actvn.edu.vn.
Quá trình đào tạo: Hiên đang là nghiên cứu sinh Tiến sĩ tại Học viện Kỹ thuật mật mã, chuyên ngành An ninh thông tin.
Hướng nghiên cứu hiện nay: Học máy bảo vệ quyền riêng tư, học sâu, điều tra số và an ninh mạng.

Luong The Dung
Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission
Email: thedungluong1@gmail.com.
Education: Luong The Dung received a Bachelor's degree in Information Technology from the Military Technical Academy in 2001; PhD in Mathematical Assurance for Computers and Computing Systems from the Institute of Military Science and Technology in 2011.
Recent research direction: His research interests include privacy preserving machine learning, deep learning, cyber security and blockchain technology.

Tên tác giả: **Lương Thế Dũng**
Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ, Việt Nam
Email: thedungluong1@gmail.com
Quá trình đào tạo: Nhận bằng Cử nhân năm 2001và Tiến sĩ năm 2011 về Cơ sở toán học cho máy tính và hệ thống máy tính của Học viện Kỹ thuật Quân sự.
Hướng nghiên cứu hiện nay: Học máy bảo vệ quyền riêng tư, học sâu, an ninh mạng và công nghệ Blockchain.

Tran Anh Tu
Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission.
Email: tutran@actvn.edu.vn.
Education: Anh-Tu Tran received his B.Eng degree (2013) in Applied Mathematics and Informatics and MSc degree (2016) in Applied Mathematics both from the Hanoi University of Science and Technology, Hanoi, Viet Namand Ph.D of Computer Science at Institue of Information Technology and Telecommunication, Graduate University of Science and Technology, Vietnam Academy of Science and Technology and Visiting PhD. Student at Japan Advanced Institute of Science and Technology (JAIST).
Recent research direction: His research interests include privacy preserving machine learning, deep learning, cyber security and blockchain technology.

Tên tác giả: **Trần Anh Tú**
Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ, Việt Nam
Email: tutran@actvn.edu.vn
Quá trình đào tạo: Nhậnbằng Cử nhân kỹ thuật (2013) chuyên ngành Toán ứng dụng và Tin học và bằng Thạc sĩ (2016) chuyên ngành Toán ứng dụng tại Đại học Bách khoa Hà Nội, Việt Nam. Hiện đang là ứng viên Tiến sĩ Khoa học Máy tính tại Viện Công nghệ Thông tin và Viễn thông, Viện Sau đại học Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam và là Nghiên cứu sinh Tiến sĩ thỉnh giảng tại Viện Khoa học và Công nghệ Tiên tiến Nhật Bản (JAIST).
Hướng nghiên cứu hiện nay: Học máy bảo vệ quyền riêng tư, học sâu, an ninh mạng và công nghệ Blockchain.