Proof of Compliance (PoC): A Consensus Mechanism to Verify the Compliance with Informed Consent Policy in Healthcare

Md Al Amin
Colorado State University
Fort Collins, Colorado, USA
Alamin@colostate.edu

Hemanth Tummala
Colorado State University
Fort Collins, Colorado, USA
Hemanth.Tummala@colostate.edu

Rushabh Shah
Colorado State University
Fort Collins, Colorado, USA
Rushabh.Shah2@colostate.edu

Indrajit Ray
Colorado State University
Fort Collins, Colorado, USA
Indrajit.Ray@colostate.edu

Abstract

Healthcare industries are subject to various laws and regulatory oversight, just like other industries, such as pharmaceuticals, telecommunications, education, and financial services. Compliance with these regulations is essential for the organization's operation and growth. To help organizations detect early non-compliance issues, this paper proposes a consensus mechanism, Proof of Compliance (PoC), where a set of distributed, decentralized, and independent auditor nodes perform audit operations to determine the compliance status of any logical operations or accesses that have already been approved, granted, or executed in the system. The Proof of Compliance consensus mechanism helps organizations minimize compliance challenges. Organizations can consider PoC outputs to take further actions to reduce non-compliance cases and avoid compliance issues and business losses. The PoC reports do not support final regulatory compliance certification. However, it is possible if one or more multiple audit nodes are deployed and maintained in the consensus mechanism by the corresponding regulatory, government, or compliance authority.

CCS Concepts

• **Security and privacy → Systems security**;

Keywords

Policy, Enforcement, Provenance, Compliance, Auditor, Regulatory Agency, Blockchain, Consensus Mechanism, Smart Contract.

ACM Reference Format:
Md Al Amin, Hemanth Tummala, Rushabh Shah, and Indrajit Ray. 2025. Proof of Compliance (PoC): A Consensus Mechanism to Verify the Compliance with Informed Consent Policy in Healthcare. In *Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy (CODASPY '25), June 4–6, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3714393.3726512

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

CODASPY '25, Pittsburgh, PA, USA

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1476-4/2025/06
https://doi.org/10.1145/3714393.3726512

1 Introduction

Electronic health records (EHRs) have emerged as a cornerstone in modernizing healthcare, offering numerous benefits that enhance efficiency and quality of care [27]. These systems facilitate immediate and remote access to patient data, a critical feature streamlining the medical care decision-making process. By transitioning from paper-based systems, EHRs significantly reduce errors and costs commonly associated with manual record-keeping, enhancing patient safety, affordable care, and care quality [11, 12]. One of the advantages of EHRs is their ability to promote interoperability across different healthcare platforms. This interconnectedness allows for the seamless sharing of patient data among various healthcare providers, leading to improved continuity of care and a more cohesive healthcare experience. They enhance clinical cooperation and increase the accuracy of diagnostics [15, 25].

However, this digital transformation also brings forth complex information security and privacy challenges, which are critical for maintaining patient trust. To address these challenges, the healthcare industry not only adopts strong security technology but is also highly regulated and subject to specific laws, privacy standards, regulations, policies, and best practices that govern healthcare operations and services [17]. Examples of these are the General Data Protection Regulation (*GDPR*) in Europe [30], the Health Insurance Portability and Accountability Act (*HIPAA*), and the Health Information Technology for Economic and Clinical Health Act (*HITECH*) in the USA. These regulations are designed to protect patients, ensure the quality of care, and prevent fraud and abuse. Many of these laws require healthcare organizations to implement technical, administrative, and physical safeguards to secure EHRs [19]. These safeguards include access controls, encryption, authentication measures, and regular security assessments. By enforcing these safeguards, the laws help prevent unauthorized access, data breaches, and identity theft.

Furthermore, some of these laws mandate the implementation of privacy policies and procedures to govern the use and disclosure of patient information. They grant patients certain rights, such as the right to access and amend their medical records. They require healthcare providers to obtain patient consent for specific uses and disclosures of their information. Failure to comply can result in security incidents, healthcare data breaches, fines and penalties, and criminal charges. It can also damage a company's reputation, making attracting and retaining customers and employees difficult.

Unfortunately, even then, unauthorized health data access and disclosure are prevalent in healthcare industries, increasing security and privacy concerns. For example, Table 1 shows the number of compliance complaints received by the U.S. Department of Health and Human Services (HHS) Office for Civil Rights (OCR) [26]. The primary reasons for the complaints are (i) impermissible uses and disclosures of PHI, (ii) lack of safeguards of PHI, (iii) lack of patient access to their PHI, (iv) lack of administrative safeguards of electronic PHI, and (v) use or disclosure of more than the minimum necessary PHI.

Table 1: OCR HHS: Compliance Complaints [26]

Year	Complains	Compliance Reviews	Technical Assistance	Total
2018	25089	438	7243	32770
2019	29853	338	9060	39251
2020	26530	566	5193	32289
2021	26420	573	4244	31237

The following issues must be addressed to avoid or minimize policy violations, protect healthcare data from unauthorized access, and preserve patients' privacy and autonomy over their consent and healthcare resources. (i) Health records access activities or audit logs must be recorded as they have happened in the healthcare systems to recreate the events. (ii) Audit logs must be protected from tampering once recorded. (iii) Compliance checking or audit review should be done correctly and timely to find the compliance status. (iv) A single entity should not perform compliance checking to avoid questions regarding transparency and any influence or bias. (v) Corresponding stakeholders' participation in the compliance checking process increases transparency and acceptability of the audit outcome. (vi) Audit reports must be presented to the corresponding entities promptly and adequately. (vii) Last but not least, healthcare organizations must take effective measures for non-compliance cases to prevent further policy violations.

To address the challenges and requirements mentioned above, this paper proposes a novel consensus mechanism, *Proof of Compliance (PoC)*, for performing audit log compliance verification. Audit logs are stored in a private blockchain network called *Audit Blockchain*. Where a set of independent auditor nodes performs compliance verification through *PoC* blockchain consensus mechanism in a decentralized and distributed manner to determine compliance status as *compliant*, *non-compliant*, and *not-determined*. After determining the compliance status, the audit log ID and compliance status are stored in another private blockchain network called the *Compliance Blockchain* (Figure 1). Private blockchain block ID and hash as integrity are stored on the public blockchain. Involved entities can verify private blockchain data integrity from the public network, as any modifications in the private block change the block integrity.

The assumptions and scope of this paper include the following: (a) required policy selection, evaluation, implementation, and enforcement are done by the healthcare organizations correctly and promptly. (b) Audit logs are captured from the healthcare system accurately, on time, and delivered to the storage unit without any integrity violations. (c) Patients' consents are stored on the public blockchain network, and the required policy lineage is maintained in the policy repository. (d) Patient consent-based policy compliance criteria indicate that accessing health records without consent is a policy violation. (e) Lastly, only logical activities are considered for compliance verification, such as patient electronic health records, physical location access, etc. Based on these assumptions, this paper focuses on maintaining the provenance and compliance checking processes using blockchain and consensus mechanisms.

We examine the architectural design of PoC, illustrating how it integrates with existing blockchain infrastructures and how it can be implemented to enforce compliance without sacrificing the core principles of decentralization, security, and scalability that blockchains offer. Moreover, we address the challenges and opportunities PoC presents in real-world applications, providing insights into how this mechanism can pave the way for broader blockchain adoption across various regulated industries. The PoC extends the blockchain's capability to autonomously verify transactions by incorporating compliance verification as an integral part of the consensus process. Unlike its predecessors, PoC is tailored to ensure that all transactions and the blocks that contain them achieve consensus through traditional means and adhere to a predefined set of compliance rules. These rules can be dynamically adjusted to meet evolving regulatory standards, internal audits, and governance frameworks, making PoC a versatile tool in the blockchain toolkit. Healthcare regulations constantly change, so providers must stay current on the latest requirements.

In the evolving landscape of blockchain technology, where the integrity and security of distributed systems are paramount, consensus mechanisms play an essential role in maintaining network agreement and trust. Traditional consensus models, such as *Proof of Work (PoW)* and *Proof of Stake (PoS)*, have been instrumental in addressing double-spending and Sybil attacks within various blockchain architectures. However, as blockchain applications permeate sensitive and highly regulated sectors, such as healthcare, finance, and supply chain management, there emerges a pressing need for a consensus mechanism that not only ensures transactional integrity and network consensus but also enforces compliance with external regulatory requirements and internal governance policies. This necessity gives rise to the concept of "*Proof of Compliance*," a novel approach designed to bridge the gap between blockchains' autonomous, trustless nature and the stringent compliance demands of modern-day applications.

2 Consensus-Based Policy Compliance Review

The main task of the blockchain consensus mechanism is to agree on a set of transactions or data in a decentralized and distributed ledger. Each node must agree and maintain the same set of transactions for the same block. To do this, a set of tasks must be done. Primary tasks include (i) collecting client transactions and keeping them in the transaction memory pool to select them for the next block. (ii) Verifying signed transactions using the clients' public keys to ensure the claimed or authenticated clients submitted the transactions. (iii) Checking the client account balances to ensure they have enough transaction processing fees and other amounts if the transaction transfers any balance, such as tokens or cryptocurrency. (iv) Ordering the transactions for the block proposal. If submitted transactions are legitimate, they come from the claimed

users and have enough account balances for transaction processing and balance transfer. (v) Proposing the block to other nodes or validators. (vi) Collecting block transaction processing fees and block rewards (if available). (vii) Lastly, taking the blame or being accountable/responsible if anything goes wrong, like invalid transactions in the proposed block.

Many users, nodes, and validators are trying to be selected as block proposers, miners, validators, or forgers to perform the above-mentioned task list. But there is only one vacancy for each block. The most popular and widely used consensus mechanisms are Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Proof of Elapsed Time (PoET), Proof of Authority (PoA), Practical/Istanbul Byzantine Fault Tolerant (P/IBFT), and others. These algorithms adopt various ways to select the block proposer, miner, validator, or forger to perform those seven (7) tasks mentioned above. For instance, the PoS uses the stake value and the age of the validators, whereas the PoW uses the computational capacity to choose the block proposer.

Compliance checking ensures that transactions or operations are executed according to the applicable policies and regulations. Activity data or audit logs must be recorded and protected from modification. The lineages of the applied policies must also be maintained at that time. Audit logs and policies together provide provenance for audit verifications. The entity that performs compliance checking must be distinct from those that carry out operations or keep track of provenance data.

Manual auditing, centralized auditing, or third-party auditing are questionable for their various challenges, such as being time-consuming, costly, prone to human error, vulnerability to attacks, lack of transparency, dependence on external entities, increased costs, etc. [22, 29]. To overcome these issues, a decentralized and distributed process is required to perform compliance reviews against applicable policies. A blockchain consensus mechanism provides these properties to ensure transparency and accountability of PHI access compliance validation.

However, the available consensus mechanisms mentioned earlier do not provide policy compliance-checking functionalities. Compliance checking involves other functionalities besides the functions carried out by the current consensus mechanism. Compliance-checking unique processes (using provenance to verify compliance status) requires a new consensus mechanism besides performing the functionalities of the available consensus mechanisms, as seven points are discussed above. To address this, this paper proposes a compliance mechanism called *Proof of Compliance* to perform compliance status validation in a decentralized and distributed manner. Where a set of independent auditor nodes perform compliance-checking of audit logs using policy lineages without any central or single entity.

3 Proof of Compliance (PoC) Mechanism

The proposed *Proof of Compliance* consensus mechanism for the healthcare industry would provide a way to ensure the compliance status for all the activities or transactions that are granted and executed in the healthcare system. The compliance status can be *(i) compliant*, *(ii) non-complaint*, and *(iii) not-determined* and checked against applicable policies, regulatory requirements, industry standards, and others required by the business natures, contractual obligations, legal jurisdiction, regulatory mandates, and so on. This mechanism would help to increase the overall trust and reliability of the healthcare ecosystem, making it a more valuable tool for patients, providers, business associates, insurance companies, regulatory agencies, and other stakeholders. Figure 1 depicts the proposed approach, whereas Figure 5 shows the Txn structure.

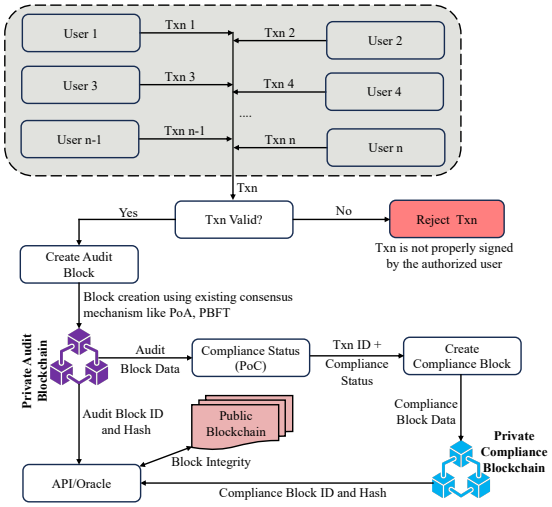

Figure 1: Proof of Compliance Process Overview

3.1 Policy Compliance Criteria and Verification

Policy compliance refers to the adherence to established rules, guidelines, or regulations, collectively known as *policy*, set by an organization, industry, or governing authority to ensure proper behavior, operational integrity, and risk management [4]. It involves meeting the following requirements: (i) Policies must be set according to business requirements and other obligations and communicated among the applicable stakeholders. (ii) Any access or operation request must be validated against the applicable policy before deciding. (iii) Any activity information or audit logs must be preserved so that past events can be reconstructed to make involved entities accountable. Under any conditions or by anyone, the logs must not be tampered with once captured and recorded. (iv) An independent entity, known as *auditor*, separated from the enforcer and audit log maintainer, must review the audit logs against the applied policy.

For healthcare industries, the major data protection laws and regulatory agencies like *GDPR and HIPAA* mandate patient consent for collecting, storing, processing, sharing, and performing other operations. The authors in [1, 2] proposed patient consent-based *PHI* access by the treatment team members and sharing beyond the treatment team for marketing, research, advanced diagnosis, and consultation from another provider. This work focuses on consent-based policy compliance criteria and validation approaches. However, the proposed *PoC* mechanism can be applicable to any policy compliance criteria set by the organizations.

3.1.1 Compliance Checking Components. The major components of the proposed consent-based policy compliance checking approach are *patient consent*, *consent execution timestamps*, *audit logs*, and *audit log timestamps*. They are discussed below in terms of the required conditions.

- Each consent represents a patient's permission to access a specific set of health records under predefined conditions. The finite set of consents is denoted as $C = \{c_1, c_2, c_3, \ldots, c_n\}$, where each element c_i corresponds to an individual consent record.
- Each consent c_i is associated with a timestamp indicating when it was executed. This set of consent timestamps is denoted as $T_C = \{t_{c_1}, t_{c_2}, t_{c_3}, \ldots, t_{c_n}\}$, where t_{c_i} records the time at which consent c_i was executed.
- Each audit log entry captures an access attempt or activity associated with approved requests. The finite set of audit logs is represented as $L = \{l_1, l_2, l_3, \ldots, l_n\}$, where each entry l_i corresponds to a recorded access action.
- Each audit log entry l_i has an associated timestamp that records the exact time of access or activity. This set of audit log timestamps is denoted as $T_L = \{t_{l_1}, t_{l_2}, t_{l_3}, \ldots, t_{l_n}\}$, where t_{l_i} marks the time at which the activity in log l_i occurred.

The conditions $(i) t_{l_i} > t_{c_i}$ and $(ii) t_{l_i} - t_{c_i} \leq \delta$ must be satisfied by both timestamps. They indicate that data access must happen after the corresponding request is evaluated, consent is executed, and a grant decision is made. The business requirements and other obligations determine the value of δ.

3.1.2 Access Token and Audit Log Capture. Only access requests that have been approved are considered in the compliance evaluation. Unsuccessful or denied requests are neither recorded nor evaluated for policy compliance. Granting access doesn't guarantee that the user will successfully access health records, even with approved requests. Figure 2 shows the process of audit log capture using *Access Token* defined in the following. After getting a request from the user, the authorization module evaluates and makes a decision. If the decision is granted, an *Access Token* is created and sent to the healthcare system and audit log recording unit. They both use time information to provide PHI access and record audit logs. An audit log is also created and stored in the log repository if no access is made.

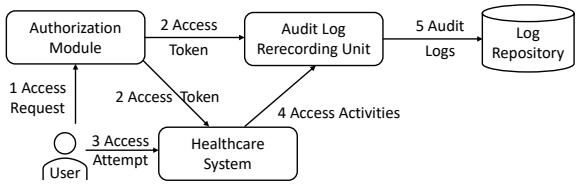

Figure 2: Audit Log Capture

Definition 3.1. **[Access Token $\mathbb{T}$]** $\mathbb{T}$ is defined as a tuple representing authorized access, composed of three components:

$$\mathbb{T} = (\mathbb{R}_i, t_{start}, t_{end})$$

Where (i) $\mathbb{R}_i$ denotes the unique *Request ID* associated with the user's access request, (ii) t_{start} represents the *Access Start Time* when the access is first permitted; users cannot access health records before this time, and (iii) t_{end} represents the *Access End Time* when the access expires. After this time, users cannot access healthcare data.

Access to data is allowed for the user only if the access attempt is made within the specified time interval:

$$t_{start} \leq t_{attempt} \leq t_{end}$$

Where $t_{attempt}$ is the timestamp of the access attempt. Any access attempt outside this interval is denied. For the access attempt of request $\mathbb{R}_i$, $T_{L_i} = t_{attempt}$ must be satisfied.

3.1.3 Compliance Status Verification.

Definition 3.2. **[Compliance Criteria ζ]** The compliance function ζ defines a mapping from each audit log entry to its corresponding executed consent, verifying that an authorized consent backs each recorded access. Formally, $\zeta : L \longrightarrow C$, where L represents the set of all recorded audit logs, and C represents the set of all executed consents.

$$\zeta = |\{l_i \mapsto c_i \mid 1 \leq i \leq n\}|$$

Algorithm 1: Proof of Compliance (PoC) Consensus Mechanism

Input : (i) list of audit logs ($Txns$) and (ii) set of policy $Plcy$
Output: (i) compliance status of the audit logs ($Txns$)

Initialization
$\mathbb{N}_{Order}$ order nodes
$\mathbb{N}_{Validator}$ validator/endorser nodes
$\mathbb{N}_{Audit}$ audit nodes
$\mathbb{N}_{Committer}$ committer nodes
Audit Logs Integrity Verification and Order
$Txn_{Valid} = [\,]$ /* accepted transaction list */
$Txn_{Invalid} = [\,]$ /* rejected transaction list */
for $i \leftarrow Txns_{Start}$ **to** $Txns_{End}$ **by** 1 **do**
  if $\zeta(PK_i, Tnx_i) == Signed_{Tnx_i}$ **then**
    $Txn_{Valid} \leftarrow Txn_{Valid} + Txn_i$
  else
    $Txn_{Invalid} \leftarrow Txn_{Invalid} + Txn_i$
  end if
end for
Policy Compliance Verification
$Txn_{Compliance} = [\,]$ /* compliance transactions */
$Txn_{NonCompliance} = [\,]$ /* noncompliance transactions */
for $i \leftarrow Txn_{Accepted_{Start}}$ **to** $Txn_{Accepted_{End}}$ **by** 1 **do**
  if $\zeta(PK_i, Tnx_i) == Signed_{Tnx_i}$ **then**
    $Txn_{Compliance} \leftarrow Txn_{Compliance} + Txn_{Accepted_i}$
  else
    $Txn_{NonCompliance} \leftarrow Txn_{NonCompliance} + Txn_{Accepted_i}$
  end if
end for
Ledger Modification
$Txn_{Compliance} = [\,]$ /* compliance checked final transactions */
$Txn_{NonCompliance} = [\,]$ /* noncompliance transactions */
for $i \leftarrow Txn_{Accepted_{Start}}$ **to** $Txn_{Accepted_{End}}$ **by** 1 **do**
  if $\zeta(PK_i, Tnx_i) == Signed_{Tnx_i}$ **then**
    $Txn_{Compliance} \leftarrow Txn_{Compliance} + Txn_{Accepted_i}$
  else
    $Txn_{NonCompliance} \leftarrow Txn_{NonCompliance} + Txn_{Accepted_i}$
  end if
end for

3.2 Participant Nodes and Transaction Flow

Multiple nodes participate in the *Proof of Compliance* mechanism to perform various functions to complete the compliance verification process for the submitted audit logs. The following discusses

the *Orderer, Validator, Auditor,* and *Committer* nodes along with their corresponding activities, message communication, and transaction flow in the proposed approach. In addition to these nodes, the *patient* gives consent, and those are deployed to the public blockchain as detailed in [1, 2]. The client nodes perform activities in the systems captured as audit logs and stored in the private audit blockchain. Algorithm 1 shows the steps of the PoC process.

(i) Orderer Node: It performs all transactions and consents ordering services. Audit logs from the audit blockchain are processed as blocks. This node gets a block from the audit blockchain and related consent from the public blockchain. Then, it transfers them to the *Validator* node for verification.

(ii) Validator Node: It verifies the audit block integrity from the public blockchain as block ID and hash as integrity stored previously. If there is no modification, the audit logs and required consents are transmitted to the auditor nodes for performing compliance review.

(iii) Auditor Nodes: These nodes are responsible for checking the compliance requirements for regulations and other applicable bodies. Auditor nodes can be hospitals, local governments, state governments, the federal government, regulatory agencies, insurance companies, business associates, accreditation bodies, independent auditors, and others from contractual obligations. Each node works as an honest entity where the provenance data, audit trails, and applicable policies are analyzed to determine the activities' compliance status. They don't store data for further analysis, share, or transfer with other users. The compliance status can be one of *Compliant, Non-Compliant*, or *Not-Determined*.

(iv) Committer Node: After performing the compliance review of the submitted block, this node writes the compliance data as a compliance block in the compliance blockchain network. After writing, it stores the compliance block ID and hash in the public blockchain for later verification. After writing the transactions to the ledger, no entity can modify the blocks or transactions.

All participant nodes must communicate with each other. The communication may be *(i) one-to-one, (ii) one-to-many, (iii) many-to-one*, and *(iv) many-to-many* according to the network requirements. They can be any of the following based on the nodes' functionalities and network requirements: *(i) unidirectional* and *(ii) bidirectional*. The message communications are done in **atomic broadcast** or **total order broadcast** fashion [6]. Where all participant nodes receive the same set of required messages in the same order, that is the same sequence of messages. The *atomic broadcast* ensures that messages are either eventually delivered correctly for all participants or all participants abort messages without side effects. However, this paper does not provide a detailed functional mechanism for message communication.

Figure 3 shows the sequence diagram for transaction flows. The network has various participating nodes, as described above. Each node performs different operations. In the following section, transaction flow and multiple operations are described.

(a) Consent and Transaction Submission: Patient consent is captured and stored in the public blockchain for authorizing health records access requests. The consent is also required for the compliance review. Client nodes submit transactions to be validated and compliance checked to be added to the ledger. The client nodes use their private keys to sign all transactions digitally.

(b) Audit Logs Integrity Verification and Order: After receiving transactions from Client nodes, Orderer nodes perform signature verification to ensure that submitted transactions come from legitimate clients who claim to be the originators of the submitted transactions. Signature verification is done through a private key pair. Clients sign transactions using their private keys. Order nodes verify signed transactions using clients' public keys. Once signatures are verified, all transactions are ordered and submitted for verification and policy compliance checking.

(c) Compliance Verification: In this stage, if a transaction complies with all the applicable policies, then auditor nodes mark the transaction as compliance. Otherwise, the transaction is marked as non-compliance or not-determined. This process repeats for all transactions validated by the validator node.

(d) Transaction Committed and Ledger Modification: Once transactions are verified, executed, and compliance checked, they are committed as finalized and can't be modified after this point. Finally, the compliance block is written to the compliance blockhain. After writing to the ledger, the compliance block ID and hash as integrity are written to the public blockchain for later verification.

3.3 PoC Decision Combining Algorithm

The algorithm aggregates these individual outcomes, applying rules to resolve conflicts and derive a unified compliance status, ensuring consistent and trustworthy decision-making. Figure 4 shows the decision-making process.

3.3.1 Decision Counting Threshold. Suppose a total s number of auditor nodes exists in the *PoC* network. A batch of transactions is sent with the required information to evaluate compliance status. It is not always the case that we will receive a s number of responses. There might be some cases where the auditors' responses may be lost due to connectivity issues, power failure, intentional result not submission, auditor node offline, or after starting the process, it goes offline due to the system error, among others [10]. Now, consider that m is the number of responses from the auditors out of s. We need to set a threshold, η, that must be satisfied to make the compliance decision for an audit log. The following conditions must be satisfied to make the compliance decision:

$$(i) s \geq m \quad \text{and} \quad (ii) s \geq m \geq \eta \quad \textbf{or} \quad s \geq D_m \geq \eta$$

Where D_m is the number of received decisions from the m number of auditors (A), and η is the minimum number of decisions that must be present to make the decision. If there is no loss, this $s = m$ is ideal. Then the conditions became:

$$(i) m \geq \eta \quad \textbf{or} \quad D_m \geq \eta$$

In the ideal case, all auditors receive the required information and return results after the compliance evaluation. The value of the η is determined and influenced by the design decision, the organization's business nature, legal requirements, contractual obligations, and others. If $m < \eta$ or $D_m < \eta$, the compliance status is assigned as *"Not-Determined"* to avoid any policy violation, it must be further investigated to check the reasons.

3.3.2 Auditor Obligations. Let A_R be a set of r numbers of obligatory auditors, defined as $A_R = \{a_{r_1}, a_{r_2}, a_{r_3}, \ldots, a_{r_r}\}$ where each a_{r_i}

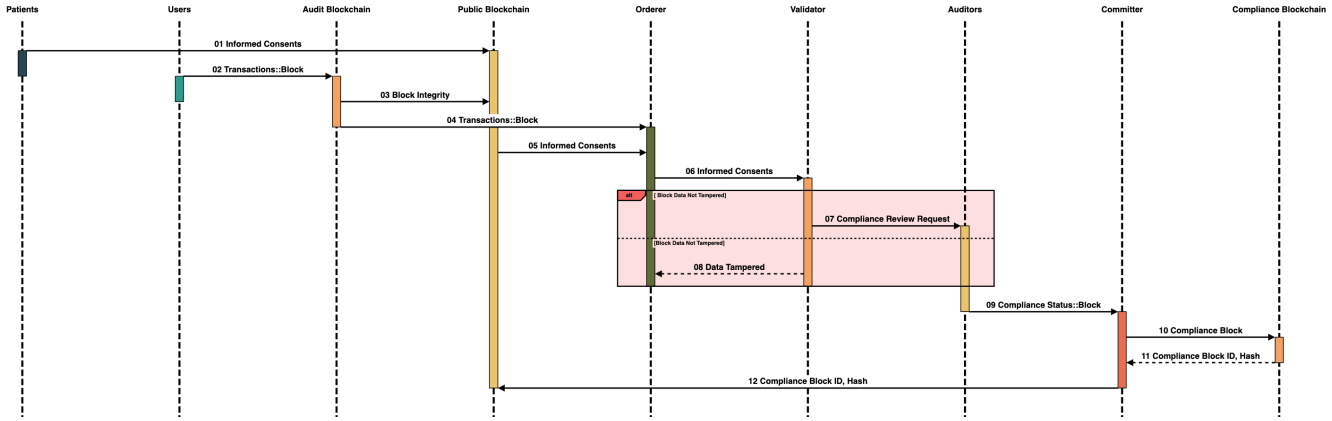

Figure 3: Proof of Compliance (PoC) Transaction Flow

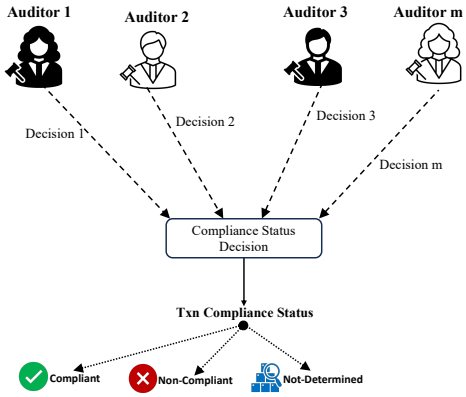

Figure 4: Proof of Compliance Decision Mechanism

represents an individual obligatory auditor node whose participant in the *PoC* process is mandatory. The number of obligatory auditor participant, φ, is counted as follows:

$$\varphi = \sum_{i=1}^{r} \partial(a_{r_i} \in A)$$

Where $\partial(.)$ is an indicator function that equals *1* if a_{r_i} is a participant of the auditor set A and *0* otherwise. The condition: $\varphi = r$ must be satisfied. If no condition is imposed regarding mandatory participation of the auditors, A_R, then the requirements are waived.

3.3.3 Auditors and Decisions. Let m be the total number of auditor nodes; the following information is given. The final compliance decision is derived based on a majority rule among decisions.

- Let A be a set of auditors, defined as $A = \{a_1, a_2, a_3, \ldots, a_m\}$, where each a_i represents an individual auditor node.
- Let D be a set of decisions corresponding to each auditor in A, defined as $D = \{d_1, d_2, d_3, \ldots, d_m\}$, where d_i is the decision made by the a_i auditor node for a given transaction, where $d_i \in \{Compliant, Non-Compliant, Not-Determined\}$
- Each auditor node a_i in set A makes a compliance decision d_i in set D. Therefore, here is a one-to-one mapping between each auditor node and its decision: $(a_1, d_1), \ldots, (a_m, d_m)$. This mapping allows us to analyze the decisions collectively and apply the *PoC decision combining algorithm.*
- Let a set of weights defined as $W = \{w_1, w_2, \ldots, w_m\}$, where w_i represents the weight of an individual auditor node a_i.

Purpose of Weighting: Weighting allows for differentiated influence among auditors. For instance, an auditor from a regulatory agency might have a higher weight due to their authority, while an internal auditor may have a standard weight. This flexibility is beneficial in settings where some nodes have greater compliance oversight responsibilities. This weighted decision-making model provides a robust framework for ensuring fair and accurate compliance outcomes in a decentralized, consensus-driven environment. The weight of the auditors would be determined and influenced by the business requirements, legal jurisdictions, regulatory mandates, contractual obligations, and others.

Weight Threshold: It ensures that a compliance decision is made only when the cumulative influence of participating auditor nodes reaches a predefined minimum level. Let Ω represent the weight threshold and W_{total} the total weight of all auditor nodes, calculated as:

$$W_{total} = \sum_{i=1}^{m} w_i$$

where w_i is the weight of the i-th auditor node and m is the total number of auditors. The decision-making process proceeds only if $W_{total} \geq \Omega$. If this condition is satisfied, the compliance mechanism calculates the weighted counts for each decision type (*Compliant, Non-Compliant, Not-Determined*) and determines the final compliance status based on defined majority rules.

If $W_{total} < \Omega$, the system delays the decision, requesting additional input to meet the threshold. This ensures that decisions are not based on insufficient or low-weight contributions, thereby enhancing the reliability and fairness of the PoC mechanism. Alternatively, the compliance status can be determined as *"Not-Determined"* to avoid policy violations.

3.3.4 Decision Counting and Combining Process with Weight. In this scope, all the auditor nodes don't bear the same weight values, where $w_1 \neq w_2 \neq w_3 \neq \cdots \neq w_m$, indicating that they don't have

Table 2: PoC Decision Combining Scope with Weight

SN	Decision Counting Combination-Weight	Final Decision-Weight($\mathbb{D}_{\mathbb{W}final}$)
1	$\mathbb{C}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$	$\mathbb{C}_{\mathbb{W}}$
2	$\mathbb{C}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$	$\mathbb{C}_{\mathbb{W}}$
3	$\mathbb{C}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}} = \mathbb{U}_{\mathbb{W}}$	$\mathbb{C}_{\mathbb{W}}$
4	$\mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$	$\mathbb{N}_{\mathbb{W}}$
5	$\mathbb{N}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$	$\mathbb{N}_{\mathbb{W}}$
6	$\mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}} = \mathbb{U}_{\mathbb{W}}$	$\mathbb{N}_{\mathbb{W}}$
7	$\mathbb{N}_{\mathbb{W}} = \mathbb{C}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$	$\mathbb{N}_{\mathbb{W}}$
8	$\mathbb{U}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$
9	$\mathbb{U}_{\mathbb{W}} > \mathbb{N}_{W} > \mathbb{C}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$
10	$\mathbb{U}_{\mathbb{W}} = \mathbb{C}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$
11	$\mathbb{U}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$
12	$\mathbb{U}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$
13	$\mathbb{C}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}} = \mathbb{U}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$

an equal impact on the decision. The weight value depends on the nature of the auditor and business requirements.

Decision Counts with Weight The total counts for each type of decision with weight are calculated as follows, where $\delta(.)$ is an indicator function that equals *1* if the inside condition is true and *0* otherwise.

$$(i) \quad \mathbb{C}_{\mathbb{W}} = \sum_{i=1}^{m} w_{a_i}.\delta(D_i = Complaint)$$

$$(ii) \quad \mathbb{N}_{\mathbb{W}} = \sum_{i=1}^{m} w_{a_i}.\delta(D_i = Non - Complaint)$$

$$(iii) \quad \mathbb{U}_{\mathbb{W}} = \sum_{i=1}^{m} w_{a_i}.\delta(D_i = Not - Determined)$$

Decision Combining Process with Weight After counting each decision type, the final decision is made based on the majority. The *Weighted Not-Determined* dictates to others if they are equal to it. The distinct combinations are given in Table 2. The final decision $\mathbb{D}_{\mathbb{W}final}$ is set based on predefined majority rules, such as:

- *Weighted Compliant Majority:* This decision is made when the majority decision is *Weighted Complaint* or $\mathbb{C}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$ and $\mathbb{C}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$ out of *m* decisions made by the auditors regardless $\mathbb{N}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$ or $\mathbb{U}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$ or $\mathbb{U}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}}$.
- *Weighted Non-Compliant Majority:* This decision is made when the majority decision is *Weighted Non-Complaint* or $\mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$ and $\mathbb{N}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$ out of *m* decisions made by the auditors regardless $\mathbb{C}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$ or $\mathbb{U}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$ or $\mathbb{C}_{\mathbb{W}} = \mathbb{U}_{\mathbb{W}}$.
- *Weighted Not-Determined Majority:* This decision is made when the majority decision is *Weighted Not-Determined* or (i) $\mathbb{U}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$ and $\mathbb{U}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$, or (ii) $\mathbb{U}_{\mathbb{W}} = \mathbb{C}_{\mathbb{W}} = \mathbb{U}_{\mathbb{W}}$, or (iii) $\mathbb{U}_{\mathbb{W}} = \mathbb{C}_{\mathbb{W}} > \mathbb{U}_{\mathbb{W}}$, or (iv) $\mathbb{U}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$ out of *m* decisions made by the auditors regardless $\mathbb{C}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}}$ or $\mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$ or $\mathbb{C}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}}$.

3.3.5 Decision Counting and Combining Process without Weight. In this scope, all the auditor nodes bear the same weight values, where $w_1 = w_2 = w_3 = \cdots = w_m$, indicating that they have an equal impact on the decision.

Decision Counts without Weight The total counts for each type of decision are calculated as follows, where $\delta(.)$ is an indicator function that equals *1* if the inside condition is true and *0* otherwise.

Table 3: PoC Decision Combining Scope without Weight

SN	Decision Counting Combination	Final Decision ($\mathbb{D}_{final}$)
1	$\mathbb{C} > \mathbb{N} > \mathbb{U}$	$\mathbb{C}$
2	$\mathbb{C} > \mathbb{U} > \mathbb{N}$	$\mathbb{C}$
3	$\mathbb{C} > \mathbb{N} = \mathbb{U}$	$\mathbb{C}$
4	$\mathbb{N} > \mathbb{C} > \mathbb{U}$	$\mathbb{N}$
5	$\mathbb{N} > \mathbb{U} > \mathbb{C}$	$\mathbb{N}$
6	$\mathbb{N} > \mathbb{C} = \mathbb{U}$	$\mathbb{N}$
7	$\mathbb{N} = \mathbb{C} > \mathbb{U}$	$\mathbb{N}$
8	$\mathbb{U} > \mathbb{C} > \mathbb{N}$	$\mathbb{U}$
9	$\mathbb{U} > \mathbb{N} > \mathbb{C}$	$\mathbb{U}$
10	$\mathbb{U} = \mathbb{C} > \mathbb{N}$	$\mathbb{U}$
11	$\mathbb{U} > \mathbb{C} = \mathbb{N}$	$\mathbb{U}$
12	$\mathbb{U} = \mathbb{N} > \mathbb{C}$	$\mathbb{U}$
13	$\mathbb{C} = \mathbb{N} = \mathbb{U}$	$\mathbb{U}$

$$(a) \quad \mathbb{C} = \sum_{i=1}^{m} \delta(D_i = Complaint)$$

$$(b) \quad \mathbb{N} = \sum_{i=1}^{m} \delta(D_i = Non - Complaint)$$

$$(c) \quad \mathbb{U} = \sum_{i=1}^{m} \delta(D_i = Not - Determined)$$

Decision Combining Process without Weight After counting, the final decision is made, and the distinct combinations are given in Table 3. The *Not-Determined* dictates to others if they are equal to it. The final decision $\mathbb{D}_{final}$ can then be set based on predefined majority rules, such as:

- *Compliant Majority:* This decision is made when the majority decision is *Complaint* or $\mathbb{C} > \mathbb{N}$ and $\mathbb{C} > \mathbb{U}$ out of *m* decisions made by the auditors regardless $\mathbb{N} > \mathbb{U}$ or $\mathbb{U} > \mathbb{N}$ or $\mathbb{U} = \mathbb{N}$.
- *Non-Compliant Majority:* This decision is made when the majority decision is *Non-Complaint* or $\mathbb{N} > \mathbb{C}$ and $\mathbb{N} > \mathbb{U}$ out of *m* decisions made by the auditors regardless $\mathbb{C} > \mathbb{U}$ or $\mathbb{U} > \mathbb{C}$ or $\mathbb{C} = \mathbb{U}$.
- *Not-Determined Majority:* This decision is made when the majority decision is *Not-Determined* or (i) $\mathbb{U} > \mathbb{C}$ and $\mathbb{U} > \mathbb{N}$, or (ii) $\mathbb{U} = \mathbb{C} = \mathbb{U}$, or (iii) $\mathbb{U} = \mathbb{C} > \mathbb{U}$, or (iv) $\mathbb{U} = \mathbb{N} > \mathbb{C}$ out of *m* decisions made by the auditors regardless $\mathbb{C} > \mathbb{N}$ or $\mathbb{N} > \mathbb{C}$ or $\mathbb{C} = \mathbb{N}$.

4 Transaction and Block Structure

The proposed *Proof of Compliance* or *PoC* mechanism takes audit logs from the audit blockchain, required informed consent from the public blockchain network, and applicable policies from the policy repository. After performing the compliance verification, the compliance status for each audit log is generated and stored in the compliance blockchain. The compliance blockchain is another private blockchain that contains audit log IDs and corresponding compliance statuses. The following describes the *(i) audit log transaction structure, (ii) audit log block structure, (iii) compliance transaction structure*, and *(iv) compliance block structure.*

4.1 Audit Block Transaction Structure

An audit log indicates a single operation has already occurred in the system. This study considers two types of audit logs, as depicted in Figure 5. Figure 5(a) shows the audit log for treatment team access. Next, Figure 5(b) shows the log structure for PHI sharing. The major components are discussed below.

- *Audit Log ID:* It is an ID to identify the audit log uniquely in the *Audit Blockchain* as well as in the *Compliance Blockchain*.
- *Timestamp Data:* A timestamp is the block creation time. The time has been given in seconds since 1.1.11970. For compliance checking, this time value is crucial.
- *Treatment-Informed Consent ID:* The HIPAA privacy law mandates patients' consent for accessing their health records [18, 21]. This work stores patient-informed consent for treatment in the public blockchain network. The detailed process can be investigated in [1]. There are four components in every given informed consent: *(i) user, (ii) PHI, (iii) operation,* and *(iv) conditions*. The complete consent can be retrieved from the public blockchain using the consent ID included in the audit log.
- *PHI:* It is an electronic version of a patient's medical data that providers keep over time. They are protected health information and sensitive patient information. PHI must be protected from unauthorized access, disclosure, and sharing. Table 4 shows the sample health records, categorized by ID, name, and description.
- *User ID:* This unique user ID performs various operations. It is also called the subject, which may be one of the treatment team members or anyone from the hospital. For this study, we don't consider external users to be treatment team members.
- *Operation:* It represents the system action authorized users can perform on the objects or PHI when certain conditions are satisfied. Examples of operations are *read*, *write*, and *update*. Not all members have access to all forms of PHI to perform their job responsibilities. In addition to the treatment team, the patient has the right to read, write, and update specific health records.
- *Sharing Informed Consent ID:* Sharing informed consent means the patient's consent to share medical data for a specific purpose. The sharing informed consent is stored in the public blockchain network, which has four components: *(i) sender, (ii) receiver, (iii) PHI,* and *(iv) purpose* [2]. All components are retrieved from the public blockchain network using this consent ID included in the audit log. Both the sender and the receiver must have consent. The sender can share specific healthcare data with the receiver, who has permission from the patient. The *sender* may be a patient treatment team member or anyone from the provider. The *receiver* may be from other hospitals, labs, medical research institutes, pharmaceutical companies, marketing departments, government officials, etc. The *purposes* may be treatment, diagnosis, marketing, research, etc.
- *Honest Broker ID & Report:* An honest broker is a trusted entity that evaluates the encryption algorithm, key size, and data anonymity status [3]. After checking, the honest broker certifies or attests to the status, which is recorded in audit trails as proof.

4.2 Audit Blockchain Block Structure

The audit log block contains a certain number of audit logs generated by the clients and captured by the log daemon or authorization module. It includes some block metadata in addition to the log data, as depicted in Figure 6. The network participants can determine the number of log records. If the number of records is fixed, the block size will always be the same. Otherwise, the block size would vary. This paper stores a certain number of log records for each block to keep all block sizes the same.

Table 4: Sample Protected Health Information Structure

PHI ID	PHI Name	PHI Description
PHI1001	Demographic Info	Patient's information
PHI1002	Previous Medical History	Old medical records from another hospital
PHI1003	Immunizations	Immunization records that are administered over time
PHI1004	Allergies	Various allergies sources, triggering condition, remediation
PHI1005	Visit Notes	Physiological data, disease description, advice, follow-up
PHI1006	Medications & Prescription	Prescribed medications including name, dosage, etc.
PHI1007	Pathology Lab Works	Blood work
PHI1008	Radiology Lab Works	Imaging and Radiology Lab results
PHI1009	Billing and Insurance	Bank account and insurance policy Information
PHI1010	Payer Transactions	Bills of doctor visit, lab works, and medications

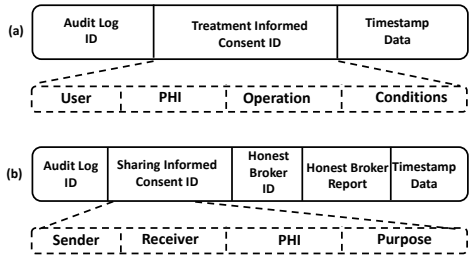

Figure 5: Audit Log Transaction Structure

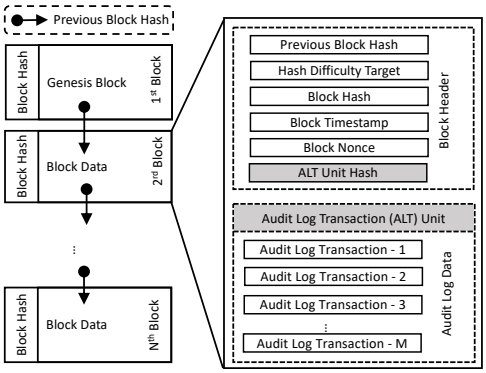

Figure 6: Audit Blockchain Block Structure

4.3 Compliance Block Transaction Structure

Compliance Status: the auditor nodes determine the compliance status based on the consensus agreement through a decision-combining algorithm. The status can be *complainant, non-compliant, or not-determined.*

- *Compliant:* It indicates that the authenticated subject operates by the relevant or applicable policies. We consider consent-based protected health information accessing or sharing. So, users access or share PHI when they have consent from the patients. Otherwise, they will not be able to access or share PHI.
- *Non-Compliant:* In this scope, the applicable policies are violated, and the authenticated subject is neither supported nor carried out in operation. This violation is subject to various corrective

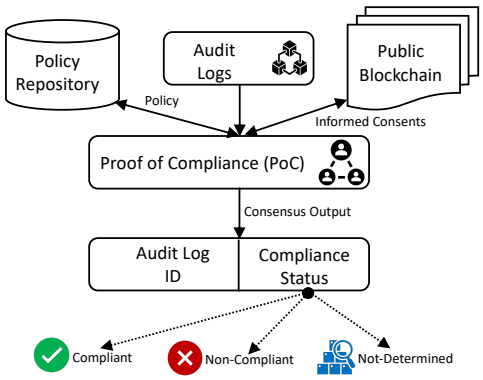

Figure 7: Compliance Block Transaction Structure

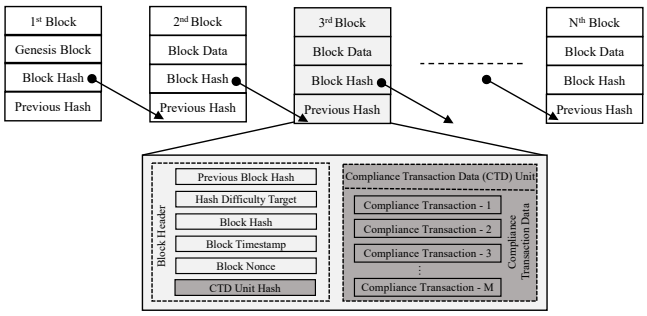

Figure 8: Compliance Blockchain Block Structure

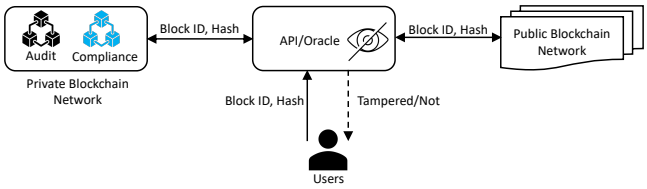

Figure 9: Storing Audit Block Integrity on Public Blockchain

actions, like employee warning, training, transferring, terminating, etc. Also, organizations need to deploy new security systems or update existing ones to help minimize violations.

- *Not-Determined:* In this situation, it is not possible to determine the status of any audit log or executed operation. This may be due to the unavailability of the required information, such as policies, informed consent, etc. Also, some auditor nodes could not determine compliance status. These cases must be investigated later to resolve the issues.

4.4 Compliance Blockchain Block Structure

The compliance block is described as an organized structure designed to record the compliance status of audit logs securely. Figure 8 shows the compliance block structure. Each compliance block includes unique audit log IDs and corresponding compliance statuses, categorized as compliant, non-compliant, or not-determined. These blocks are then stored within the private compliance blockchain, providing an immutable record of all verified compliance checks.

4.5 Private Block Integrity on Public Blockchain

A private blockchain is a system configured for a select group of participants who establish the consensus mechanism and specific features such as block structure, block size, and block contents [5]. To safeguard against the intentional alteration of audit and compliance blocks, the proposed approach stores the block ID and hash as block integrity on a public blockchain, such as *Ethereum*, to maintain block integrity. Figure 9 shows the process of storing block ID and hash of private blocks on the public blockchain network. This dual-layer approach ensures that the private blockchain retains integrity over its operations while leveraging the security and immutability of the public blockchain [23]. An *API/Oracle* is developed and deployed to store and retrieve block integrity information to/from the blockchain network. Here, the *API/Oracle* is a secured, trusted, and blind entity that doesn't reveal any information without authorized users. The authorized users submit requests to know the integrity status of any private block. The *API/Oracle* checks and returns a response as to whether the requested block is modified.

5 Experimental Evaluations

This section provides experimental evaluations to demonstrate the functionality of the proposed consensus mechanism and assess the compliance status of logical health record access. The evaluations focus on the following aspects: (i) Setting up private Ethereum networks for the audit and compliance blockchain. (ii) Measuring the gas cost for writing block integrity data to public blockchain networks. (iii) Analyzing the time required for writing and reading block integrity data to/from public blockchain networks. (iv) Evaluating the time needed for compliance block construction. (v) Assessing the throughput of compliance checks. Each aspect is discussed below with the necessary data, figures, and tables.

5.1 Environment Setup

To implement the blockchain model, we used *Node.js* to develop server-side programs for different roles in our network: client, orderer, auditor, and committer. Each node operates as an independent JavaScript program, performing essential functions for blockchain operations. The client node features a *Command Line Interface (CLI)* to generate synthetic data, mimicking healthcare transaction activities and audit logs. This data tests the blockchain system's performance and reliability. Nodes were encapsulated in containers in *Docker* to ensure isolated, consistent environments, simplifying dependency management and network communications [24]. We used *Go Ethereum Docker* image *version 1.13.15* for our private network setup. This setup imitates a distributed ledger effectively. Testing was conducted on an *Apple Mac M1 Air* running *macOS Sonoma version 14.3*, with *256 GB storage* and *8 GB of unified memory*.

5.2 Block Integrity Writing Cost

In the proposed approach, audit logs are stored in the audit blockchain, and compliance status is stored in the compliance blockchain. Both are private blockchain networks, where participants are limited to organizations. This doesn't provide the public with trust. To avoid this, block ID and hash as integrity are stored in a public network like *Ethereum*. Figure 10 shows the block integrity storage cost in tokens for three public blockchain networks: *Ethereum, Binance*

Smart Chain, and Optimism. The USD costs are depicted in Figure 11. The first two networks are *Layer 1*, and the third network is *Layer 2* [7, 9]. *Layer 1* is the core blockchain framework for implementing the network's consensus mechanism, transaction validation and storage, and native token functionality. *Layer 2* is a secondary framework built on top of an existing *Layer 1* blockchain to enhance the scalability and efficiency of the *Layer 1* blockchain without compromising its security or decentralization. It performs transaction validation and storage outside the *Layer 1* network but stores proof on it. The *Layer 2* solution handles more transactions per second, reducing transaction costs and speeding up confirmation times.

5.3 Time Requirements

Before processing any data from the private network, the block integrity is verified from the public network. Doing this requires access to a public blockchain network to read stored private blockchain block IDs and hashes. This study leverages *Ethereum's Remote Procedure Call (RPC) API* services for deploying smart contracts and performing transactions on these networks [14]. Writing and reading time requirements are measured for three networks: *Ethereum, Binance Smart Chain, and Optimism.* Table 5 shows ten (10) writing time requirements. Table 6 shows ten (10) reading time requirements. Additional time requirements are introduced since transactions are traveled through the API servers. Maintaining a local public blockchain node where access to block data is possible in real-time can reduce reading time. The system continuously synchronizes with the blockchain network to update the ledger data. The providers can maintain local nodes for faster integrity verification. However, time differences are not considered in this study.

Table 5: Writing Time to Public Blockchain Networks

Writing SN.	Ethereum	Binance Smart Chain	Optimism
1	6.719 Sec	6.854 Sec	**8.459 Sec**
2	**5.961 Sec**	**6.068 Sec**	7.785 Sec
3	5.972 Sec	6.338 Sec	7.738 Sec
4	6.309 Sec	6.063 Sec	7.762 Sec
5	6.085 Sec	6.081 Sec	8.163 Sec
6	6.015 Sec	2.476 Sec	**7.482 Sec**
7	10.117 Sec	6.521 Sec	7.718 Sec
8	10.041 Sec	**2.451 Sec**	8.268 Sec
9	10.045 Sec	6.662 Sec	7.736 Sec
10	**14.039 Sec**	2.458 Sec	7.797 Sec
Average Time	*8.130 Sec*	*5.197 Sec*	*7.891 Sec*

Table 6: Reading Time from Public Blockchain Networks

Reading SN.	Ethereum	Binance Smart Chain	Optimism
1	0.834 Sec	0.541 Sec	0.631 Sec
2	0.573 Sec	0.468 Sec	0.453 Sec
3	0.570 Sec	**0.620 Sec**	0.404 Sec
4	**0.391 Sec**	0.501 Sec	**0.650 Sec**
5	0.514 Sec	0.488 Sec	0.406 Sec
6	0.583 Sec	0.566 Sec	0.495 Sec
7	**1.577 Sec**	0.579 Sec	0.421 Sec
8	0.463 Sec	**0.442 Sec**	0.438 Sec
9	0.580 Sec	0.504 Sec	0.415 Sec
10	0.483 Sec	0.495 Sec	**0.398 Sec**
Average Time	*0.660 Sec*	*0.532 Sec*	*0.470 Sec*

5.4 Compliance Block Construction Time

After performing compliance checking and block finalization, it is the required time to confirm a compliance block. The *Auditor* nodes are responsible for compliance checking and making final compliance status decisions. The *Committer* nodes perform the block finalization by writing the compliance block to the compliance blockchain ledger. It does not include the time the *Orderer* nodes require to fetch audit logs from the private audit blockchain, get informed consent from the public blockchain network, and collect applicable policies from the policy repository. Figure 12(a) shows the compliance block construction time, where the maximum time is 4.317, the minimum is 4.134, and the average is 4.19 seconds.

5.5 Compliance Checking Throughput

It is the number of transactions per second that can be processed after performing all required operations. For the *Proof of Compliance* consensus mechanism, the performed operations are compliance checking and compliance block finalization by the *Auditor* and *Committer* nodes. Figure 12(b) depicts the throughput in transactions per second (TPS), where the maximum throughput is 48.38, the minimum is 46.33, and the average is 47.70 TPS.

6 Related Works

García-Berná et al. present a novel workflow designed to enhance the usability audits of personal health records (PHRs) through an automated, computer-aided usability evaluation (CAUE) tool named Usevalia [8]. This approach integrates multiple components, including a set of usability heuristics, a catalog of usability requirements, a corresponding checklist, and predefined tasks to understand the functionalities of PHRs to be audited. The workflow leverages Usevalia to centralize and streamline the usability evaluation process, allowing for coordinated work among auditors and providing remote access to all necessary evaluation materials.

Stevovic et al. on compliance-aware cross-organization medical record sharing tackle the complex challenge of sharing electronic health records (EHRs) across various healthcare organizations while adhering to differing regulatory and business requirements [28]. Their proposed solution, CHINO, allows healthcare providers to define and enforce their specific security and compliance needs during data sharing. The critical point is the integration of business processes that map high-level regulatory policies to particular data management operations, ensuring each organization's internal systems and processes remain compliant. The implemented prototype was successfully integrated with OpenMRS, illustrating the system's ability to manage and enforce diverse regulatory and business requirements across healthcare settings.

Dae-young et al. developed a sophisticated framework to securely manage the exchange of extensive health data while ensuring strict compliance with health regulations such as HIPAA [13]. Amidst the challenges of the COVID-19 pandemic, their framework utilizes semantic web technologies to ensure secure and compliant data exchanges through dynamically applied policies. A key feature of their approach is the Trust Score, which assesses each participant's reliability in handling sensitive data. The authors demonstrated the framework's effectiveness and scalability by applying it

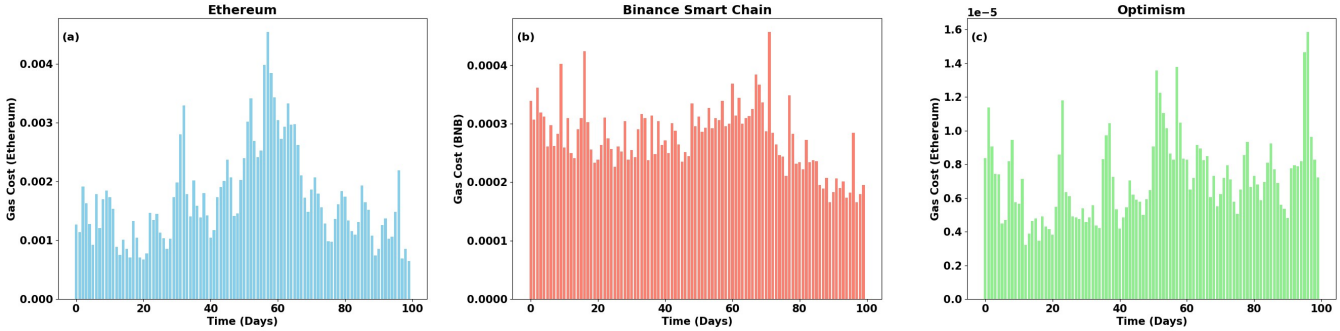

Figure 10: Private Block ID and Integrity Storage Token Cost—Public Blockchain Networks

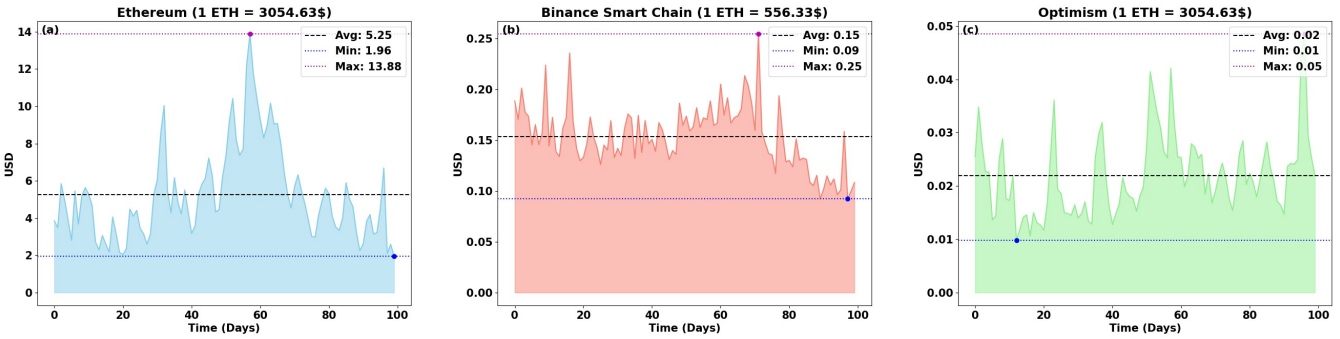

Figure 11: Private Block ID and Integrity Storage USD Cost—Public Blockchain Networks

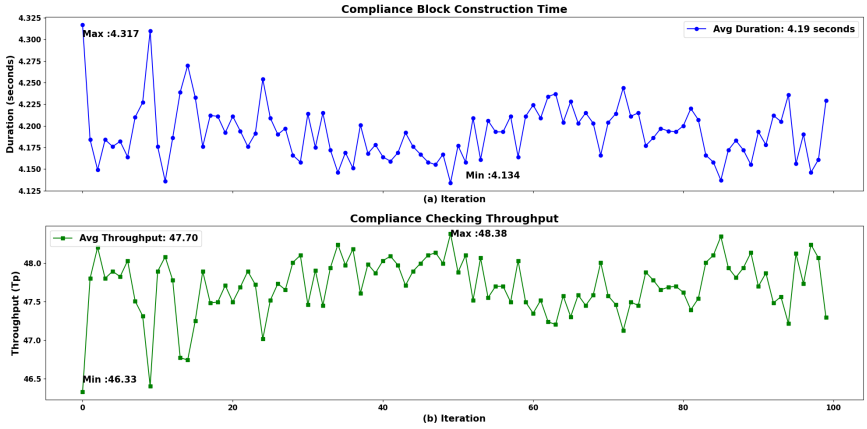

Figure 12: (a) Compliance Block Construction Time (b) Compliance Checking Throughput

to a simulated scenario involving over one million synthetic contact tracing records from the CDC.

Koreff et al. [16] critically examined data analytics in healthcare fraud audits, focusing on how these tools influence power dynamics and potentially abuse authority. Through a qualitative analysis involving interviews and document reviews, their study revealed that algorithmic decision-making can justify harsh measures against healthcare providers based on possibly inaccurate data interpretations. This misuse of power affects individual providers and has broader implications for the industry's power structure. The research highlighted the need for greater transparency and accountability in using data analytics within regulatory frameworks, pointing out the ethical considerations in balancing technological efficiencies against fair governance.

Mohammed Abdul proposed a detailed analysis of blockchain's dual scalability and regulatory compliance challenges through a literature review [20]. They looked into more advanced options like sharding, which splits the blockchain into parts that can be processed in parallel to speed up transactions, and layer-2 methods such as rollups and the Lightning Network, which take transactions off of the main chain to lower latency and make it easier to scale.

7 Conclusion and Future Directions

In conclusion, the proposed Proof of Compliance consensus mechanism offers a transformative solution for compliance checking in the healthcare industry. Blockchain technology and consensus mechanisms provide a transformative solution for checking compliance requirements burdened by regulatory requirements. Compliance with security and privacy policies and regulations indicates the status of healthcare organizations' adherence to them. It also shows the integrity of the operations in handling sensitive patients' health records. This mechanism not only helps to increase the overall trust and reliability of the healthcare ecosystem but also incentivizes participants to maintain high levels of compliance. Overall, the PoC mechanism has the potential to redefine the intersection of trust and compliance with regulatory standards in the healthcare industry, providing a secure and reliable platform for all stakeholders.

As our next step, we want to verify the Proof of Compliance consensus mechanism formally. Formal verification of PoC is essential in establishing its reliability and robustness, particularly in compliance-sensitive sectors such as healthcare. This process involves creating precise mathematical models of the PoC protocol to prove unequivocally that it adheres to its intended compliance rules under all conditions. This rigorous analysis helps ensure that the PoC mechanism meets performance and security standards and dynamically aligns with evolving regulatory requirements, fostering trust and facilitating wider adoption in healthcare industries. Such verification is essential for confirming the security and effectiveness of PoC systems before they are deployed in sensitive and critical healthcare application environments.

Acknowledgements

This work was partially supported by the U.S. National Science Foundation under Grant No. 1822118 and 2226232, the member partners of the NSF IUCRC Center for Cyber Security Analytics and Automation – Statnett, AMI, NewPush, Cyber Risk Research, NIST, and ARL – the State of Colorado (grant #SB 18-086), and the authors' institutions. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or other organizations and agencies.

References

[1] Md Al Amin, Amani Altarawneh, and Indrajit Ray. 2023. Informed Consent as Patient Driven Policy for Clinical Diagnosis and Treatment: A Smart Contract Based Approach. In *Proceedings of the 20th International Conference on Security and Cryptography - SECRYPT*. INSTICC, SciTePress, 159–170.

[2] Md Al Amin, Hemanth Tummala, Rushabh Shah, and Indrajit Ray. 2024. Balancing Patient Privacy and Health Data Security: The Role of Compliance in Protected Health Information (PHI) Sharing. In *Proceedings of the 21st International Conference on Security and Cryptography - SECRYPT*. INSTICC, SciTePress, 211–223.

[3] Mauro Lemus Alarcon, Minh Nguyen, Saptarshi Debroy, Naga Ramya Bhamidipati, Prasad Calyam, and Abu Mosa. 2021. Trust model for efficient honest broker based healthcare data access and processing. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 201–206.

[4] Puzant Balozian and Dorothy Leidner. 2017. Review of IS security policy compliance: Toward the building blocks of an IS security theory. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems* 48, 3 (2017), 11–43.

[5] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. 2017. Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM international conference on management of data*. 1085–1100.

[6] Adam Gągol, Damian Leśniak, Damian Straszak, and Michał Świętek. 2019. Aleph: Efficient atomic broadcast in asynchronous networks with byzantine nodes. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. 214–228.

[7] Ankit Gangwal, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. 2023. A survey of Layer-two blockchain protocols. *Journal of Network and Computer Applications* 209 (2023), 103539.

[8] José A García-Berná, Raimel Sobrino-Duque, Juan M Carrillo de Gea, Joaquín Nicolás, and José L Fernández-Alemán. 2022. Automated Workflow for Usability Audits in the PHR Realm. *International Journal of Environmental Research and Public Health* 19, 15 (2022), 8947.

[9] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. 2020. Sok: Layer-two blockchain protocols. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*. Springer, 201–226.

[10] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. 2007. PeerReview: Practical accountability for distributed systems. *ACM SIGOPS operating systems review* 41, 6 (2007), 175–188.

[11] Tina Highfill. 2019. Do hospitals with electronic health records have lower costs? A systematic review and meta-analysis. *International Journal of Healthcare Management* (2019).

[12] Sushil Kumari Jindal and Faryal Raziuddin. 2018. Electronic medical record use and perceived medical error reduction. *International Journal of Quality and Service Sciences* 10, 1 (2018), 84–95.

[13] Dae-young Kim, Lavanya Elluri, and Karuna P. Joshi. 2021. Trusted Compliance Enforcement Framework for Sharing Health Big Data. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, Baltimore, MD, USA, 4715–4724.

[14] Shinhae Kim and Sungjae Hwang. 2023. EtherDiffer: Differential Testing on RPC Services of Ethereum Nodes. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1333–1344.

[15] Jennifer King, Vaishali Patel, Eric W Jamoom, and Michael F Furukawa. 2014. Clinical benefits of electronic health record use: national findings. *Health services research* 49, 1pt2 (2014), 392–404.

[16] Jared Koreff, Martin Weisner, and Steve G. Sutton. 2021. Data analytics (ab)use in healthcare fraud audits. *International Journal of Accounting Information Systems* 42 (2021), 100523.

[17] Juhee Kwon and M Eric Johnson. 2013. Security practices and regulatory compliance in the healthcare industry. *Journal of the American Medical Informatics Association* 20, 1 (2013), 44–51.

[18] Divakaran Liginlal, Inkook Sim, Lara Khansa, and Paul Fearn. 2012. HIPAA Privacy Rule compliance: An interpretive study using Norman's action theory. *Computers & Security* 31, 2 (2012), 206–220.

[19] Scholas Mbonihankuye, Athanase Nkunzimana, and Ange Ndagijimana. 2019. Healthcare data security technology: HIPAA compliance. *Wireless communications and mobile computing* 2019 (2019), 1–7.

[20] Shezon Saleem Mohammed Abdul. 2024. Navigating Blockchain's Twin Challenges: Scalability and Regulatory Compliance. *Blockchains* 2, 3 (2024), 265–298.

[21] Wilnellys Moore and Sarah Frye. 2019. Review of HIPAA, part 1: history, protected health information, and privacy and security rules. *Journal of nuclear medicine technology* 47, 4 (2019), 269–272.

[22] Arno Nuijten, Mark Van Twist, and Martijn Van der Steen. 2015. Auditing interactive complexity: Challenges for the internal audit profession. *International Journal of Auditing* 19, 3 (2015), 195–205.

[23] William Pourmajidi, Lei Zhang, John Steinbacher, Tony Erwin, and Andriy Miranskyy. 2021. Immutable log storage as a service on private and public blockchains. *IEEE Transactions on Services Computing* 16, 1 (2021), 356–369.

[24] Babak Bashari Rad, Harrison John Bhatti, and Mohammad Ahmadi. 2017. An introduction to docker and analysis of its performance. *International Journal of Computer Science and Network Security (IJCSNS)* 17, 3 (2017), 228.

[25] B Vasantha Rani and Parminder Singh. 2022. A survey on electronic health records (EHRS): Challenges and solutions. In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 655–658.

[26] Office for Civil Rights (OCR). 2008. HIPAA Enforcement. https://www.hhs.gov/hipaa/for-professionals/compliance-enforcement/index.html Last Modified: 2021-06-28T08:59:34-0400.

[27] Sharon Silow-Carroll, Jennifer N Edwards, and Diana Rodin. 2012. Using electronic health records to improve quality and efficiency: the experiences of leading hospitals. *Issue Brief (Commonw Fund)* 17, 1 (2012), 40.

[28] Jovan Stevovic, Fabio Casati, Bilal Farraj, Jun Li, Hamid R. Motahari-Nezhad, and Giampaolo Armellin. 2023. Compliance Aware Cross-Organization Medical Record Sharing. *IEEE Symposium on Integrated Network Management* (2023).

[29] John Ugoani and Grace Iyi Ibeenwo. 2022. External Audit Process Failures: Unethical Practices and Business Demise. *Business, Management and Economics Research* 8, 1 (2022), 1–11.

[30] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.