DISSERTATION

HEALTHCARE SECURITY AND PRIVACY POLICY COMPLIANCE: A BLOCKCHAIN AND SMART CONTRACT-BASED ASSURANCE FRAMEWORK

Submitted by

Md Al Amin

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2026

Doctoral Committee:

Advisor: Indrajit Ray

Indrakshi Ray
Yashwant K Malaiya
Leo R Vijayasarathy

ABSTRACT

HEALTHCARE SECURITY AND PRIVACY POLICY COMPLIANCE: A BLOCKCHAIN AND SMART CONTRACT-BASED ASSURANCE FRAMEWORK

Access to electronic health records (EHRs) is heavily regulated by various policies, including federal-level policies, state-level statutes, international data protection laws, and local and organizational-level policies. These policies may include procedures to ensure compliance with other organizational-level regulations. In addition, individual patients can establish agreements, formally known as patient-provider agreements (PPA), with their healthcare providers to express their consent to access or share their protected health information (PHI). When such policies are adequately specified and implemented, they go a long way toward protecting EHR data. However, research has shown that significant policy compliance problems or gaps often go undetected until after a breach or security incident. Further, a recent study shows that subcultures within a healthcare organization influence whether employees violate policies, perhaps unintentionally. These observations motivate us to revisit the compliance and provenance aspects of policies.

This dissertation proposes a blockchain-powered, smart contract-based policy-compliance assurance framework to enforce patient-provider agreements and other applicable policies and attributes, ensuring policy compliance and provenance in the healthcare sector. This work proposes a novel compliance review mechanism, Proof of Compliance (PoC), that conducts reviews through a set of independent, distributed, decentralized auditor nodes from various stakeholders, such as healthcare organizations, insurance companies, federal and other government agencies, regulatory agencies, and others mandated by the business requirements. Blockchain smart contracts appear to be a promising new technology for enforcing policies. In addition, blockchains' immutable storage properties and strong integrity guarantees provide hope that an adequate trail of policy compliance (or non-compliance) can be maintained, thereby facilitating provenance.

ACKNOWLEDGEMENTS

This work was partially supported through grants from the following agencies and organizations: the U.S. National Science Foundation (grant #1822118, #2226232), the member partners of the NSF IUCRC Center for Cyber Security Analytics and Automation – AMI, NewPush, Statnett, Cyber Risk Research, NIST and ARL – the State of Colorado (grant #SB 18-086) and the Colorado State University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation, or other organizations and agencies

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Electronic health records (EHRs) have emerged as a cornerstone in modernizing healthcare. This offers numerous benefits that enhance efficiency, accuracy, and quality of care and provide a patient-centered approach to healthcare service delivery [14, 15]. These systems provide immediate and remote access to patient data, a critical feature that streamlines administrative and medical decision-making processes [16, 17]. By transitioning from paper-based systems, EHRs significantly reduce errors and costs commonly associated with manual and paper-based record-keeping, enhancing patient safety and care quality and optimizing resources [18, 19]. One of the critical advantages of EHRs is their ability to promote interoperability across different healthcare platforms through machine-readable data formats such as *Extensible Markup Language (XML)* and various protocols like *Health Level Seven - Clinical Document Architecture (HL7 CDA)*, *Fast Healthcare Interoperability Resources (FHIR)* [20, 21].

The HIPAA law binds healthcare providers (outlined in *45 CFR §164.524(c)(2)*) to provide healthcare data to patients. This interconnectedness enables seamless sharing of patient data among healthcare providers, thereby improving continuity of care and the overall healthcare experience [22, 23]. The trend of storing patient information electronically in local databases or cloud servers underscores the healthcare industry's commitment to improving the efficiency and precision of patient care [24, 25]. Furthermore, EHR systems play a vital role in strengthening healthcare services. The constant availability of up-to-date health records is essential for maximizing treatment productivity and ensuring timely and accurate medical interventions [26, 27].

However, this digital transformation also brings complex information security and privacy challenges. These are critical for maintaining patient trust and compliance with the applicable regulatory standards and data protection laws. Data security and privacy violations are increasingly seen across the healthcare sector [28]. Many of these can be attributed to this industry's widespread use of smartphones, internet-connected devices, sensors, wearable devices, mobile-based health

applications, and other IT-dependent services [29]. Cybersecurity risks, potential breaches, and the need for stringent access controls raise significant questions. Additionally, interoperability issues, human factors such as user authentication, and the imperative of legal compliance further compound the challenges associated with implementing and maintaining EHR systems [30, 31].

Laws, policies, and regulations are pivotal to addressing EHR challenges and safeguarding healthcare data security and patient privacy, thereby promoting trust between patients and healthcare providers. In various global regions, diverse privacy standards, including the *General Data Protection Regulation (GDPR)* in Europe [32], the *Health Insurance Portability and Accountability Act (HIPAA)* in the United States [33], and *My Health Record (MHR)* in Australia [34], have been established to protect patient privacy and personal data [20].

Healthcare organizations must implement technical, administrative, and physical safeguards to secure EHRs [33]. By enforcing these safeguards, HIPAA helps prevent unauthorized access, data breaches, identity theft, and other unwanted security incidents. Furthermore, HIPAA mandates the implementation of privacy policies and procedures to govern the use and disclosure of patient information. It grants patients certain rights, including access to and amendment of their medical records. It requires healthcare providers to obtain patient consent for specific uses and disclosures of their information.

HIPAA also imposes penalties for noncompliance, incentivizing healthcare organizations to prioritize data security and privacy. These penalties can range from fines to criminal charges, depending on the severity of the violation. The violation can also lead to reputational damage, eroding trust from clients and the public. Organizations may face exclusion from federal programs, financial strain due to legal costs, and increased scrutiny. HIPAA violations can also lead to provider confusion, increased documentation time, alert fatigue, and potential patient safety issues [35]. According to the *Office of Civil Rights Data* study, since October 2009, massive security breaches may have affected more than half of the population in the USA [36]. At least 173 million medical records were breached due to the policy noncompliance.

Table 1.1: OCR HHS - Compliance Complaint [1–4]

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

Table 1.1 shows the number of compliance complaints received by the *U.S. Department of Health and Human Services (HHS) Office for Civil Rights (OCR)* [1]. The major reasons for the complaints are *(i) impermissible uses and disclosures of protected health information (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.*

While advancements in security and privacy technology are essential for enhanced protection of patient data from such incidents, substantial evidence indicates that improper adoption, implementation, and enforcement of policies contribute significantly to unauthorized access—without a legitimate "need to know"—to EHR data [37]. Whether intentional or unintentional, users are often granted access privileges they should not possess. Policies are often not adhered to accurately, resulting in delays in checking or implementing access control rules. Instances have been observed in which identical roles and privileges are assigned to all employees. Additionally, individual patient-level policies are often not rigorously enforced. Auditing and monitoring gaps are also prevalent, typically occurring only in response to severe complaints or legal mandates [38].

Sarkar et al. [39] present an alternative perspective on security and privacy breaches within the healthcare industry. They explore the existence of distinct professional subcultures within healthcare organizations. Through a qualitative study, the authors identify factors that inadvertently lead these subculture groups to violate information security policies. For instance, when a doctor, positioned at the top of the subculture hierarchy, seeks access to information beyond their authorized scope, lower-ranking employees within the subculture may not intervene to prevent it.

Another important factor contributing to policy violations is the lack of effective enforcement mechanisms arising from the existence of multiple policy bodies. Healthcare is often subject to a complex web of regulations and guidelines established by various organizations at the national and

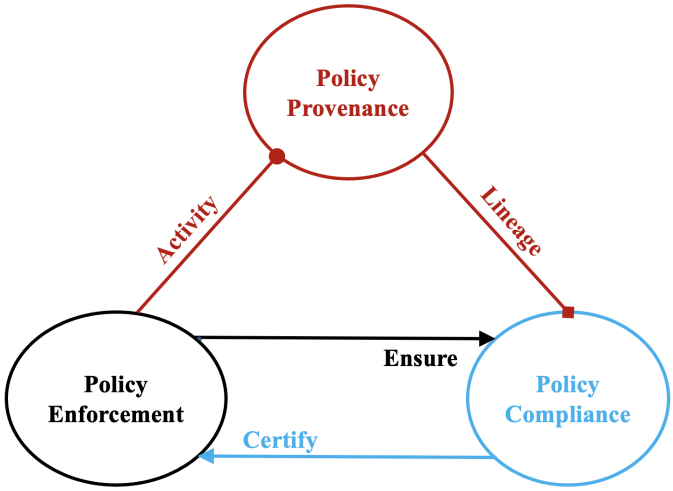

Figure 1.1: Policy Enforcement, Compliance, and Provenance [2, 12]

international levels. This diversity of policy bodies can create challenges and potential conflicts for healthcare providers and professionals [40, 41].

While ensuring the enforcement of system security policies is crucial, it is equally vital to establish provenance to validate adherence to these policies. Figure 1.1 illustrates the interplay among enforcement, compliance, and provenance. Effective policy enforcement safeguards healthcare data against unauthorized access and misuse. When policies are adequately enforced, policy compliance becomes possible, as it requires aligning all actions with the relevant policies. However, this compliance alone lacks quantifiable measurement or validation. Maintaining the integrity of policy enforcement activities is essential for accurate assessment of policy compliance. Enforcement integrity ensures that events are accurately recorded as they unfold. An independent auditor conducts a policy audit to verify the policy's compliance status. Provenance, in turn, offers a chronological record of policy enforcement activities as they unfold.

Security and privacy policy violations do not arise solely from software bugs or other technical issues. They also stem from various non-technical factors, including user ignorance, misuse of technology, inadequate training and experience with the required systems, and limited awareness of regulatory and legal implications. Moreover, organizational and professional influences, including subcultures, technical privileges, and administrative authority, further contribute to noncompliance. Influences and technical issues are the main drivers of provenance for policy compliance.

Tampering activities occurred in the applied policies and rules, audit trails, access logs, and other provenance sources. These combined factors pose significant challenges to maintaining provenance in policy compliance. Tampering with applied policies and rules, audit trails, access logs, and other provenance sources undermines the integrity of compliance mechanisms. This raises serious concerns about the transparency and accountability of policy compliance, making it difficult to attribute responsibility or hold individuals accountable for their actions, especially in cases of policy violations.

This dissertation proposes a blockchain-powered, smart contract-based policy-compliance assurance framework to enforce patient-provider agreements (PPAs) and other applicable policies and attributes, ensuring policy compliance and provenance in the healthcare sector. This research introduces a novel compliance review mechanism that conducts reviews through independent, distributed, and decentralized auditor nodes from various stakeholders, including healthcare organizations, insurance companies, federal and other government agencies, regulatory agencies, and other entities mandated by business requirements. The consensus process in a blockchain network ensures that smart contracts function as intended, without user interference. The proposed architecture enforces applicable policies and PPAs by keeping audit logs linked to enforced policies and enforcing access based on smart contracts. Since the blockchain network stores all user policies and event logs, it provides provenance services. Once smart contracts are fully deployed and functional, the conditions and mechanisms written into the code can't be changed.

The remainder of this dissertation is organized as follows: Blockchain and related technologies used in this research are explained in relation to HIPAA and other data protection laws for healthcare policy compliance in Chapter 2. Chapter 3 contains the related works for healthcare policy compliance. Chapter 4 presents a method for incorporating informed consent into the PPA and the contact-based enforcement mechanism. A private or enterprise blockchain-based policy provenance mechanism is discussed in Chapter 5. Chapter 6 discusses a blockchain consensus mechanism, Proof of Compliance (PoC), for the proposed framework to verify compliance status. The paper concludes with a brief discussion, a conclusion, and future directions in Chapter 7.

Chapter 2

Preliminaries/Background

This chapter presents the foundational concepts and theoretical basis necessary to understand the development of a blockchain-based framework for healthcare security and privacy policy compliance. Its primary objective is to establish the context, definitions, and core technologies that guide this research. We discuss technologies and related topics relevant to the compliance framework to help the interested reader understand and relate the proposed approach's functionalities. Overall, this chapter lays the groundwork for the proposed framework and methodology, linking emerging blockchain capabilities with regulatory requirements such as HIPAA and GDPR to achieve verifiable policy compliance. It explains how these properties can enhance trust and automate compliance in distributed healthcare environments.

2.1 Blockchain and Related Technologies Fundamentals

2.1.1 Blockchain

Blockchain is a distributed ledger technology (DLT) designed to securely record transactions across a decentralized network of computers (nodes) in a transparent, tamper-resistant, and verifiable manner [42]. Unlike traditional centralized databases controlled by a single authority, blockchains maintain a consensus-driven record of data shared among participants, ensuring trust without intermediaries [43]. Each transaction is grouped into a block, cryptographically linked to its predecessor, forming an immutable chronological chain. This structure guarantees data integrity, transparency, and non-repudiation, as any attempt to alter historical data would invalidate subsequent blocks. Blockchain networks rely on consensus algorithms, such as *Proof of Work (PoW)*, *Proof of Stake (PoS)*, or *Practical Byzantine Fault Tolerance (PBFT)*, to validate transactions and maintain synchronization across their distributed nodes [44]. Beyond its origin in cryptocurrencies, blockchain

has evolved into a foundational technology for secure data sharing, digital identity management, and policy compliance across various sectors [45].

In the healthcare domain, blockchain enables trusted data exchange, patient-driven consent management, and automated policy enforcement through smart contracts, thus ensuring verifiable compliance with privacy regulations such as HIPAA and GDPR while enhancing interoperability and transparency in healthcare ecosystems [46].

Main Characteristics of Blockchain

Blockchain technology exhibits several distinctive characteristics that make it suitable for secure, transparent, and auditable data management in healthcare policy compliance [47]. The main characteristics are summarized as follows:

- **Decentralization:** Blockchain operates on a peer-to-peer (P2P) network where each node maintains a synchronized copy of the ledger. This eliminates the need for intermediaries, minimizes single points of failure, and enhances trust among distributed healthcare stakeholders.
- **Immutability:** Once a transaction is validated and recorded in a block, it cannot be altered or deleted without the consensus of the network. This ensures reliable, tamper-proof audit trails for healthcare operations such as patient consent management and access verification.
- **Transparency and Traceability:** All network participants can verify transactions stored on the ledger, ensuring end-to-end traceability. In healthcare, this provides accountability by maintaining verifiable logs of policy enforcement and inter-organizational data exchange.
- **Security:** Blockchain employs advanced cryptographic mechanisms, including digital signatures and hashing, to protect transaction integrity and participant identities. Each block contains the cryptographic hash of the previous one, making the chain resistant to tampering.
- **Consensus Mechanism:** Transactions are collectively validated through consensus algorithms such as *PoW*, *PoS*, or *PBFT*. These protocols ensure network-wide agreement without requiring a central authority.

- **Smart Contracts:** Blockchain supports self-executing scripts, known as smart contracts, which automatically enforce predefined conditions when triggered. In healthcare policy compliance, they can encode access control rules, consent policies, and privacy regulations for automated enforcement.
- **Tokenization and Incentivization:** Many blockchain platforms issue digital tokens representing assets, rights, or privileges. These can be used to incentivize participation, compliance, or auditing activities within healthcare ecosystems.
- **Resilience and Fault Tolerance:** The distributed nature of blockchain provides high resilience to failures or cyberattacks. Even if specific nodes are compromised, the network continues operating without data loss, ensuring the continuous availability of compliance and audit records.

These characteristics collectively make blockchain a robust foundation for developing secure, transparent, and accountable healthcare data ecosystems, enabling verifiable compliance with privacy and regulatory standards.

2.1.2 Smart Contract

A smart contract is a self-executing digital agreement encoded as a program on the blockchain that automatically enforces predefined terms and conditions without intermediaries [48]. First introduced by Nick Szabo in 1994, smart contracts embody the principle of *"code as law,"* meaning that contractual clauses are translated into executable code rather than relying solely on legal interpretation. Each smart contract is stored on the blockchain and executed deterministically by network nodes when specific conditions are met, ensuring transparency, reliability, and tamper resistance [49].

In a blockchain network, smart contracts handle key functionalities, including data validation, access control, payment processing, and policy enforcement [50]. Once deployed, the contract's code and results are immutable, making it a trusted mechanism for automating multi-party interactions.

In healthcare security and privacy policy compliance, smart contracts play a crucial role in automating consent management, enforcing policies, and facilitating regulatory audits. For example, a patient's consent agreement can be encoded as a smart contract that grants data access only to authorized healthcare providers under specific conditions, automatically logging every access event for compliance verification. This enables fine-grained access control, non-repudiation, and real-time assurance of compliance across distributed healthcare systems. Despite their benefits, smart contracts also pose challenges, including the immutability of buggy code, gas-cost optimization, and alignment with legal frameworks [51]. Therefore, careful design, formal verification, and audit mechanisms are essential for the secure deployment of smart contracts in healthcare environments.

2.1.3 Consensus Mechanism

A **consensus mechanism** is the fundamental protocol that enables nodes in a blockchain network to agree on the validity of transactions and maintain a consistent, tamper-resistant distributed ledger. In decentralized environments without a single authority, consensus algorithms ensure that all participants share a common version of the truth, preventing double-spending, unauthorized modifications, or conflicting transaction histories [52]. Each consensus protocol defines how new blocks are proposed, validated, and appended to the chain. Common consensus mechanisms include:

- **Proof of Work (PoW):** In PoW, network participants known as miners compete to solve complex cryptographic puzzles to validate transactions and generate new blocks. Although highly secure and decentralized, PoW requires substantial computational resources and energy, making it less suitable for resource-constrained healthcare environments.
- **Proof of Stake (PoS):** PoS selects validators based on the number of tokens they hold and are willing to stake as collateral. It is more energy-efficient and achieves faster transaction finality than PoW, making it well-suited for permissioned healthcare blockchains that require efficient, scalable compliance verification.
- **Delegated Proof of Stake (DPoS):** DPoS introduces a voting-based model in which stakeholders elect a limited set of trusted nodes to validate transactions on their behalf. This

approach increases transaction throughput and governance control, which is beneficial for consortium-based healthcare blockchain networks.

- **Practical Byzantine Fault Tolerance (PBFT):** PBFT enables network nodes to reach a consensus even in the presence of faulty or malicious participants. It is commonly used in private or consortium blockchains, such as *Hyperledger Fabric*, providing low-latency and high-trust consensus, which is ideal for healthcare data sharing and compliance auditing.
- **Proof of Authority (PoA):** PoA relies on a predefined set of verified validators whose identities are known and trusted. It ensures high transaction throughput and predictability, making it effective for permissioned healthcare systems where all participants are regulated entities.

In the context of **healthcare security and privacy policy compliance**, selecting an appropriate consensus mechanism requires balancing trust, scalability, privacy, and governance requirements. Table 2.1 compares various consensus mechanisms across criteria such as full form, validation basis, energy consumption, transaction speed, scalability, full tolerance, and others. While PoW ensures maximum decentralization, mechanisms such as PBFT or PoA are more practical for healthcare consortia, where trusted participants collaboratively maintain regulatory compliance and data integrity.

2.1.4 Public, Private, & Consortium Blockchain Network

Blockchain networks are classified based on the accessibility of ledger data, the node participation process, the consensus mechanism, block mining incentives, and other related factors [53]. This section provides an overview of public, private, and consortium blockchain networks, including their properties and applications in the healthcare industry, which aim to deliver improved services and compliance assurance [54].

Public Blockchain Network

A public blockchain network is an open, permissionless distributed ledger in which anyone can join, participate, and verify transactions without prior authorization [55]. In this model, all

Table 2.1: Comparison of Consensus Mechanisms in Blockchain [5–7]

Criteria	PoW	PoS	DPoS	PBFT	PoA
Full Form	Proof of Work	Proof of Stake	Delegated Proof of Stake	Practical Byzantine Fault Tolerance	Proof of Authority
Validation Basis	Computational effort (mining)	Token stake and collateral	Elected delegates by stakeholders	Agreement among trusted nodes	Pre-approved authorities
Energy Consumption	Very High	Low	Low	Moderate	Very Low
Transaction Speed	Slow	Moderate to Fast	Fast	Very Fast	Very Fast
Decentralization Level	High	Moderate to High	Moderate	Low to Moderate	Low
Scalability	Limited	Moderate	High	High	High
Fault Tolerance	High (by design)	High	Moderate	Very High	Moderate
Security Level	Very High (computationally secure)	High (stake-based trust)	Moderate (delegate trust)	Very High (Byzantine fault tolerant)	High (identity-based trust)
Governance Model	Open, public competition	Stakeholder-driven	Delegated and democratic	Permissioned and consensus-based	Centralized authority control
Energy Efficiency	Very Low	High	High	High	Very High
Typical Networks	Bitcoin, Ethereum (legacy)	Ethereum 2.0, Cardano	EOS, Tron	Hyperledger Fabric, Tendermint	VeChain, EnergyWeb
Suitability for Healthcare	Limited due to energy cost	Suitable for scalable private systems	Suitable for consortiums with elected nodes	Highly suitable for healthcare consortium networks	Ideal for regulated healthcare entities with known participants

transactions and blocks are visible to every participant in the network, ensuring transparency, decentralization, and immutability. Consensus mechanisms such as *PoW* and *PoS* are typically used to validate transactions and maintain ledger integrity. Prominent examples include *Bitcoin, Ethereum,* and *Polygon*. Public blockchains offer robust trust guarantees because the ledger's state is collectively maintained by a large number of independent nodes, thereby reducing the risk of tampering or centralized control.

In the context of healthcare data compliance, public blockchains can serve as neutral audit layers, enabling verifiable proof of policy enforcement, consent records, or anonymized compliance events. However, their open accessibility also raises concerns about privacy and scalability, necessitating the integration of off-chain storage or privacy-preserving cryptographic techniques to protect sensitive

health information. Therefore, while public blockchains ensure transparency and trust, their use in healthcare must strike a balance between openness and strict data confidentiality and regulatory compliance requirements.

Private Blockchain Network

A private blockchain network is a permissioned and access-controlled distributed ledger in which participation is restricted to a predefined group of trusted entities [55]. Unlike public blockchains, which allow anyone to read or write data, private blockchains restrict access to authorized participants, typically governed by a central organization or consortium. This model provides greater control, confidentiality, and flexibility in compliance, making it highly suitable for domains such as healthcare, finance, and government [56]. Platforms such as *Hyperledger Fabric, Quorum,* and *Corda* are widely adopted private blockchain frameworks that support fine-grained access control, modular consensus algorithms, and customizable privacy settings.

In the context of healthcare policy compliance, private blockchains enable secure sharing of patient information among hospitals, laboratories, insurance providers, and regulators while preserving data privacy, regulatory alignment, and operational efficiency [57]. Since the participating nodes are known and vetted, private networks can enforce role-based permissions, implement smart contract–driven compliance rules, and maintain auditable transaction records without exposing sensitive information to the public. Thus, private blockchain architectures offer a practical balance between trust, decentralization, and data confidentiality, supporting secure and compliant healthcare data ecosystems.

Consortium Blockchain Network

A consortium blockchain network represents a hybrid model that combines the transparency of public blockchains with the controlled access of private ones [58]. In this setup, the network is governed by a group of preselected organizations, rather than a single entity, forming a semi-decentralized trust structure. Each participating member operates a validating node, and consensus

decisions are made collaboratively by the consortium. This model enhances transparency, trust, and interoperability while maintaining restricted access and oversight of compliance.

In healthcare environments, consortium blockchains are particularly valuable because they enable secure collaboration among multiple stakeholders, including hospitals, diagnostic centers, insurance providers, and regulatory authorities [59]. For instance, patient consent updates, clinical trial data, or policy compliance records can be shared and verified among consortium members without relying on a single centralized authority. Frameworks such as *Hyperledger Fabric, R3 Corda,* and *Quorum* are often used to implement such systems, offering modular privacy controls and customizable consensus mechanisms.

Table 2.2 compares public, private, and consortium blockchain networks across criteria such as access control, governance, consensus mechanism, transaction speed, scalability, data privacy, trust model, auditability, and others. By balancing decentralization with governance accountability, consortium blockchain networks provide an ideal foundation for multi-institutional healthcare compliance frameworks, ensuring data integrity, regulatory alignment, and mutual trust across the healthcare ecosystem.

2.1.5 Layer 1 Blockchain Network

A *Layer 1* blockchain network refers to the base architecture of a blockchain system, which forms the foundational layer responsible for consensus, data structure, and transaction validation [60]. It serves as the primary protocol for processing and recording all transactions directly on the blockchain ledger. Prominent examples of Layer 1 networks include *Bitcoin*, *Ethereum*, and *Binance Smart Chain*, each providing the essential components of a decentralized network, such as peer-to-peer communication, distributed consensus algorithms, and cryptographic validation.

In Layer 1 environments, scalability and performance are typically constrained by the consensus mechanism (e.g., *Proof of Work* or *Proof of Stake*), prompting innovations such as sharding and rollups to enhance throughput [61, 62]. In the context of healthcare policy compliance, Layer 1 networks provide trust, immutability, and transparency, serving as the secure foundation for

Table 2.2: Comparison of Public, Private, and Consortium Blockchain Networks [8, 9]

Criteria	Public Blockchain	Private Blockchain	Consortium Blockchain
Access Control	Permissionless — anyone can join, read, and write data	Permissioned — only authorized participants can join and transact	Partially permissioned — access limited to consortium members
Governance	Fully decentralized; managed by public consensus	Centralized; managed by a single organization	Federated; governed by multiple trusted entities
Consensus Mechanism	Typically Proof of Work (PoW) or Proof of Stake (PoS)	Configurable (e.g., RAFT, PBFT, or custom)	Collaborative consensus among consortium members (e.g., PBFT, Raft)
Transaction Speed	Low to moderate due to network size and public validation	High, as nodes are fewer and trusted	Moderate; faster than public but slower than fully private systems
Scalability	Limited — constrained by on-chain validation	High — smaller network size and optimized protocols	Moderate — balanced scalability and decentralization
Data Privacy	Low — transactions are publicly visible	High — data visibility restricted to authorized participants	Moderate to high — data shared within trusted consortium only
Trust Model	Trustless — relies on public consensus	Trusted — depends on central authority	Semi-trusted — distributed trust among consortium members
Auditability	Full transparency and traceability	Controlled auditing; internal logs	Shared auditability among consortium members
Regulatory Compliance	Challenging due to data exposure	Easier to enforce compliance controls	Highly suitable for multi-party compliance enforcement
Use Case Suitability in Healthcare	Public verification of anonymized audit trails or research data	Internal data management within a single healthcare organization	Inter-organizational collaboration (e.g., hospitals, regulators, insurers)
Examples	Ethereum, Bitcoin, Polygon	Hyperledger Fabric (private mode), Quorum (private)	Hyperledger Fabric (consortium mode), R3 Corda

deploying higher-level applications—such as smart contract–based compliance frameworks. These characteristics make Layer 1 blockchains a reliable platform for recording immutable audit trails and enforcing data access policies in a verifiable and tamper-resistant manner.

2.1.6 Layer 2 Blockchain Network

A Layer 2 blockchain network operates as an extension or scaling solution built on top of a Layer 1 blockchain [60, 63]. It is designed to enhance transaction throughput, reduce latency, and minimize transaction costs without compromising the security and decentralization provided by the underlying base layer. Layer 2 solutions achieve this by processing most transactions off-chain and periodically anchoring summarized results back to the Layer 1 network for final settlement and

verification. Common Layer 2 technologies include *State Channels*, *Sidechains*, *Rollups* (*Optimistic* and *Zero-Knowledge*), and *Plasma* frameworks [64, 65]. Platforms such as *Polygon*, *Arbitrum*, and *Optimism* exemplify practical Layer 2 implementations built on *Ethereum* [66, 67].

In healthcare data management and policy compliance, Layer 2 networks enable efficient execution of smart contracts and secure data interactions at scale, thereby making blockchain-based compliance frameworks more practical for real-world deployment. By significantly reducing gas costs and improving transaction speed, Layer 2 solutions facilitate large-scale healthcare operations—such as consent verification, access control, and audit logging—while maintaining cryptographic integrity and verifiable linkage to the Layer 1 ledger.

While Layer 1 and Layer 2 blockchain networks share the common goal of ensuring secure and verifiable digital transactions, they differ significantly in their architecture, performance, and application scope. Layer 1 networks form the core of consensus and security, providing immutability, decentralization, and trust through native validation protocols such as *PoW* or *PoS*. However, this robustness often comes at the cost of limited scalability and high transaction fees, making large-scale data operations costly and time-consuming [60]. In contrast, Layer 2 networks are designed to enhance scalability and efficiency by offloading computation and transaction processing from the base layer, while still periodically committing verified results to the base layer. This approach preserves the security guarantees of Layer 1 while enabling high throughput and low-cost execution [60].

In the context of blockchain-based healthcare policy compliance, Layer 1 networks can serve as the trusted audit backbone, ensuring data integrity and non-repudiation. In contrast, Layer 2 solutions can support real-time operations, such as frequent consent updates, access requests, or dynamic compliance verification, enabling seamless integration. Together, they provide a balanced architecture that combines the security of Layer 1 with the performance and scalability of Layer 2, making blockchain adoption both technically feasible and economically viable in healthcare ecosystems.

2.1.7 Blockchain Main Network & Test Network

Blockchain networks maintain a distributed ledger that contains data and executable code (smart contracts) as transactions in cryptographically and chronologically linked blocks. All participants' nodes in the network maintain the same copy of the ledger, ensuring data consistency and integrity, as verified through cryptographic hash values [68]. Transactions are submitted by users and signed using their private keys. They are then verified using the corresponding public keys and, after completing blocks via the consensus mechanism, added to the ledger, ensuring agreement among all participating network nodes. This section discusses an overview of the functionalities of the blockchain main and test networks for developing, deploying, and testing blockchain-based or decentralized applications (dApps).

Blockchain Main Network

A *Blockchain Main Network (Mainnet)* is the fully operational, publicly accessible version of a blockchain platform on which real transactions occur and hold actual economic value. It represents the production environment that has transitioned beyond the testing and development phases (testnets) and operates with its native cryptocurrency, such as *Ether (ETH)* on *Ethereum* or *MATIC* on *Polygon*. All activities on the mainnet are permanently recorded on the immutable distributed ledger and validated by network nodes through consensus protocols such as *Proof of Work* or *Proof of Stake*. Deploying smart contracts or *decentralized applications* on the mainnet incurs real transaction fees (gas costs), reflecting the computational resources required to process and confirm transactions.

Once a block is finalized and added to the ledger, data and code are permanent on the main network [69]. All blocks are accessible to all participants' nodes in the public network and to selected nodes in the private and consortium networks. Therefore, care must be taken to ensure data security (confidentiality) and users' privacy, and to address smart contract bugs and other issues in the deployed code, which remains permanently on the ledger [70]. To avoid these challenges, it is crucial to thoroughly evaluate and test any applications and smart contracts before deploying

them to the main network. The test networks play a key role in this context, as discussed in the next section.

Blockchain Test Network

A *Blockchain Test Network (Testnet)* is a dedicated experimental environment that replicates the functionality of a main blockchain network but uses valueless tokens to enable safe, cost-free testing [71]. Testnets enable developers and researchers to deploy smart contracts, simulate transactions, and evaluate network performance without the financial risk or security implications associated with mainnet deployment. Popular test networks include *Ethereum Goerli, Sepolia,* and *Polygon Amoy*, each designed to mirror mainnet conditions, including consensus algorithms, block times, and transaction structures [72]. Tokens on these networks are typically obtained from cryptocurrency faucets, which distribute small test amounts for development use [73].

Testnets are invaluable for prototyping and validating smart contract logic, such as patient consent workflows, policy enforcement conditions, and access control rules, before migrating to the production mainnet. This ensures that issues related to gas efficiency, logic correctness, and data privacy are resolved in a controlled environment. Ultimately, the use of testnets provides a secure and iterative development pathway, reducing the likelihood of errors, vulnerabilities, and non-compliance in live healthcare blockchain systems.

In the context of healthcare security and privacy policy compliance, blockchain networks serve as a trusted, transparent execution layer for deploying compliance-related smart contracts, maintaining immutable audit trails, and verifying consent or policy adherence in real time. Because data recorded on the mainnet is tamper-proof and globally verifiable, they provide high assurance of accountability and non-repudiation [74]. However, it also requires strict attention to data privacy, cost management, and scalability when handling sensitive healthcare information.

In the proposed framework, we utilize blockchain test networks (*Sepolia, Georli*) to deploy smart contracts and execute transactions, thereby storing data and performing other operations. Table 2.3 compares blockchain main and test networks across criteria such as purpose, token value, data permanence, transaction cost, security level, healthcare compliance use, and others. These help

Table 2.3: Comparison Between Blockchain Main Network and Test Network

Criteria	Main Network (Mainnet)	Test Network (Testnet)
Purpose	Production environment where real transactions occur and have monetary value	Development environment for testing and validation without financial risk
Token Value	Uses real cryptocurrency (e.g., ETH, MATIC) with market value	Uses valueless tokens obtained from faucets for experimental use
Transaction Cost	Real gas fees paid in native cryptocurrency	No real cost; only simulated gas fees
Data Permanence	Permanent and immutable record stored on the public ledger	Temporary or resettable data for testing purposes
Network Accessibility	Publicly accessible and globally verifiable ledger	Publicly available but isolated from mainnet operations
Security Level	High — validated by a large decentralized network of nodes	Moderate — suitable for controlled testing, not for production data
Healthcare Compliance Use	Deployment of verified smart contracts for real compliance monitoring and audit trails	Prototyping, simulation, and performance testing of compliance logic
Examples	Ethereum Mainnet, Polygon Mainnet, Bitcoin Network	Goerli Testnet, Sepolia Testnet, Polygon Amoy Testnet

us evaluate gas costs, measure the time required to write to and read from blockchain networks, and assess the functionality of the components of the proposed healthcare security and privacy policy compliance frameworks.

2.1.8 Blockchain Transaction & Multi-Signature Transaction

The state of the blockchain is modified by writing data to the distributed ledger. Users submit signed transactions using their private keys to write data to the ledger via the network consensus mechanism. Before finalization, all transactions are verified using their corresponding public keys. This section discusses single-signature and multi-signature transactions and their application in the proposed framework for performing various experimental evaluations.

Blockchain Transaction

A blockchain transaction represents a digitally signed record that facilitates the transfer of data, assets, or information between network participants in a verifiable and tamper-resistant manner. Each transaction includes essential elements, such as sender and receiver addresses, input and output values, digital signatures, and a timestamp [75, 76]. Once a transaction is created, it is

broadcast to the network, validated by nodes through a consensus mechanism (e.g., Proof of Work or Proof of Stake), and permanently recorded in a block on the distributed ledger. Because every block references the hash of its predecessor, the resulting chain of transactions becomes immutable, ensuring traceability and integrity [77].

In the context of healthcare policy compliance, blockchain transactions can encode sensitive operations such as patient consent approvals, policy updates, or audit log entries. These transactions, when executed via smart contracts, enable automated enforcement of data-sharing agreements and provide transparent, verifiable evidence of compliance among healthcare entities.

Multi-Signature Transaction

A multi-signature (multisig) transaction is a blockchain mechanism that requires multiple authorized digital signatures to approve and execute a transaction, rather than relying on a single private key [78, 79]. This approach enhances security, control, and accountability by ensuring that no single entity can unilaterally authorize critical operations. A multisig wallet is typically configured as an M-of-N scheme, where any M out of N authorized participants must sign a transaction for it to be validated by the network [80].

In blockchain-based healthcare systems, multi-signature transactions play a vital role in policy enforcement and data access governance [81]. For example, patient data access or record modification may require joint approval from the patient and the healthcare provider to ensure compliance with privacy and consent policies. This multi-party authorization model aligns with healthcare's legal and ethical requirements by providing non-repudiation, shared accountability, and protection against insider misuse. Furthermore, multi-signature mechanisms can be combined with smart contracts to automate conditional approvals, thereby establishing a transparent and verifiable compliance framework within the blockchain network.

In this proposed healthcare policy compliance framework, multi-signature transactions are used to approve healthcare data sharing beyond the treatment team for advanced diagnosis, marketing, and research purposes [3]. In an emergency, the provider and senior supervisor review, evaluate, and approve emergency PHI access using a multi-signature transaction to save a patient's life [10].

Additionally, we introduce this transaction to process health insurance claims and detect and mitigate insurance fraud [82]. All involved entities are accountable for signing any transaction or operation using their private keys, since the keys are assumed not to be compromised and publicly available.

2.1.9 Blockchain Wallet & Faucet Cryptocurrency

In the following, we discuss blockchain wallets and faucet cryptocurrency. Wallets store users' wallet addresses and private keys and sign transactions before submitting them to the blockchain network. Additionally, faucet cryptocurrency is used to cover the transaction costs on the blockchain test network.

Blockchain Wallet

A blockchain wallet is a digital tool that enables users to securely store, manage, and interact with cryptographic keys used to authorize blockchain transactions. Each wallet generates a unique pair of cryptographic keys. This public key serves as the wallet's address for receiving transactions, and the corresponding private key is used to sign and authorize outgoing transactions [83]. The private key must remain confidential, as its compromise can result in irreversible loss of access or unauthorized transactions. Blockchain wallets can be classified as software wallets (hot wallets) and hardware wallets (cold wallets) based on their internet connectivity [84].

In healthcare platforms, blockchain wallets can serve as digital identities for stakeholders, including patients, healthcare providers, and regulatory authorities. Through wallet-based authentication, participants can securely sign consent forms, approve policy updates, or access sensitive medical data under predefined conditions of a smart contract. Thus, blockchain wallets not only facilitate secure transactions of assets and data but also serve as a trust anchor in ensuring identity verification, non-repudiation, and compliance traceability in decentralized healthcare ecosystems. In the proposed framework, we use the *Metamask Digital Wallet* to sign transactions on behalf of various users, including doctors, patients, and others, for the experimental evaluations [85, 86].

Faucet Cryptocurrency

In blockchain networks, every transaction—such as writing data or deploying code—incurs a computational cost determined by the operation's size and complexity. Users need to pay transaction fees (also known as gas fees) to compensate network validators and miners for processing and storing this data [87]. On main networks (mainnets) such as *Bitcoin* and *Ethereum*, these fees are paid using real cryptocurrencies (e.g., *BTC*, *ETH*, or *MATIC*), which costs actual money.

In contrast, test networks (testnets) are designed for development and experimentation without financial risk. Transactions on testnets do not require real currency; instead, users use faucet currency or tokens that mimic the behavior of actual cryptocurrencies but have no economic value. These faucet tokens allow developers and researchers to freely test smart contract deployment, transaction behavior, and gas consumption before migrating to the mainnet. Thus, test networks, such as Ethereum's test network or Polygon's test network, provide a cost-free and test environment for deploying smart contracts, executing transactions, validating system functionalities, identifying bugs and weaknesses, and optimizing solutions before deploying actual blockchain or *DApps* to the mainnet [88].

For the experimental evaluation of the proposed approach, all transactions are submitted to the Ethereum test networks (*Sepolia, Goerli*) using the *Ethereum Faucet* [89]. Faucet tokens are collected by visiting designated faucet websites and sharing posts on social media platforms like *X* (formerly Twitter) and Facebook. The use of the Faucet cryptocurrency enables us to deploy and test smart contracts, execute transactions, and evaluate other functionalities without incurring any monetary expenses.

2.2 Healthcare Policy Compliance Requirements

Healthcare policies vary significantly across countries, influenced by political systems, economic conditions, cultural values, and historical developments. In India, the regulatory framework includes the *Personal Data Protection Bill* and the *Information Technology Act*. European countries adhere to the *General Data Protection Regulation*; Australia utilizes *My Health Record (MHR)*; the

USA follows HIPAA; and Canada abides by the *Personal Information Protection and Electronic Documents Act (PIPEDA)*. The U.S. government adheres to both domestic and international laws to maintain the integrity and confidentiality of data. The *Emergency Medical Treatment and Labor Act* is a federal law enacted in 1986 that requires hospitals to provide emergency medical treatment to individuals regardless of their ability to pay or their insurance status. It prohibits patient dumping, i.e., the refusal of care or the transfer of patients with unstable medical conditions.

Children's Health Insurance Program (CHIP) is a joint federal and state program that provides health coverage to children in low-income families who do not qualify for Medicaid but cannot afford private insurance. In 2016, the *21st Century Cures Act* was enacted as a federal law to expedite advances in health research. It addresses both privacy concerns and the law, which also prohibit information blocking, in which organizations engage in activities that hinder or prevent access to electronic health information. Additionally, the 21st Century Cures Act introduces provisions enabling the compassionate sharing of mental health and substance abuse treatment details with family members and caregivers. Violations of this prohibition can result in fines of up to 1 million dollars per instance.

Apart from international and federal policies, health care must even abide by state laws to address the privacy and security of medical records, like *California Confidentiality of Medical Information Act (CMIA)*, *Colorado Medical Records Privacy Act (CMRPA)*, Arizona *Health Information Exchange (HIE)*, etc. Local and city governments also play a supportive role in healthcare through public health department regulations and community health initiatives, which supplement the regulatory framework established by federal and state laws. The HIPAA policy also allows the organization to develop its own policies and rules to ensure data security. The Facility Access Control and Workstation Security rules of the physical safeguard security policy are established and managed in accordance with organizational laws. We are implementing HIPAA and HITECH policies within our framework to ensure the protection of sensitive data.

2.2.1 HIPAA Overview

HIPAA, or the Health Insurance Portability and Accountability Act, is a U.S. federal law enacted in 1996 that establishes standards and safeguards for the protection of sensitive patient health information, known as protected health information [90]. Ensuring adherence to government policies outlined in HIPAA is a crucial aspect when constructing any information security framework. This is particularly important because patients value the confidentiality of their healthcare records. Patients who do not trust the healthcare framework may withhold essential information from healthcare providers. The HIPAA policy is divided into four rules as shown in Figure 2.1: *(i) privacy rule, (ii) security rule, (iii) omnibus rule,* and *(iv) breach notification rule.*

The *HIPAA Privacy Rule* outlines the standards for protecting individuals' medical records and other personal health information, known as protected health information, held by covered entities and their business associates. The Privacy Rule sets forth rules and procedures that covered entities must follow to ensure the confidentiality and privacy of PHI. It establishes individuals' rights to access their health information, request corrections, and control its disclosure.

The *HIPAA Security Rule* is a set of regulations established under the *Health Insurance Portability and Accountability Act (HIPAA)* to protect electronic protected health information (ePHI). It outlines standards and safeguards that covered entities, such as healthcare providers and health plans, must follow to secure ePHI. The rule encompasses administrative, physical, and technical safeguards and requires entities to implement policies, procedures, and technologies to safeguard the confidentiality, integrity, and availability of electronic health information.

The *HIPAA Omnibus Rule* introduced significant modifications to strengthen the HIPAA policy. It expanded liability by making business associates directly accountable for compliance with specific HIPAA provisions, particularly the Security Rule. The rule heightened breach notification requirements, specifying what constitutes a reportable breach and establishing procedures for notifying affected parties. Additionally, it incorporated amendments related to the *Genetic Information Nondiscrimination Act (GINA)*, further safeguarding genetic information from misuse in health plans.

The *HIPAA Breach Notification Rule* requires covered entities and their business associates to notify affected individuals, the *U.S. Department of Health and Human Services (HHS)*, and, in some instances, the media of a breach of protected health information. A breach is the unauthorized acquisition, access, use, or disclosure of PHI that compromises its security or privacy. Covered entities must conduct a risk assessment to determine the probability of compromise. If the risk is low, breach notification requirements may be waived. Non-compliance with breach notification rules can result in financial penalties.

The U.S. government also enacted the HITECH Act (*Health Information Technology for Economic and Clinical Health Act*), which introduced several policies and provisions to promote the adoption and meaningful use of health information technology, particularly electronic health records (EHRs). HITECH significantly amended HIPAA by revising its rules and strengthening enforcement mechanisms. It introduced new requirements and increased penalties for non-compliance. HITECH expanded the liability of business associates by holding them directly accountable for compliance with specific HIPAA provisions. Business associates are now subject to many of the same rules and penalties as covered entities. HITECH also significantly increased the penalties for HIPAA violations, providing a tiered structure based on the level of negligence. The maximum annual penalty for a single contravention increased substantially. HITECH also introduced additional mandates on data breach notifications. Under the HITECH Breach Notification Rule, in the event of a data breach, HIPAA-covered entities must notify the individuals affected. Furthermore, notification must be provided to both the Secretary of Health and Human Services and the media if the breach involves more than 500 individuals.

2.2.2 HIPAA Regulated Organizations

The regulations outlined in the HIPAA Rules pertain to both covered entities and business associates. Entities, organizations, and agencies that fall within the scope of a covered entity, as defined by HIPAA, are obligated to adhere to the Rules. These regulations mandate safeguarding the privacy and security of health information and grant individuals specific rights regarding their

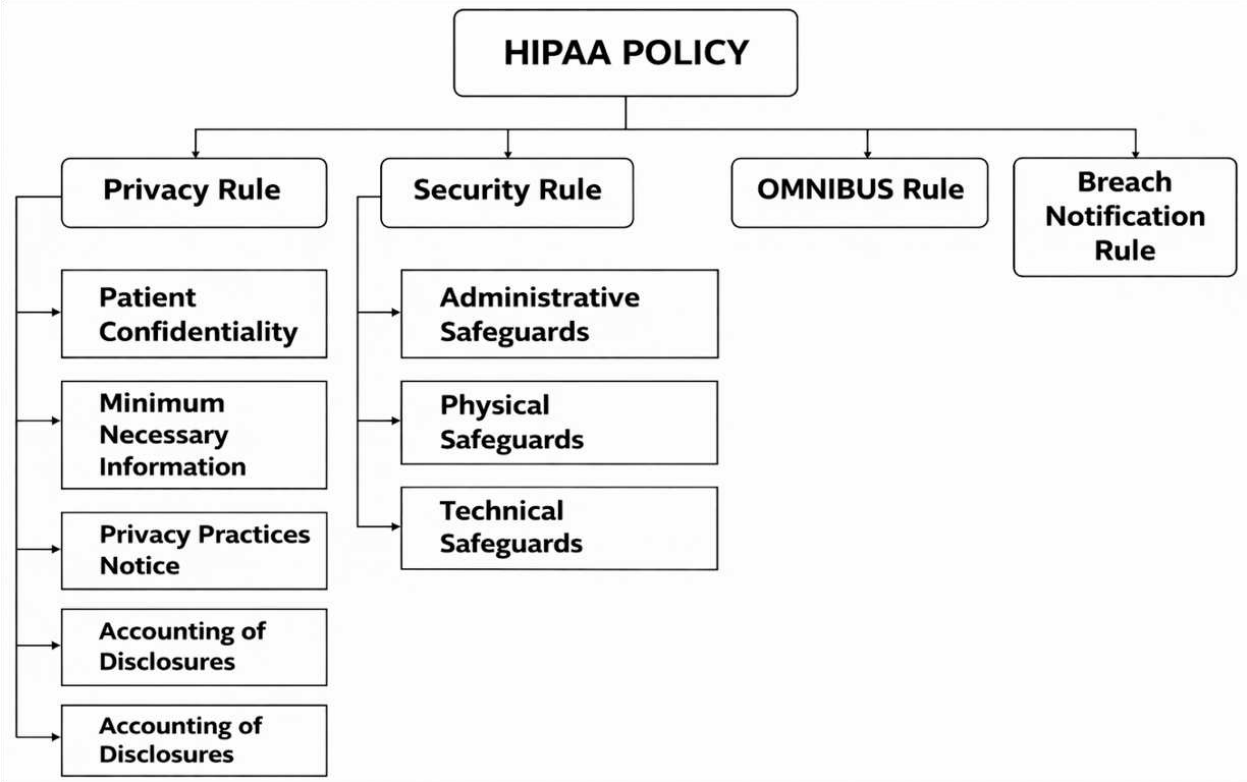

Figure 2.1: Types of HIPAA Rules [2]

health data. In cases where a covered entity engages a business associate for healthcare-related activities, a formal contract or arrangement must be in place. Figure 2.2 shows the HIPAA-regulated organizations.

The HIPAA regulation typically requires covered entities and their business associates to establish contracts, known as business associate agreements (BAAs), to ensure the proper protection of protected health information. These contracts also define and, as necessary, restrict the acceptable uses and disclosures of protected health information by the business associate. According to the updated regulations, business associates are now directly accountable under HIPAA and subject to enforcement actions in the same manner as covered entities.

A formal agreement between a covered entity and a business associate must include an outline of the allowed and mandatory uses and disclosures of protected health information by the business associate, and stipulate that the business associate will not utilize or disclose the information beyond what is permitted by the contract or as mandated by law, and mandate the implementation of

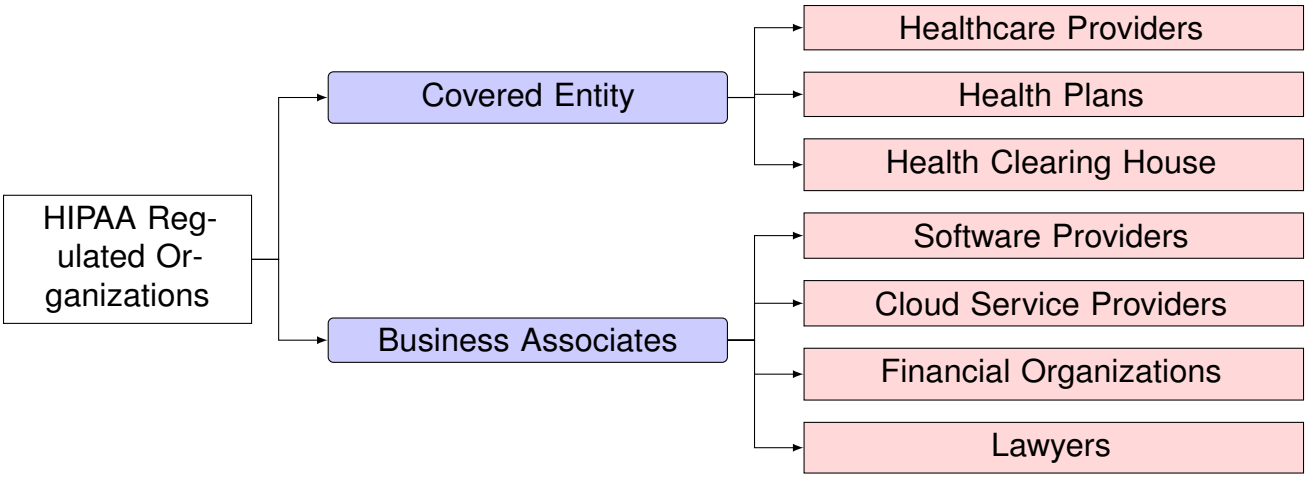

Figure 2.2: HIPAA Regulated Organizations [2]

adequate safeguards by the business associate to prevent unauthorized use or disclosure, including adherence to the requirements of the HIPAA Security Rule concerning electronic protected health information, etc.

2.2.3 HIPAA Rules Incorporation in Proposed Framework

The proposed framework incorporates the privacy and security rules of the HIPAA/HITECH policy to protect patient information, adheres to the rules of the regulated organization, and doesn't implement the omnibus and breach notification rules as shown in Figure 2.1. The framework adheres to the following privacy and security guidelines:

- The patient grants consent for the treatment team members to access their information. The patient also controls the extent of information shared with specific team members and the type of access, such as read, write, and update permissions. As a result, the privacy and HIPAA policies related to authorization and disclosure accounting are in compliance.
- The framework enables patients to grant access rights to their emergency contacts, allowing them to access information. It also allows restricted access to specific information, ensuring compliance with the confidential communications policy.
- Due to frequent updates to HIPAA policies, those managing patient information must stay informed of these changes. The healthcare prototype manages workforce training and security

in accordance with HIPAA policies by implementing an expiration date for HIPAA training, ensuring compliance with relevant regulations. Authorities cannot access patient health records once their HIPAA training expires, and access is reinstated only after they complete the required training renewal.

- Password authentication and information consent mechanisms are employed within the framework to enforce HIPAA policies related to access and audit control. Access to information is restricted to authorized entities, and the system generates regular audit logs that notify relevant authorities.
- The workstation security physical safeguard security policy is enforced as the framework is a licensed application and will be installed on secured workstations. Restricting the application license to a select number of fully authenticated systems installed on workstations helps minimize the risk of data breaches.

The developed framework enhances patient transparency by informing patients who access their information. This instills confidence in patients, assuring them that their data is secure and aligns with most privacy and security policies outlined in HIPAA.

2.3 Consent-Based Healthcare Data Access

Table 2.4 compares significant global data protection and privacy laws that emphasize consent as a legal basis for processing personal or healthcare data. This table lists key frameworks, including *HIPAA, GDPR, PIPEDA, PDPA*, and *CCPA*, along with relevant articles or sections that highlight consent requirements.

Summary: Across jurisdictions, modern data protection frameworks share a common principle: **individual consent is the cornerstone of lawful data processing**, especially for sensitive categories such as healthcare data. Regulations such as the GDPR (EU), HIPAA (USA), PDPA (Singapore), and the DPDP Act (India) all emphasize that personal health information may not be collected, used, or disclosed without the explicit, informed, and voluntary consent of the data subject. This

convergence reflects a global commitment to respecting individual autonomy, transparency, and accountability in the handling of data.

In the context of healthcare, consent serves not only as a legal requirement but also as a mechanism of **trust and empowerment**, allowing patients to control who accesses their medical information, under what conditions, and for what purposes. However, enforcing and auditing consent across multiple healthcare entities remains a persistent challenge in centralized systems. This challenge motivates the exploration of **blockchain-based consent management frameworks**, where smart contracts can automate, verify, and record consent transactions immutably. Such systems can ensure that every access or processing event is cryptographically verifiable, privacy-preserving, and fully compliant with international data protection principles and regulations.

Table 2.4: Global Data Protection and Privacy Laws Emphasizing Patient or Data Subject Consent in Healthcare Data Processing

Law / Regulation	Jurisdiction / Region	Consent-Related Provisions (Key Articles / Sections)	Focus on Patient or Data Subject Consent
HIPAA (Health Insurance Portability and Accountability Act, 1996)	United States	Privacy Rule, 45 CFR §164.506–508: Requires patient authorization before using or disclosing Protected Health Information (PHI) for non-treatment, payment, or healthcare operations.	Explicit written consent is required for sharing PHI beyond primary care purposes.
GDPR (General Data Protection Regulation, 2016/679)	European Union	Article 6(1)(a): Consent as a lawful basis for data processing; Article 7: Conditions for valid consent; Article 9(2)(a): Explicit consent required for processing special categories of data, including health data.	Informed, specific, freely given, and revocable consent is mandatory for processing health-related data.
UK Data Protection Act, 2018	United Kingdom	Section 8 and Schedule 1, Part 1(1): Consent as a lawful basis under GDPR principles; special provisions for health and social care data.	Patient consent required under GDPR-equivalent standards for sensitive health data.
PIPEDA (Personal Information Protection and Electronic Documents Act)	Canada	Schedule 1, Principle 3: Requires knowledge and consent of the individual for the collection, use, or disclosure of personal information.	Implied or express consent, depending on data sensitivity; express consent required for medical data.
PDPA (Personal Data Protection Act)	Singapore	Section 13–15: Requires consent before collecting, using, or disclosing personal data; Section 17: Consent must be informed and specific.	Explicit consent is required before processing health-related or sensitive data.
CCPA (California Consumer Privacy Act, 2018)	California, USA	Section 1798.120: Grants consumers the right to opt-out of data sale and mandates disclosure about data usage; emphasizes affirmative authorization for sensitive data use.	Patient consent is implicit in opt-in or opt-out provisions for sensitive personal data.
LGPD (Lei Geral de Proteção de Dados, 2018)	Brazil	Article 7(I): Requires consent for data processing; Article 11(I): Explicit consent required for sensitive personal data, including health data.	Informed consent is essential for processing or sharing patient information.
POPIA (Protection of Personal Information Act, 2013)	South Africa	Section 11(1)(a): Processing permitted if the data subject consents; Section 26: Prohibits processing of special personal information, including health data, without explicit consent.	Explicit patient consent is required for processing medical or health records.
DPDP Act (Digital Personal Data Protection Act, 2023)	India	Section 4(1) and 6(1): Personal data may be processed only for lawful purposes with consent of the Data Principal; Section 7: Specifies requirements for valid consent.	consent must be free, specific, informed, and capable of withdrawal—applicable to digital health data.
Privacy Act, 1988 (as amended)	Australia	Australian Privacy Principle (APP) 3 and APP 6: Consent required for collection and disclosure of sensitive information, including health records. Express consent is required for the handling of patient data, unless otherwise exempted by law.	

Chapter 3

Related Works

This chapter reviews the existing literature relevant to the development of the proposed healthcare policy compliance framework. It examines prior studies, models, and technologies that have addressed various aspects of policy compliance in healthcare data management and access control. To provide a structured understanding of the state of the art, the related works are categorized into five thematic areas: (i) healthcare policy compliance, (ii) treatment team PHI access policy compliance, (iii) PHI sharing beyond treatment team policy compliance, (iv) emergency PHI access policy compliance, and (v) policy compliance review. They are discussed in the following sections. This categorization highlights how current solutions have evolved and where critical gaps remain, thereby motivating the proposed blockchain-based compliance framework presented in the subsequent chapters.

3.1 Healthcare Policy Compliance

In the burgeoning landscape of electronic health record (EHR) management, researchers have proposed innovative blockchain-based frameworks to address challenges such as secure storage, scalability, and interoperability. Shuaib et al. [91] discussed the physical, technical, and administrative needs of healthcare compliance with the Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR) and sketched potential blockchain-based EHR systems and potential areas for improvement.

Piao et al. [92] proposed a blockchain-based GDPR compliance data sharing scheme, aiming to promote compliance with regulations and provide a tool for interaction between users and service providers to achieve data security sharing. This work focuses primarily on user authentication and private data sharing, with less emphasis on provenance. Wu et al. [93] propose an architecture for a public blockchain-based ledger that can provide policy compliance activities.

Haque et al. [94] presented an architecture for a GDPR-compliant COVID-19 vaccination passport (VacciFi) that stores vaccination data in off-chain storage. They use a permissioned blockchain to enable participating entries to track activities more efficiently. The following four design principles are identified for GDPR compliance: *(i) access to data should be traceable; (ii) data subjects' consent should be collected using a smart contract, (iii) blockchain data needs to be deleted or modified upon request,* and *(iv) data controllers and processors must be identified*. This work provides insights into the design requirements of our proposed system.

Hasselgren et al. [95] discussed the GDPR articles on compliance matters and their implications for the healthcare industry. They also perform comparative analyses of four blockchain-based healthcare application systems, MedRec [96], EMRShare [97], FHIRChain [98], and VerifyMed [99], to demonstrate compliance with GDPR. There are several recommendations for blockchain researchers and developers when designing blockchain-based healthcare applications to ensure compliance with relevant policies and regulations.

Shahnaz et al. [100] propose a blockchain-based framework for EHR and provide secure storage of electronic records with granular access rules for users. It also addresses the scalability problem of blockchain by utilizing off-chain storage. This paper focuses exclusively on ensuring secure information storage, without exploring the implementation of information consent or other government-mandated privacy standards.

Mayer et al. [101] analyze that Blockchain technology can provide secure, tamper-resistant storage of medical records, ensuring data integrity and authenticity, and also suggest that the blockchain directory model and the chain structure can support the continuous growth of medical records. The blockchain should implement standards such as OpenEHR, HL7 FHIR, HIPAA, GDPR, IHE, ISO, SNOMED, DICOM, HIE, and PII to facilitate interoperability and ensure the uniformity of healthcare information for all stakeholders.

Wang et al. [102] develop a combined attribute-based/identity-based encryption and signature blockchain mechanism to minimize the utilization of different cryptographic systems for different security requirements in the EHR. This system ensures the integrity and traceability of medical data.

This work focuses solely on encrypting patient signatures and identities, without addressing the safeguarding of other sensitive patient information, such as health history and insurance details.

Existing studies suggest the potential effectiveness of blockchain for storing compliance-related information for provenance; most focus on regulatory policy compliance and do not address individual policies within patient consent and organizational access control. We are deploying the patient policy smart contract in the blockchain framework to uphold the integrity of the patient's records.

3.2 Treatment Team PHI Access Policy Compliance

Several proposals have been made to adopt blockchain technology in healthcare and e-health systems. So far, this research has focused on how blockchain can protect medical information and facilitate the storage and sharing of medical data, analytics, and informed consent systems for clinical or research experiments. Research on informed consent for clinical diagnosis and treatment has received some attention [103]. To our knowledge, ours is the first work to employ blockchain and smart contracts for clinical treatment-informed consent management and enforcement.

Azaria et al. [96] introduce *MedRec*, a healthcare data management system built on blockchain technology to improve access and permissions for electronic medical records. The proposed model addresses four significant challenges: (i) fragmented access to medical data, (ii) a lack of system interoperability, (iii) limited patient control over their information, and (iv) the need for enhanced data quality and quantity for research purposes. MedRec provides patients with a comprehensive, immutable record of their medical history, facilitating easy retrieval of information from healthcare providers and treatment facilities. By aggregating and encoding references to various types of medical data on a blockchain ledger, MedRec establishes a transparent and accessible historical trail for medical information. Our proposed approach also manages permission and data access using blockchain, similar to the MedRec architecture. It ensures transparency in patient-record access by obtaining patient consent before sharing the records with team members.

Xia et al. [104] propose a blockchain-based data-sharing framework that addresses access-control challenges for sensitive data stored in the cloud by leveraging the blockchain's immutability and built-in autonomy. Yue et al. [105] propose a blockchain-powered application, *Healthcare Data Gateway*, that enables patients to store, manage, and securely distribute their healthcare information. The system facilitates the secure processing of healthcare data by untrusted parties through secure multi-party computation, thereby ensuring patient privacy. Zyskind et al. [106] introduce the use of blockchain technology for managing access control and securely storing data, with encryption used to protect the data stored on servers. Fan et al. [107] propose MedBlock, a blockchain-driven information management system to streamline access and retrieval of electronic medical records (EMR). It protects users' privacy through custom access-control protocols and encryption while sharing data.

Tith et al. [108] propose an electronic consent (*e-consent*) management model that uses *Hyperledger Fabric* blockchain and a purpose-based access control framework. This system records all patient data, consents, and metadata related to data access on the blockchain, thereby making it accessible to participating organizations. A specific chaincode executes the business logic for handling patient consent, allowing patients to initiate, modify, or revoke their consent directly on the blockchain. The proposed model can be used to donate data to biobanks for research purposes, in addition to sharing patient data. However, the Hyperledger blockchain is a permissioned network in which participants are limited to organizations, potentially reducing transparency to the broader public. To address transparency issues, our proposed approach uses the public Ethereum blockchain, allowing participants with stakes to join and maintain the ledger, thereby providing immutable information to untrusted network participants. Most importantly, Ethereum's public consensus mechanism adds more transparency than a permissioned network. In addition, Ethereum smart contracts are the most widely used, with numerous projects currently under development and refinement.

Cunningham et al. [109] propose *Non-Fungible Tokens (NFTs)* as the mechanism for recording and transmitting records of patients' consent for medical data use. The proposed model enables

individuals to record signed consent documents, thereby permitting *Data Consumers* to request health information from *Data Providers* in accordance with the subjects' consent. Nevertheless, the application of NFTs to track data provenance in compliance with regulatory standards such as *HIPAA/GDPR* remains under exploration.

Albalwy et al. [110] introduce a blockchain-based consent management system, *ConsentChain*, facilitating clinical genomic data exchange. Utilizing the *Ethereum* blockchain, it employs smart contracts to represent the roles and permissions of patients (who grant or revoke access to their data), data creators (who gather and maintain patient information), and data requesters (who seek access to this information). While this work primarily focuses on facilitating the exchange of genomic data among clinicians, researchers, and bioinformaticians, clinical treatment presents distinct challenges for consent management compared to genomic data sharing. The treatment process involves various user actions, such as reading, writing, and modifying data, with access privileges assigned based on users' roles. Recognizing the diverse requirements of clinical treatment processes, this work proposes a consent management framework to address complex permission assignments across roles, including treatment team members, insurance agents, external doctors, and pharmacists.

Numerous proposals have been made for integrating blockchain technology into healthcare and e-health systems. Existing research primarily focuses on using blockchain to safeguard medical records and facilitate the storage and sharing of medical data, analytics, and systems for managing informed consent in clinical or research settings. There has been some research dedicated to informed consent in diagnosis and treatment as well [103]. To the best of our knowledge, our study is the first to leverage blockchain and smart contracts to manage and enforce informed consent in clinical diagnosis and treatment (Section 4.4).

3.3 PHI Sharing Beyond Treatment Team Policy Compliance

Blockchain technology has increasingly been adopted in healthcare for various services, particularly for sharing protected health information among healthcare providers, patients, and other

stakeholders. Blockchain facilitates a more efficient, transparent, and patient-centered delivery of healthcare services, making it an essential component in modern healthcare infrastructure.

Fan et al. [107] propose a blockchain-based secure system, MedBlock, to share electronic medical records among authorized users. It provides security and privacy with access control protocols and encryption technology while sharing patient healthcare data. Shah et al. [111] propose a medical data management framework to facilitate data sharing. It gives patients full control over access to their medical data. It also ensures that patients know who can access their data and how it is used.

Zhuang et al. [112] propose a blockchain-based patient-centric health information-sharing mechanism that protects data security and privacy, ensures data provenance, and provides patients with complete control over their health data. However, consent structure and compliance requirements are not addressed, which are crucial for giving patients confidence in how their consent is executed and how data is protected.

Alhajri et al. [113] examine the importance of implementing legal frameworks to safeguard privacy in fitness apps. By examining how various fitness apps handle consent and privacy policies, the research highlighted the crucial role of consent under the GDPR. The authors proposed adopting blockchain technology to govern user consent for the sharing, collection, and processing of fitness data, ensuring a process centered on human needs and compliant with legal standards. Nonetheless, the study did not present a technical architecture for its blockchain-based proposal.

Amofa et al. [114] develop a blockchain-based personal health data-sharing framework that incorporates an underlying mechanism to monitor and enforce acceptable-use policies associated with patient data. Generated policies are consulted by smart contracts to determine when the intended data can be shared or otherwise. All entities cooperate to protect patient health records from unauthorized access and computations.

Balistri et al. [115] design the *BlockHealth* solution for sharing health data with tamper-proofing and protection guarantees. They store the patient's healthcare data in a private database, and the hash of that data is stored on the blockchain to ensure data integrity. Shen et al. [116] propose

MedChain, a blockchain-based health data-sharing approach in which data streams are continuously generated by sensors and other monitoring devices across various patients' bodies. The collected data are shared with laboratories and health organizations for diagnosis, advanced treatment, and further research.

As mentioned in this section, the papers summarize the applications and benefits of blockchain for healthcare data sharing and essential services. However, they failed to address the security and privacy requirements mandated by various laws and regulatory agencies, such as HIPAA and GDPR. The primary requirements for sharing health records are patient consent and appropriate protection, such as encryption. In addition, it is crucial to maintain audit logs and verify that these activities do not violate any policies. This work proposes sharing informed consent as the smart contract for authorization with provenance and compliance-checking mechanisms (Section 4.5).

3.4 Emergency PHI Access Policy Compliance

Yang et al. [117] introduced a novel lightweight break-glass access control (*LiBAC*) system designed for the Healthcare IoT, enhancing the security and accessibility of medical data. The system employs a dual-access method: attribute-based access for routine use and break-glass access for emergencies, ensuring timely access to patient information by authorized personnel. The *LiBAC* is rigorously proven secure in the standard model, with a formal proof substantiating its resilience against potential cyber threats. Despite its efficiencies, the model relies heavily on a predefined set of emergency contacts, which may limit its effectiveness in unexpected situations where those contacts are unavailable or when new, unforeseen stakeholders need access.

Loos et al. [118] investigated the tension between emergency accessibility and security in medical devices, highlighting the lack of comprehensive break-glass systems tailored to these devices. They categorized break-glass mechanisms into patient records and medical devices. The authors explore various emergency access solutions, including proximity-based access, biometric authentication, UV tattoos, RFID chips, and passive radiopaque markers. Despite proposing innovative mechanisms, they highlight several challenges, including the need to balance usability

and security, patient acceptance, and the lack of standardization. The paper urges further research into unified security protocols that reconcile emergency access needs with robust patient data protection.

Aski et al. [119] proposed integrating break-glass mechanisms with attribute-based access control (*ABAC*) to address emergencies in healthcare *IoT* systems. In addition to authorizing users in everyday situations, they introduced a break-glass mechanism that allows emergency handlers (*ESH*) to respond to emergencies. The *ESH* bypasses standard authentication and swiftly accesses critical patient data when immediate medical action is required. Security measures include data encryption and key management, with *ESH* verification through pre-distributed passwords to prevent unauthorized access and misuse. Experimental analysis indicates the scheme's efficiency compared to existing access control systems.

Schefer-Wenzl et al. [120] conduct a survey to investigate delegation and break-glass-based emergency access control, where standard access policies are insufficient. In delegation models, a user can transfer access rights or roles to another, discussing role-based and permission-based approaches while considering constraints such as separation of duty (*SoD*) and binding of duty (*BoD*). The break-glass models are designed for emergencies, enabling temporary bypass of standard access controls with actions logged to prevent misuse. Analyzing 329 articles and detailing 35 key approaches, the authors compare models with respect to policy enforcement, support for entailment constraints, and integration with business processes.

Van Bael et al. [121] described a new access control system that uses *IoT* sensors to collect contextual data, thereby making break-glass mechanisms more flexible in emergencies. It includes non-repudiation features by logging all actions during a break-glass event, thereby ensuring accountability through evidence such as biometric data or badge scans. A fail-safe mechanism is also incorporated to cancel emergency access if activated erroneously. However, the prototype demonstrates that it is possible and achieves reasonable response times. The proposed approach relies on the availability and dependability of *IoT* sensors. It is vulnerable if the integrity of contextual data is compromised, and establishing comprehensive access policies for all emergencies is challenging.

The papers in this section summarize the application of emergency access control mechanisms, such as the break-glass protocol. However, they failed to address the security and privacy compliance requirements mandated by various laws and regulatory agencies, such as *HIPAA* and *GDPR*. This work proposes a policy compliance framework for emergency PHI access, ensuring that applicable security and privacy policies are followed while accessing PHI to save a patient's life in critical moments (Section 4.6).

3.5 Policy Compliance Review

García-Berná et al. [122] 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*. 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. [123] address the complex challenge of sharing electronic health records (EHRs) across healthcare organizations while adhering to varying regulatory and business requirements in their work on compliance-aware cross-organizational medical record sharing. Their proposed solution, CHINO, enables healthcare providers to define and enforce their security and compliance requirements 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. [124] develop a sophisticated framework to securely manage the exchange of extensive health data while ensuring strict compliance with health regulations such as HIPAA. Amidst the challenges of the COVID-19 pandemic, their framework utilizes semantic web technolo-

gies 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 to a simulated scenario using over 1,000,000 synthetic contact-tracing records from the CDC.

Koreff et al. [125] critically examine data analytics in healthcare fraud audits, focusing on how these tools influence power dynamics and potentially abuse authority. Through qualitative analysis of interviews and documents, the study revealed that algorithmic decision-making can justify harsh measures against healthcare providers based on potentially inaccurate interpretations of data. 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 the use of data analytics within regulatory frameworks, and the ethical considerations in balancing technological efficiency with fair governance.

Mohammed et al. [126] propose a detailed analysis of blockchain's dual scalability and regulatory compliance challenges through a literature review. They explored more advanced options, including sharding, which partitions the blockchain into parallel-processing components to expedite transactions, and layer-2 methods such as rollups and the Lightning Network, which remove transactions from the main chain to reduce latency and facilitate scaling.

While these works offer hope that blockchains could be a viable solution for storing compliance-related provenance information, most focus on regulatory compliance. Compliance with individual policies expressed in a patient-provider agreement, patient consent management, and organizational access control policies is outside the scope of these works.

Chapter 4

Policy Enforcement

Policy enforcement is the process of applying the required or applicable set of policies through a mechanism that evaluates an access request against them. In addition to the relevant policies, it also considers other components, such as subject and object attributes, environmental conditions, and other factors mandated by business requirements. Applicable policies are adequately selected, evaluated, implemented, tested, and deployed to be available for execution. These policies are selected according to business requirements, legal jurisdictions, regulatory mandates, contractual obligations, and other factors. The entity that does this part is known as the policy implementer.

Proper mechanisms must be in place to execute, evaluate, and enforce the policies mentioned above that are relevant to any authorization or access request. A decision, granted or rejected, is made based on the applicable policies and attributes of the subject, the requested object, and the operation. The entity doing this part is the policy enforcer or authorization module. Policy implementers and enforcers must be separate entities to ensure the *Separation of Duties* and avoid potential *Conflicts of Interest*.

Healthcare policy enforcement is the process of applying and implementing healthcare privacy, security, and consent management rules through defined procedures, governance mechanisms, and technical safeguards to ensure that *protected health information* is handled lawfully, ethically, and appropriately. In the context of HIPAA and related healthcare regulations, policy enforcement ensures that PHI access, use, and sharing decisions are consistently governed by patient consent, clinical role, treatment purpose, organizational policy, contextual conditions, and legal requirements. Its objective is not only regulatory compliance but also the preservation of patient trust, confidentiality, accountability, and continuity of care. Therefore, robust enforcement mechanisms are essential to ensure that covered entities, business associates, healthcare professionals, and affiliated organizations remain accountable for safeguarding PHI and for enforcing privacy and security requirements throughout treatment, information sharing, and emergency response workflows.

4.1 PHI Access Classification

The primary regulatory bodies and data protection laws, such as HIPAA, GDPR, and others, mandate patient consent-based access to and sharing of protected health information. Table 4.1 shows ten (10) types of PHI, considered for each patient, with PHI ID, name, description, and potential creators. There are three major classifications of PHI access by the healthcare stakeholders. They are (i) treatment team access for regular treatment and operations, (ii) sharing beyond the treatment team for better diagnosis, consultation, research, and marketing, and (iii) emergency access when the patient is conscious or severely injured. Figure 4.1 shows them with the definition.

4.1.1 Treatment Team Access

Authorized treatment team members access healthcare data within healthcare systems to provide required medical care and services and perform healthcare operations. This includes doctors, nurses, and specialists collaborating to make informed decisions about diagnosis, treatment plans, and ongoing care. In addition to direct patient care, health records support essential business operations, including billing, insurance claims processing, scheduling, and quality assurance. Ensuring seamless access for healthcare providers while maintaining data privacy and security is critical. Robust access controls and encryption protocols are essential for safeguarding sensitive information against unauthorized access or data breaches. The authors propose a consent-based approach to PHI access compliance for this group [13].

4.1.2 Sharing Beyond Treatment Team

Healthcare data are often shared with parties beyond direct care providers to improve patient outcomes and drive broader healthcare initiatives. Consultations with specialists, for instance, allow for more accurate diagnoses and more effective treatment plans. Healthcare data is also leveraged in research to identify trends, develop new treatments, and improve overall healthcare quality. Furthermore, anonymized patient information may be used for marketing purposes, such as promoting relevant health services. However, these practices require strict adherence to data

Table 4.1: Sample Patient Protected Health Information (PHI) Structure [3, 4, 10, 11]

PHI ID	PHI Name	PHI Description	PHI Creator
PHI-1001	Demographic Information	Basic personal information like name, date of birth, gender, contact	Patient, Support Staff
PHI-1002	Previous Medical History	Old medical records from another hospitals and providers	Patient, Support Staff
PHI-1003	Immunizations, Vaccinations	Immunization records that are administered over time	Patient, Pathology Lab Technician
PHI-1004	Allergies	Various allergies sources, triggering condition, remediation	Patient, Support Staff, Path Lab Tech
PHI-1005	Visit Notes	Physiological data, advice, follow-up, visit details	Doctor, Nurse
PHI-1006	Medications, Prescription	Pharmacy information, prescribed medications like name, dosage	Doctor
PHI-1007	Pathology Lab Works	Biological samples analysis like blood, tissue, other substances	Pathology Lab Technician
PHI-1008	Radiology Lab Works	Imaging results such as X-rays, CT, MRI, Ultrasound, PET scans	Radiology Lab Technician
PHI-1009	Billing, Insurance	Bank account, credit/debit card, and insurance policy information	Patient, Support Staff, Billing Officer
PHI-1010	Payer Transactions	Bills of doctor visit, lab works, and medications	Billing Officers, Insurance Agent

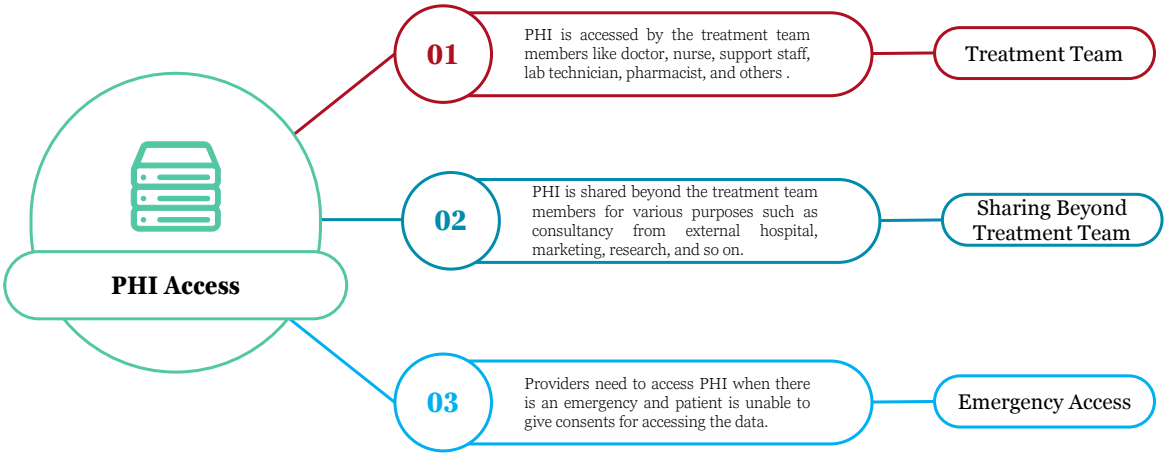

Figure 4.1: Protected Health Information (PHI) Access Classification [4]

protection regulations to maintain patient privacy and obtain informed consent. The authors of [3] proposed a policy-compliance assurance framework that relies on patient consent to share PHI.

4.1.3 Emergency Access

In life-and-death situations, such as when a patient is unconscious or critically injured and admitted to the emergency room, immediate access to their healthcare information becomes crucial for treatment. In most cases, healthcare providers obtain consent from patients before accessing or sharing their medical data with other specialists. However, in emergencies, consent cannot be obtained in advance due to the unpredictable nature of the situation. Additionally, emergencies may occur far from a patient's home or primary care provider, further complicating access to their medical history. In these scenarios, obtaining consent from the injured or incapacitated patient is impossible, as they may be unconscious or unable to communicate. This creates a unique challenge for healthcare professionals, who must act swiftly to provide life-saving care.

Emergency access protocols, such as the *Break-Glass Protocol*, enable healthcare providers to temporarily bypass consent, ensuring they can access essential information while maintaining compliance with privacy regulations and audit controls. If a patient is admitted to the same hospital, who is the primary care provider? Transferring data is unnecessary, as doctors would access it through the same EHR system. Data access can be performed from the emergency room while treating the patient or in the ambulance while transferring the patient from home or the accident scene to the hospital. When a patient receives regular treatment and medical services from one hospital but is admitted to another hospital for emergency treatment. The patient's health data must be shared between the primary care provider and the current provider. Providers must satisfy additional data protection and patient privacy requirements for transferring data. We assume data is transferred from the primary provider to the emergency provider through the proper channel.

Each sort of PHI access has a different set of compliance requirements. This research integrates patient consent with the *Patient-Provider Agreement* and deploys it as smart contracts in the blockchain network. Then, the PPA is enforced while authorizing the PHI access request. Enforcement activities are captured and stored for later compliance review.

4.2 Policy Enforcement - Proposed Approach Overview

The proposed policy enforcement approach centers on capturing patient-informed consents for three critical healthcare scenarios: treatment team access, PHI sharing beyond the treatment team, and emergency PHI access. These consent requirements are formally integrated into a *Patient–Provider Agreement*, which serves as the patient-specific contractual foundation for governing PHI access and disclosure. Once the PPA is established, its relevant consent components are extracted, verified for integrity, and deployed to the blockchain as patient-associated smart contracts. In this architecture, the blockchain acts as a trusted, tamper-evident platform for storing and executing consent rules, while the smart contract functions as the authoritative source for consent validation during access-control decisions. When a PHI access or sharing request is submitted, the authorization module queries the corresponding smart contract and evaluates the returned consent

information together with applicable policies, subject and object attributes, purpose, environmental conditions, and regulatory requirements. Based on this combined evaluation, the system determines whether the requested PHI access or sharing action should be permitted or denied. This approach strengthens policy enforcement by making patient consent executable, transparent, immutable, and directly usable in real-time authorization decisions for routine treatment, inter-organizational sharing, and emergency access scenarios.

4.3 Patient-Provider Agreement (PPA)

The patient-provider agreement aims to clarify each party's responsibilities in the treatment process. The goal is to improve outcomes, lower risks, and better educate patients. A multicenter study [127] evaluated the utility of the PPA, the extent to which patients understood it, its ability to educate patients in an unbiased manner about treatment, and the feasibility of incorporating a PPA into clinical practice. Both patients and doctors believe this PPA helped them decide on a course of treatment and was fair in laying out the treatment's risks and benefits. Most patients reported that the PPA was "somewhat helpful" or "very helpful" in deciding on a course of treatment and that it was "easy to understand."

A PPA, also known as a contract, varies by organization. Healthcare organizations adjust what they need from patients and what they expect of them to align with those needs, treatments, and responsibilities. This is done based on the nature and needs of treatment and services. Additionally, the components and representation of the PPA vary across hospitals and clinics. Examples include general hospitals, emergency rooms, urgent care or walk-in clinics, dental care, cancer treatment, physiotherapy, etc. Figure 4.2 shows the major components of a PPA. Algorithm 1 illustrates the iterative process of creating a PPA with the required components. A PPA is formally composed of six (6) tuples:

$$PPA = (PC, PrC, TIC, SIC, EIC, ROC)$$

satisfying the following requirements:

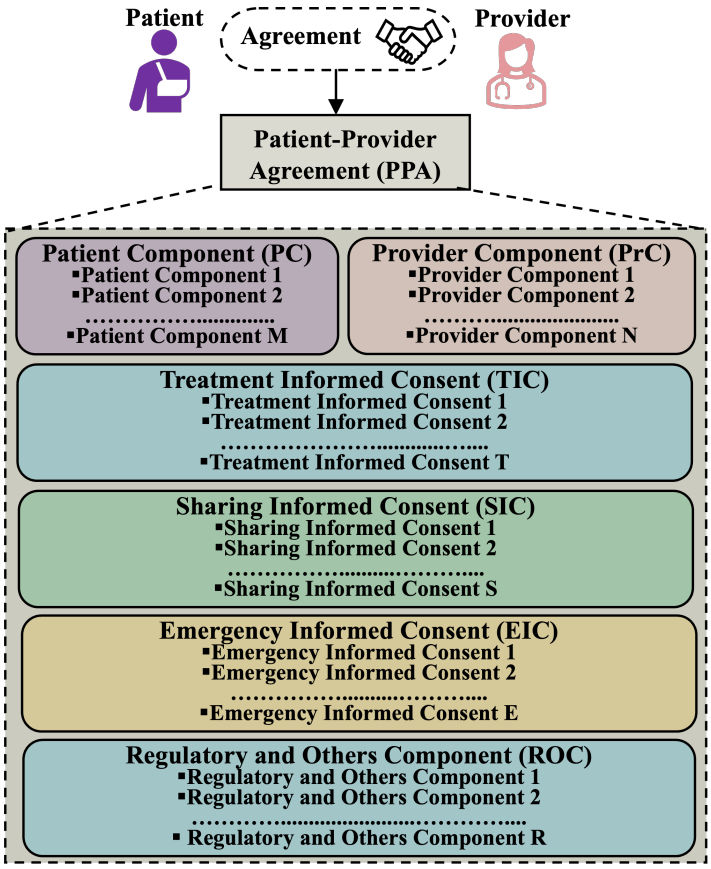

Figure 4.2: Patient-Provider Agreement (PPA) Components [2–4, 10–12]

(A) PC is a finite set of patient components containing the patient's personal information, contact information, mailing information, pharmacy information, billing and insurance information, emergency contact, and others. The patient is responsible for providing and maintaining valid, accurate, and updated information for these components.

(B) PrC is a finite set of provider components, including the treatment team, prescription, and others. The provider is responsible for creating an effective team to provide appropriate care. All aspects of treatment, insurance coverage, and billing are considered during the patient's treatment period.

(C) TIC is a finite set of treatment-informed consent components. It denotes that the patient has permitted the designated treatment team to access medical records. Treatment team members include doctors, nurses, support staff, lab technicians, billing officers, emergency contact persons, and others assigned by the authority.

Algorithm 1: Patient-Provider Agreement (PPA) Formation [2, 3, 11, 13]

Input : (i) PC, (ii) PrC, (iii) TIC, (iv) SIC, (v) EIC, (vi) ROC, (vii) $\mathbb{R}_{PPA}$, (viii) $\mathbb{BN}_{SC}$

/* $\mathbb{R}_{PPA}$: secured PPA repository, BN_{SC}: blockchain network smart contract */

Result : A formal PPA

Input Parameters Initialization

$PPA_i \leftarrow \{PC_i, PrC_i, TIC_i, SIC_i, EIC_i, ROC_i\}$ where i is patient identity

(i) $PC \leftarrow \{PC_1, PC_2, PC_3, PC_4, PC_5, PC_6 ..... PC_M\}$

(ii) $PrC \leftarrow \{PrC_1, PrC_2, PrC_3, PrC_4, PrC_5, PrC_6 ..... PrC_N\}$

(iii) $TIC \leftarrow \{TIC_1, TIC_2, TIC_3, TIC_4, TIC_5, TIC_6 ..... TIC_T\}$

(iv) $SIC \leftarrow \{SIC_1, SIC_2, SIC_3, SIC_4, SIC_5, SIC_6 ..... SIC_S\}$

(v) $SIC \leftarrow \{EIC_1, EIC_2, EIC_3, EIC_4, EIC_5, EIC_6 ..... EIC_S\}$

(vi) $ROC \leftarrow \{ROC_1, ROC_2, ROC_3, ROC_4, ROC_5, ROC_6 ... ROC_R\}$

PPA Components Integrity Calculation

/* $\mathbb{H}(\partial)$ calculates hash of ∂ */

(a) $\mathbb{H}_{PC} \leftarrow \mathbb{H}(PC_1, PC_2, PC_3, PC_4, PC_5, PC_6 ..... PC_M)$

(b) $\mathbb{H}_{PrC} \leftarrow \mathbb{H}(PrC_1, PrC_2, PrC_3, PrC_4, PrC_5, PrC_6 ..... PrC_N)$

(c) $\mathbb{H}_{TIC} \leftarrow \mathbb{H}(TIC_1, TIC_2, TIC_3, TIC_4, TIC_5, TIC_6 ..... TIC_T)$

(d) $\mathbb{H}_{SIC} \leftarrow \mathbb{H}(SIC_1, SIC_2, SIC_3, SIC_4, SIC_5, SIC_6 ..... SIC_S)$

(e) $\mathbb{H}_{EIC} \leftarrow \mathbb{H}(EIC_1, EIC_2, EIC_3, EIC_4, EIC_5, EIC_6 ..... EIC_S)$

(f) $\mathbb{H}_{ROC} \leftarrow \mathbb{H}(ROC_1, ROC_2, ROC_3, ROC_4, ROC_5, ROC_6 .. ROC_R)$

(g) $\mathbb{H}_{PPA_i} \leftarrow \mathbb{H}(\mathbb{H}_{PC}, \mathbb{H}_{PrC}, \mathbb{H}_{TIC}, \mathbb{H}_{SIC}, \mathbb{H}_{EIC}, \mathbb{H}_{ROC})$

PPA Finalization

if PPA_i *is complete* **then**

/* presence of PC, PrC, TIC, SIC, EIC, ROC */

  if $(\mathbb{R}_{PPA} + PPA_i)$ *contains no conflicts* **then**

    (i) do $\mathbb{R}_{PPA} \leftarrow (\mathbb{R}_{PPA} + PPA_i)$

    (ii) add $\mathbb{ID}_{PPA_i}$ to patient profile, $\mathbb{P}_i$

    (iii) call $\mathbb{BN}_{SC}(\mathbb{ID}_{PPA_i}, \mathbb{H}_{PPA_i})$

/* PPA integrity verification reference */

    Return: Success (PPA_i added to $\mathbb{R}_{PPA}$)

  else

    Error: $(\mathbb{R}_{PPA} + PPA_i)$ contains conflicts

/* PPA_i revision required to add */

  end if

else

  Error: PPA_i cannot be created (incomplete PPA)

end if

(D) SIC is a finite set of sharing informed consent components. It denotes the patient's consent to sharing medical data for a specific purpose. Both the sender and the receiver must have consent.

(E) EIC is a finite set of emergency informed consent components. It denotes that the patient has permitted the designated treatment team to access medical records.

(F) ROC is a finite set of regulatory and other components. It has security and privacy policies in place to comply with the requirements of local, state, federal, foreign, and regulatory agencies (e.g., HIPAA, GDPR) as needed. It also includes contractual obligations in some cases.

A patient-provider agreement is formed upon a patient's hospital visit. The terms and conditions of the contract become invalid after a certain period. A single patient may have several contracts. Several patient-provider agreements must be created and adequately documented to deliver health-

care services. Managing many contracts involves tasks such as contract creation, development, testing, updating, and related activities. If the requests include contracts, the authorization module must consider them alongside other required policies when making access decisions.

4.4 Treatment Team PHI Access Policy Compliance

There are many users in the healthcare system. Each user plays a distinct role and assumes distinct responsibilities in performing their job. The treatment team for a patient includes doctors, nurses, support staff, lab technicians, billing officers, the patient's emergency contact, and other hospital employees assigned by the authority. Some outside members are insurance agents, pharmacists, or pharmacy technicians; doctors or lab technicians from another hospital. For a patient's treatment period, all aspects, from treatment to insurance coverage and billing, are taken into account. Informed consent users can be anyone from five groups of people: *(i) treatment team member, (ii) emergency contact, (iii) external users, (iv) insurance company agent,* and *(v) pharmacy*. External users are from different hospitals when a patient is transferred for better treatment, as needed. Usually, external users have temporary access to admitted patients' health records.

The term "object" refers to an electronic version of a patient's medical history maintained by the healthcare provider over time. It may include all the administrative and clinical information pertinent to the patient's care under a specific provider, such as demographics, progress notes, issues, medications, vital signs, previous medical history, immunizations, laboratory information, and radiology reports. These objects must be protected from unauthorized users. The primary purpose of informed consent is to permit patients to allow users to perform certain operations.

In the healthcare industry, many operations are performed by authorized personnel to fulfill required tasks. Some common operations are *view/read*, *add/write*, *update/modify*, *delete*, etc. In the *view* operation, users can only view or read healthcare records or resources if the request is valid and complies with all applicable policies. The state of the data is unchanged by this operation, ensuring data integrity. However, granting access without appropriate credentials can compromise

confidentiality and privacy. On the other hand, the *write* operation changes the state of the records or healthcare data. If proper policy enforcement is not enforced, data integrity is compromised.

4.4.1 Treatment Informed Consent (TIC)

Figure 4.3 shows the components of *Treatment Informed Consent*. The Treatment Informed Consent formally is composed of four tuples: $TIC = (U, O, OP, CON)$ satisfying the following requirements:

(a) U is a finite set of authorized users denoted as $\{u_1, u_2, u_3, ...u_u\}$. The user can perform operations on healthcare resources under specific conditions.

(b) O is a finite set of protected objects, otherwise known as protected healthcare resources. A finite set of protected objects (O) denoted as $\{o_1, o_2, o_3, ...o_o\}$.

(c) OP is a finite set of operations denoted by $\{op_1, op_2, op_3, ...op_p\}$. Operations represent the system actions that authorized users can perform on the objects. Examples of operations are read, write, and update.

(d) CON is a finite set of conditions. It indicates the conditions that must be satisfied by the user to perform operations on the protected objects. A finite set of conditions, CON, can be denoted as $\{con_1, con_2, con_3, ...con_n\}$.

There may be various constraints or conditions under which consent can be enforced, rejected, revoked, or otherwise. The conditions can be, but are not limited to:

(i) ***Time Constraints:*** In time constraints, any user can access a patient's healthcare data within a particular time. For example, the time condition for consent is regular office hours: 8 am–5 pm. In this case, the request is rejected if any subject seeks access to the patient's record beyond this time. The attempt is recorded as an audit trail event.

(ii) ***Date Constraints:*** The date constraints limit the calendar date. No access request is granted beyond the intended date.

Figure 4.3: Informed Consent Components [2, 11, 13]

(iii) ***Day Constraints:*** Day conditions can include work days (Monday-Friday), weekends (Saturday-Sunday, holidays, etc. Based on the day, the subject can access data. Suppose a regular doctor has a duty on workdays. On weekends, the doctor is unavailable.

(iv) ***Location-based Constraints:*** The location-based condition allows users to access information from a specific location, like a hospital building, inside an emergency room for treating emergency patients, and others.

(v) ***IP-based Constraints:*** IP-based condition limits healthcare users from accessing resources from specific IPs. Device IPs must be from the known list; otherwise, no access is granted.

(vi) ***Access Frequency Dependent:*** A user can operate for a certain number of times in access frequency-dependent conditions. Suppose an external doctor is given five times the view permission. Once the doctor has read the patient's specified records five times, the consent expires, and access is denied. There is no access without getting new consent.

The abovementioned list is not fixed under the conditions, but we consider them in this study. Other conditions may arise depending on the nature of the treatment, patient characteristics, provider business policies, and other relevant factors. With sophisticated technology, malicious attackers can

spoof conditions to trick the system into accessing sensitive healthcare data and other compromised credentials. Proper layered defense mechanisms must be deployed to ensure that the credentials of conditions are accurate, not fabricated or manipulated.

4.4.2 Treatment Informed Consent (TIC) Smart Contract Deployment

Once a Patient-Provider Agreement is created and stored in the repository, all informed consent components are deployed as smart contracts. Steps *4a*, *4b*, and *4c* in Figure 4.4 show the process. The authorization module must access these smart contracts and integrate them into the decision-making process alongside other required components. These components include the subject, object, operational attributes, environmental conditions, and organizational, regulatory, and additional policies as necessary. In this approach, a single smart contract serves as the consent container. If there is no contract, then a new contract is created and added to the patient profile and hospital systems for services. The contract address is an identifier for a smart contract on the blockchain that stores and retrieves informed consents. This smart contract comprises both functions and data, structured into two distinct data units: the consent repository and the consent archive (Figure 4.6). The repository holds active informed consents, accessible to the authorization unit for processing access requests. Conversely, the archive stores inactive, read-only historical consents. They are not executable for current authorizations and are crucial for compliance verification and resolving disputes in investigative or legal contexts.

The smart contract deployment unit, *SCDU*, collects all consent components from the PPA and verifies their integrity to confirm that the collected consents have not been deliberately or inadvertently modified. In step 3 in Figure 4.4, PPA integrity as the hash from Algorithm 1 ($\mathbb{H}_{PPA_i}$) is stored in the blockchain network along with *PPA ID*. To verify PPA integrity, *SCDU* calls the corresponding smart contract function to retrieve the PPA integrity value stored in the network. After receiving, it compares with the current integrity from the PPA repository. Any modification of consent components voids the consent. If no modifications are made, SCDU creates and deploys smart contract(s) to the blockchain network. The SCDU serves as a secure, trusted API that

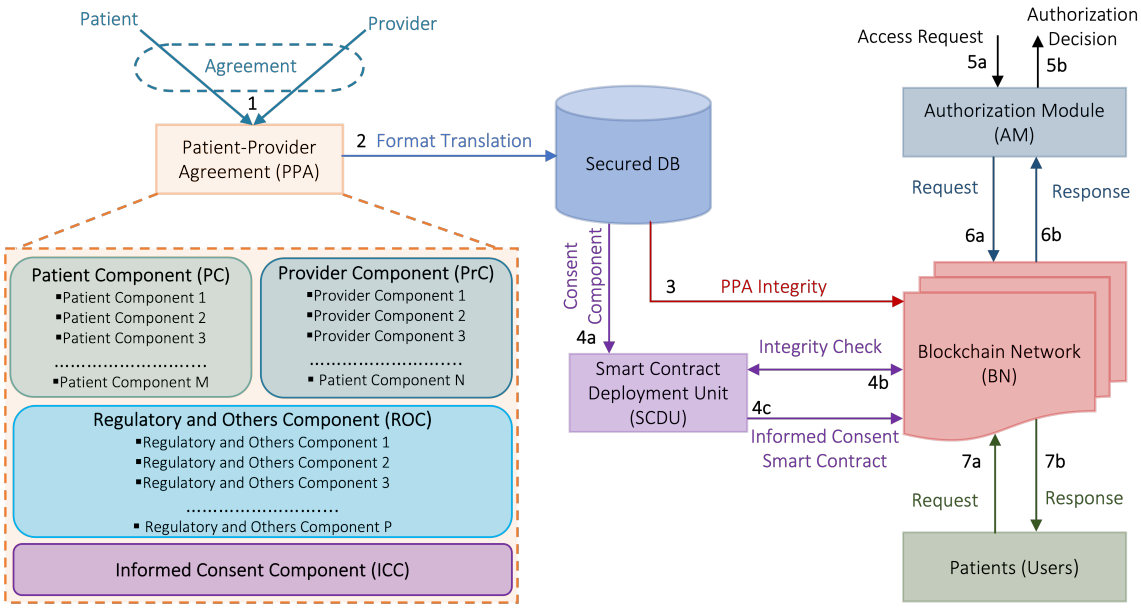

Figure 4.4: Treatment Informed Consent - Smart Contract Deployment [2, 11, 13]

maintains the integrity of consent components without modification. It also ensures that no consent-related information is disclosed to any unauthorized entities. However, this paper doesn't provide a detailed architecture and functional mechanisms of SCDU.

4.4.3 Patient Treatment Team Members

The treatment team for a patient includes doctors, nurses, support staff, lab technicians, billing officers, insurance agents, and the patient's emergency contact. As the treatment period for a patient, we encompass all aspects from treatment initiation to insurance coverage and billing. For this study, we consider one member from each category, such as doctors, nurses, and other healthcare professionals. However, there might be multiple team members in each category. Table 4.2 shows the patient-treatment team members and their responsibilities during and after treatment. Team members are randomly selected from those available. An emergency contact person is designated by the patient, not randomly selected for the treatment team. Algorithm 2 shows the steps of treatment team creation for a patient. Table 4.3 contains the PHI operations for treatment team members.

Table 4.2: Patient Treatment Team Members and Responsibilities [2, 11]

SN	Treatment Team Member	Responsibilities
1	Doctor (DOC)	Viewing patient's healthcare data
2	Nurse (NRS)	Creating new patient's healthcare data
3	Support Staff (STF)	Correcting erroneous or appending patient's healthcare data
4	Billing Officer (BLO)	Viewing patient's healthcare data
5	Radiology Lab Tech (RLT)	Creating new patient's healthcare data
6	Pathology Lab Tech (PLT)	Correcting erroneous or appending patient's healthcare data
7	Emergency Contact (EMC)	Viewing patient's healthcare data
8	Pharmacist (PHR)	Creating new patient's healthcare data
9	Insurance Agent (INA)	Correcting erroneous or appending patient's healthcare data

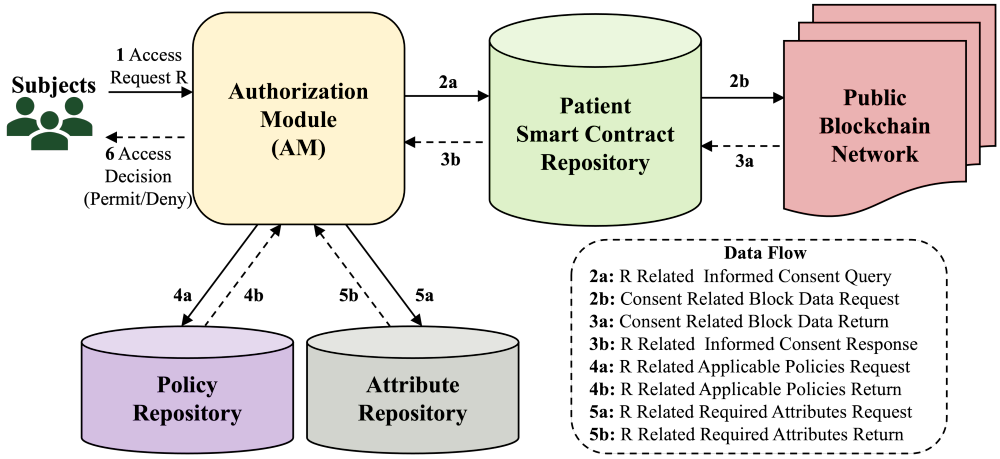

Figure 4.5: Informed Consent Enforcement Process for PHI Access Authorization [11]

4.4.4 Treatment Team PHI Access Authorization Process

Capturing and storing informed consent is not enough. There must be mechanisms in place to enforce consent for making PHI access authorization decisions for the treatment team member. Consent enforcement ensures that related consents are executed and that access decisions are made for requests. In the proposed model, all consents are stored on the public blockchain network as smart contracts and cannot be enforced until invoked. The *Authorization Module* considers all applicable consents from a patient while making an authorization decision for access requests. The *AM* also considers applicable policies and required attributes. The attributes can be subject, object, operation, or environmental attributes. Figure 4.5 shows the consent enforcement process.

When a subject sends an access request (R) in Step *1*, the *AM* checks the blockchain by contacting the patient's smart contract (Steps *2a* and *2b*) to find information on the informed consent

Algorithm 2: Patient Treatment Team (PTT) Creation [2]

Input : (i) DOC, (ii) NRS, (iii) STF, (iv) BLO, (v) RLT, (vi) PLT, (vii) EMC, (viii) PHR, (ix) INA

/* $\mathbb{R}_{PTT}$: secured PTT repository, BN_{SC}: blockchain network smart contract */

Result : A formal treatment team

PTT Member Initialization

$PTT_i \leftarrow \{DOC_i, NRS_i, STF_i, BLO_i, RLT_i, PLT_i, EMC_i, PHR_i, INA_i\}$ for patient identity i

(i) $DOC \leftarrow \{DOC_1, DOC_2, DOC_3, ..., DOC_D\}$

(ii) $NRS \leftarrow \{NRS_1, NRS_2, NRS_3, ..., NRS_N\}$

(iii) $STF \leftarrow \{STF_1, STF_2, STF_3, ..., STF_S\}$

(iv) $BLO \leftarrow \{BLO_1, BLO_2, BLO_3, ..., BLO_B\}$

(v) $RLT \leftarrow \{RLT_1, RLT_2, RLT_3, ..., RLT_R\}$

(vi) $PLT \leftarrow \{PLT_1, PLT_2, PLT_3, ..., PLT_P\}$

(vii) $EMC \leftarrow \{EMC_1, EMC_2, EMC_3, ..., EMC_E\}$

(viii) $PHR \leftarrow \{PHR_1, PHR_2, PHR_3, ..., PHR_H\}$

(ix) $INA \leftarrow \{INA_1, INA_2, INA_3, ..., INA_I\}$

PTT Finalization

if PTT_i *is complete* **then**

/* complete: presence of all members */

  if $(\mathbb{R}_{PTT} + PTT_i)$ *contains no conflicts* **then**

    (i) do $\mathbb{R}_{PTT} \leftarrow (\mathbb{R}_{PTT} + PTT_i)$

    (ii) add $\mathbb{ID}_{PTT_i}$ to patient profile, $\mathbb{P}_i$

    (iii) call $\mathbb{BN}_{SC}(\mathbb{ID}_{PTT_i}, \mathbb{H}_{PTT_i})$

/* later PTT integrity verification */

    Return: Success (PTT_i added to $\mathbb{R}_{PTT}$)

  else

    Error: $(\mathbb{R}_{PTT} + PTT_i)$ contains conflicts

/* PTT_i revision required to add */

  end if

else

  Error: PTT_i cannot be created (incomplete PTT)

end if

related to the request (Steps *3a* and *3b*). It also looks up policies related to the request in Steps *4a* and *4b* and gathers the necessary attributes in Steps *5a* and *5b*. After reviewing the consent, policies, and attributes, the *AM* decides whether to grant access. This decision is returned to the subject in Step *6*. Following the decision, the AM records details of the decision-making process on the blockchain as event logs through the smart contract.

Table 4.3: Treatment Team Member-Oriented PHI Operations [2, 11]

PHI ID	Read Operation	Write Operation	Update Operation
PHI1001	*Patient*, DOC, STF, EMC	*Patient*, STF	*Patient*, STF
PHI1002	DOC, *Patient*	*Patient*, DOC	*Patient*, DOC
PHI1003	DOC, *Patient*, PLT	PLT	PLT
PHI1004	DOC, *Patient*, NRS	*Patient*, PLT	*Patient*, PLT
PHI1005	DOC, NRS, *Patient*, EMC	DOC	DOC
PHI1006	DOC, *Patient*, NRS, PHR, INA, EMC	DOC	DOC
PHI1007	PLT, DOC, *Patient*, EMC	PLT	PLT
PHI1008	RLT, DOC, *Patient*, EMC	RLT	RLT
PHI1009	*Patient*, BLO, INA	BLO, *Patient*	BLO, *Patient*
PHI1010	*Patient*, BLO, INA	BLO, INA	BLO, INA

This study assumes the authorization module is secure and has not been tampered with. The communication between the *AM* and the blockchain is protected against attacks by malicious users. We don't provide the detailed structure of the authorization module. However, we refer the interested readers seeking a deeper understanding of the module's functionalities to the *Attribute-Based Access Control (ABAC)* model, as detailed in the work of Hu et al. [128].

4.4.5 Treatment Team PHI Access - Experimental Evaluation

Ethereum Virtual Machine (EVM)-based blockchains are selected for the experiment of the proposed approach. It offers a Turing-complete smart contract language, *Solidity*, which enables the implementation of our model's logic. We developed smart contracts for storing and retrieving informed consent and tested them on the test networks *Arbitrum*, *Polygon*, and *Optimism* to ensure reliability before deployment. Since smart contracts, once deployed, are immutable and errors can incur financial and reputational costs, rigorous testing on these networks is crucial. Ethereum's *Remote Procedure Call (RPC) API* is used to deploy smart contracts to these test networks [129]. Using public RPC, a toolkit for blockchain application development, eliminates the need to maintain a blockchain node for contract interactions, while incurring minimal resource usage (CPU, HDD, bandwidth) on the local machine. Faucet *ETH* and *MATIC* serve as gas to authorize transactions using the *Metamask* wallet [85].

Figure 4.6 shows the smart contract structure that acts as a container. Each smart contract has functions and data as storage. Functions perform specific operations, including consent creation, alteration, termination, and expiration. The contract also stores consent as data. There are two storage scopes: an informed consent repository that contains active consent executable for authorizations, and an informed consent archive that contains historical consent resulting from consent alteration, termination, and expiration operations. The archive provides a read-only repository, so consent from here cannot be executed. A transaction is depicted in Figure 4.7 from *Blockchain Explorer*. The following discusses gas consumption and time requirements to assess the functionalities of the proposed approach for three test networks: *Polygon*, *Arbitrum*, and *Optimism*.

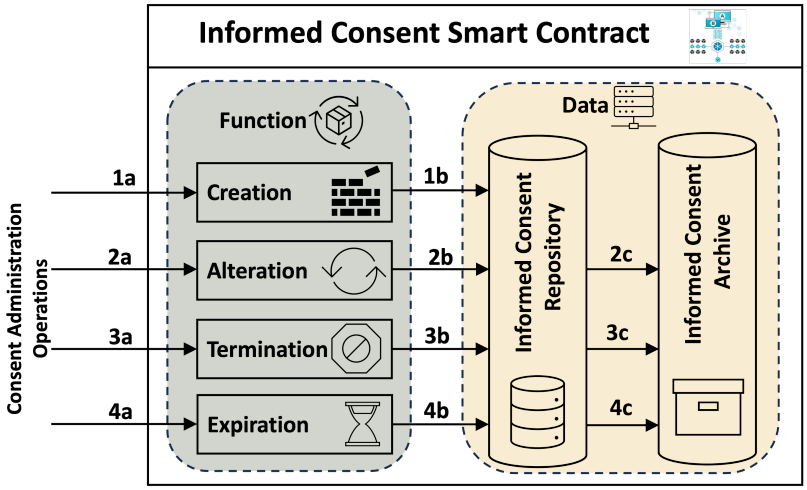

Figure 4.6: Informed Consent Smart Contract Structure [11]

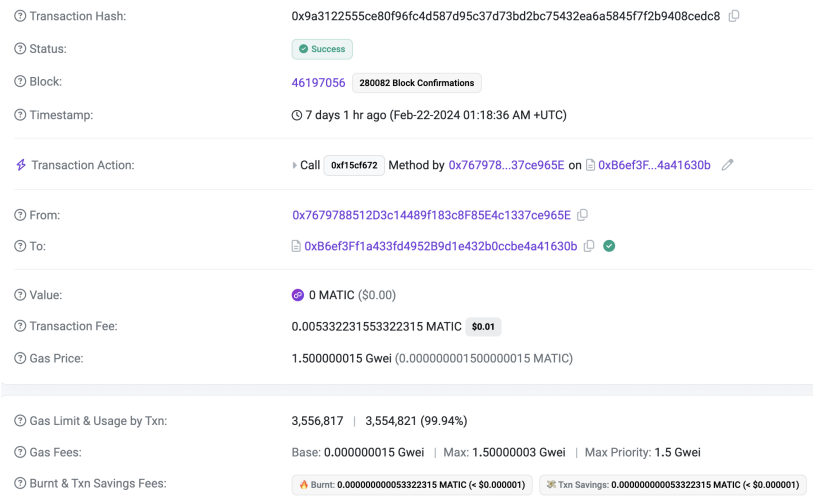

Figure 4.7: Informed Consent Transaction Information [11]

Smart Contract Deployment and Consents Storage Cost

Gas is required for any public Blockchain activity that writes or modifies data [130]. Some functions send ether (or any other ERC20 token), mint and send NFTs, deploy smart contracts, modify blockchain state, and so on. For this work, we only need to consider the costs of smart contract deployment and function calls for writing data to the blockchain network. The cost of calling a function depends on the number of times it is invoked and on the amount of data that must be stored or modified on the blockchain network.

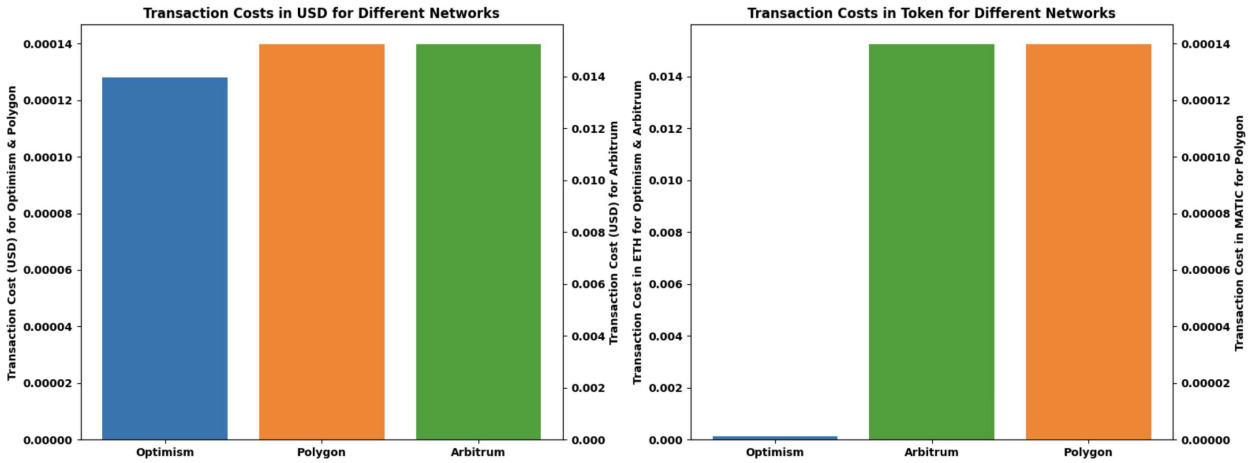

Figure 4.8: Patient-Provider Agreement - Integrity Writing Cost [11]

PPA Integrity Storage Cost: Figure 4.8 displays two side-by-side bar graphs comparing transaction costs for *PPAIngerityContract* smart contract deployed in *Optimism, Polygon*, and *Arbitrum* test networks. The size of a PPA hash for integrity is 32 bytes. The left graph shows costs in USD, while the right graph presents costs in native tokens (*ETH* for Optimism and Arbitrum, *MATIC* for Polygon). A clear trend is evident: Arbitrum's transaction costs are substantially higher than those of Optimism and Polygon, with its bars reaching the upper limits of the graphs. This suggests that users may face significantly higher fees on the Arbitrum network, which could influence their choice of platform for transactions or smart contract interactions.

Patient Smart Contract Deployment Cost: Figure 4.9 displays two bar charts comparing the gas costs for deploying identical smart contracts across three test networks. The left chart uses a logarithmic y-axis to highlight the large difference in deployment costs, particularly showing that Arbitrum incurs higher costs than Optimism. The variation in deployment costs is attributed to the cost of native tokens and network congestion levels at the time of deployment. These factors contribute to the observed cost differences despite identical contract codes, as illustrated in the figure.

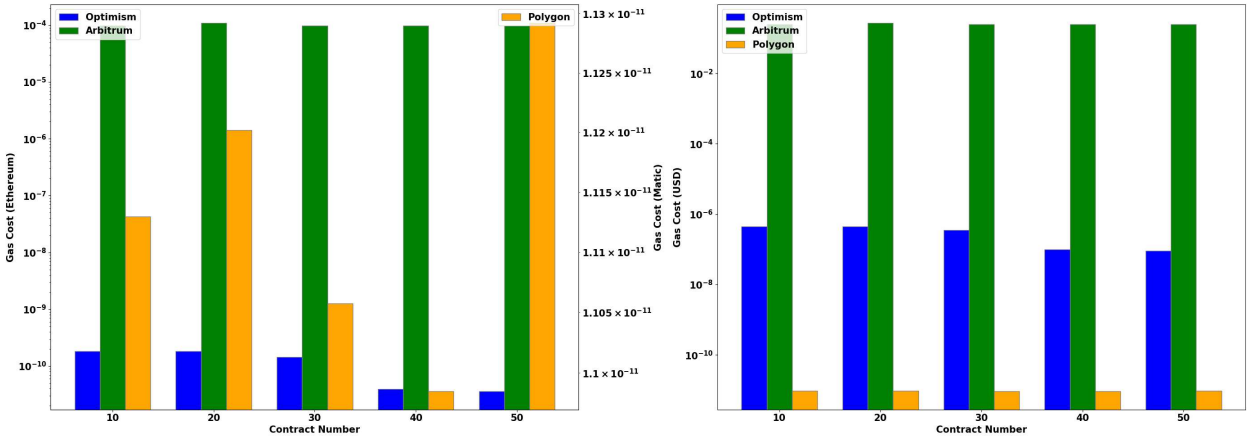

Figure 4.9: Treatment Informed Consent - Smart Contract Deployment Cost [11]

Informed Consents Writing and Reading Time Requirement

All the read calls of smart contracts are gas-free. Smart contract deployment and execution stages are the basis for the time cost of on-chain activities. Any blockchain-based applications require two kinds of time requirements: *(i) block data writing* and *(ii) block data reading*.

Writing Time: Writing time includes smart contract deployment and adding data. A new block is added to the Ethereum main network every 12 seconds on average, ideal for the proposed purposes [131]. So long as there is sufficient space in new blocks, a new transaction would take, on average, no more than 12 seconds. If block congestion occurs, the time required for a transaction to be included in a block may increase. However, users can influence this by paying more gas for faster block confirmation. Given that users may artificially extend the confirmation time of their transactions. Table 4.4 shows the writing time for various consent numbers for test networks. Table 4.10 depicts the writing-time consent administration operations for the same test networks: alteration, termination, and expiration. These operations require moving consents from the active repository to a read-only archive. In both tables, *Arbitrum* requires less time than the other two networks. This is because of the sequencer design and network congestion management [132]. The same smart contracts and consents are used for all test networks.

Reading Time: The reading time indicates the required time to get data from the block of the blockchain ledger. All the read calls of smart contracts are gas-free. Table 4.5 depicts the reading

Table 4.4: Treatment Informed Consent Writing Time [11]

Consents	Polygon	Optimism	Arbitrum
4	5.708 Sec	2.744 Sec	1.885 Sec
8	6.689 Sec	2.702 Sec	1.633 Sec
12	7.092 Sec	3.562 Sec	6.825 Sec
16	5.418 Sec	8.464 Sec	3.181 Sec
20	7.448 Sec	7.363 Sec	1.586 Sec
24	5.457 Sec	8.375 Sec	5.778 Sec
28	6.772 Sec	7.805 Sec	1.730 Sec
32	5.972 Sec	7.943 Sec	3.390 Sec
36	5.542 Sec	7.736 Sec	1.834 Sec
40	6.128 Sec	7.573 Sec	2.119 Sec
44	7.536 Sec	7.390 Sec	7.536 Sec
48	5.521 Sec	7.698 Sec	3.394 Sec

Table 4.5: Treatment Informed Consent Reading Time [11]

Consents	Polygon	Optimism	Arbitrum
4	0.466 Sec	0.228 Sec	0.427 Sec
8	0.472 Sec	0.52 Sec	0.289 Sec
12	0.497 Sec	0.208 Sec	0.727 Sec
16	0.591 Sec	0.201 Sec	0.975 Sec
20	0.504 Sec	0.223 Sec	0.330 Sec
24	0.923 Sec	0.221 Sec	0.304 Sec
28	0.600 Sec	0.235 Sec	0.305 Sec
32	0.909 Sec	0.245 Sec	0.32 Sec
36	0.526 Sec	0.229 Sec	0.719 Sec
40	0.812 Sec	0.257 Sec	0.363 Sec
44	0.742 Sec	0.457 Sec	0.247 Sec
48	0.631 Sec	0.557 Sec	0.266 Sec

time for various consent numbers for test networks. For the same test networks, the reading time for consent administration operations is tabulated in Table 4.11. Maintaining a node locally reduces network read time, enabling real-time access to block data. The system continuously synchronizes with the blockchain network to update the ledger data. Hospital authorities can maintain local nodes to expedite authorization decisions.

4.5 PHI Sharing Beyond Treatment Team Policy Compliance

The secure and compliant sharing of protected health information is paramount in the rapidly evolving digital healthcare landscape. This paper underscores the dual focus on the benefits of PHI sharing and the stringent need for security and privacy policy compliance. PHI sharing significantly enhances the quality of patient care and coordination, contributing to more accurate diagnoses, efficient treatment plans, and a comprehensive understanding of the patient's history. However, compliance with strict privacy and security policies, such as those mandated by laws such as HIPAA, is required to realize these benefits. A critical innovation in this domain is the integration of blockchain technology, which provides a decentralized, tamper-evident ledger system. This system ensures the authenticity and integrity of PHI while facilitating patient consent management. The incorporation of smart contracts into blockchain platforms further revolutionizes the sharing of PHI. These contracts automate consent-related processes, ensuring that PHI access and sharing comply with patient preferences and legal requirements. This approach streamlines the consent management process and enhances trust and transparency between patients and healthcare providers. The synergy between efficient PHI sharing, stringent policy compliance, and blockchain-enabled consent systems promises to significantly improve healthcare delivery while upholding the highest data privacy and security standards.

4.5.1 PHI Sharing Problem Motivation

Electronic health record systems have significantly improved healthcare services, such as enhanced collaboration among healthcare professionals, more accurate diagnoses, faster treatment, and convenient access to patient-protected health information [133]. EHR systems greatly facilitate access to and sharing of digitized healthcare information, enabling providers to exchange sensitive medical data with other professionals easily. Data sharing is essential for numerous aspects of patient care, including improving diagnosis and treatment plans through consultations with specialists, leveraging advanced technologies to enable more precise radiology and pathology analyses, and enhancing the overall quality of patient care [134]. Furthermore, healthcare data are used for

research and marketing, provided that specific requirements are met [135]. Health records can be shared through the EHR system using health information exchanges (HIE), specialized networks that rely on interoperable systems to share electronic health information seamlessly and securely [136]. Providers also share PHI through email or other electronic mediums [137]. Regardless of the PHI-sharing mechanism, ensuring the security of health data and patient privacy is mandatory.

Acquiring patient consent for healthcare information sharing is paramount for adhering to policy compliance, particularly concerning regulations like the *HIPAA* in the U.S. and the *GDPR* in the E.U. [138]. These regulatory frameworks emphasize the protection of health information and the patient's right to privacy. Patient consent is a cornerstone of these regulations, ensuring individuals have control over their health data and its dissemination. Under HIPAA, healthcare entities must obtain explicit consent before sharing healthcare data for purposes beyond treatment, payment, or healthcare operations. Similarly, the GDPR imposes strict guidelines on data consent, processing, and privacy, granting individuals the 'right to be forgotten' and the autonomy to decide how their data is used and shared. From a policy compliance perspective, obtaining proper patient consent is both a legal requirement and a trust-building measure, reinforcing the patient-provider relationship. It ensures transparency in data handling and builds patient confidence, knowing their sensitive information is shared respectfully and responsibly. As healthcare continues to integrate with various technologies, upholding these consent protocols is crucial for maintaining the security and privacy of patient data and adhering to global data protection standards.

Unauthorized access to and disclosure of health data are everyday occurrences in the healthcare industry and heighten security and privacy concerns. Table 1.1 shows the number of compliance complaints received by the *U.S. Department of HHS OCR* [1]. 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. These issues can be minimized by obtaining patients' consent for data access and sharing decisions and by employing appropriate data protection mechanisms, such as encryption and anonymity. Consent enables patients to control

their healthcare journey, allowing them to make choices that align with their best interests and well-being [139].

Enhanced security and privacy technologies are essential for protecting patient data against compromise, misuse, or disclosure. However, substantial evidence indicates that the root of many unauthorized EHR access and sharing lies in inadequate policy adoption, implementation, and enforcement [140, 141]. Often, users are granted access privileges inappropriately, whether intentionally or not. Policy compliance often falls short, and access control measures are not rigorously monitored or enforced in a timely manner. A common oversight is the blanket assignment of identical roles and privileges to all employees, neglecting the nuances of individual patient-level policies. Moreover, auditing and monitoring practices are typically reactive, triggered only by serious complaints or legal mandates, rather than proactive and consistent. These policy specification and enforcement flaws significantly affect informed consent policies, underscoring the need for a more accurate and systematic approach to protect patient healthcare data and preserve privacy.

It is essential to address the following concerns to guarantee compliance with the applicable privacy and security policies, industry best practices, and contractual obligations for sharing PHI: (i) Patient-level policies or consents are often not timely or adequately enforced in healthcare data sharing. (ii) Patients lack assurance that consent for access or sharing purposes is carried out strictly by designated users, and only if the stipulated conditions are met are all other requests rejected. (iii) Data sharing over email or other mediums is insecure due to the absence of encryption or the use of inadequate and weak encryption algorithms and key sizes. (iv) The centralized hospital system serves as a singular source of truth and a potential single point of failure for managing audit trails. (v) The absence of a verifiable, unaltered record for consent execution and sharing PHI highlights the need for comprehensive consent provenance. (vi) Compliance assessments and audits are not conducted accurately and promptly to check compliance status.

To address the aforementioned challenges and requirements, this paper proposes a blockchain- and smart-contract-based framework for managing and enforcing informed consent when sharing PHI with entities outside the treatment team. The approach ensures that PHI is shared only when

the sender has obtained the necessary permission from the patient and that the sharing aligns with specific, predefined purposes. In addition to enforcing patient consent, this approach integrates other relevant security policies and industry best practices to ensure data protection. The *HIPAA Security Rule* mandates that the requirements for transmission security are outlined under *45 CFR § 164.312(e)(1) Technical Safeguards* [142]. However, the proposed approach does not directly guarantee security mechanisms like encryption for data protection. Instead, it leverages an honest broker that acts as a blind, secure entity to evaluate the intended PHI and certify its status, with respect to whether the required protection mechanisms are satisfied [143].

The broker's attestation is then recorded in blockchain-based audit trails, along with other relevant activity data, to support future compliance evaluations and validation. It supports the use of blockchain-based audit trails or provenance mechanisms, which are essential for tracking PHI-sharing activities. Moreover, the proposed framework provides a compliance-checking mechanism in data-sharing activities, ensuring adherence to applicable policies.

Smart contracts, [144], provide an automated, transparent system that upholds the integrity and accountability of consent to share PHI. Through this smart contract-based approach, the proposed framework not only automates processes but also guarantees the accurate execution of informed consent, thereby enhancing the security and reliability of PHI sharing. Blockchain technology ensures the immutability of submitted records, safeguarding the integrity of the audit trail and enabling the detection of any unauthorized alterations. Blockchain security features, including non-repudiation, ensure that participants cannot deny their actions [145]. Smart contracts are also designed to generate notifications for operational activities, enhancing transparency and accountability [146].

This work is the first to capture patients' informed consent for PHI sharing, to ensure policy compliance by preserving provenance, and to conduct compliance checks. It also considers and enforces other applicable security policies and industry best practices mandated by laws, regulations, standards, and contractual obligations to meet compliance requirements. Specifically, this paper makes the following contributions.

- Implementing a mechanism to capture patients' consent for sharing healthcare data beyond the treatment team members.
- Storing obtained consents in decentralized and distributed networks (blockchain) to overcome a single point of truth sources and failure.
- Considering applicable security and privacy policies, regulatory requirements, and contractual obligations to ensure compliance-based sharing.
- Enforcing informed consent and applicable policies while making authorization decisions to share health records.
- Equipping blockchain-based audit trail mechanisms to guarantee data provenance.
- Incorporating compliance assessment methods to identify compliance and non-compliance with PHI sharing.
- Last but not least, offering consent services to provide precise and comprehensive insights into the consent granted and the extent of its execution.

4.5.2 PHI Sharing Policy Compliance - Proposed Approach Overview

The main objective is to ensure compliance with applicable security and privacy policies for PHI sharing. To ensure compliance, we need effective policy enforcement, including the maintenance of provenance and the prompt and adequate performance of compliance status checks. For enforcement, this paper considers patient-informed consent, in which the sender has the patient's permission to share the intended PHI with the receiver for specific purposes. Also, proper data protection mechanisms are considered. However, rather than directly ensuring data protection, this work relies on an honest broker to verify and certify the data protection mechanism. PHI-sharing activities are recorded as audit trails to provide provenance and reconstruct events in a manner that reflects their actual occurrence. A private blockchain-based approach is proposed (Chapter 5). Finally, a blockchain consensus mechanism, *Proof of Compliance*, is introduced in Chapter 6 for auditing. This audit rigorously examines enforcement actions against policy standards and informed consent, using provenance data to verify and certify the policy's compliance status when sharing health

records. The seamless integration of policy enforcement, provenance, and the auditing process underpins a secure and compliant system.

4.5.3 Sharing Informed Consent (SIC) Structure

Sharing informed consent is formally composed of four tuples: $SIC = (S, R, PHI, P)$ satisfying the following requirements:

(a) S is a finite set of authorized senders denoted as $\{s_1, s_2, s_3, ...s_s\}$ for s number of senders. The sender may share certain healthcare data with the receiver, provided the receiver has the patient's permission. The sender may be a member of the patient's treatment team or a representative of the provider.

(b) R is a finite set of authorized users who receive protected health information from authorized senders. A finite set of r number authorized receivers denoted as $\{r_1, r_2, r_3, ...r_r\}$. The receiver may be from other hospitals, labs, medical research institutes, pharmaceutical companies, marketing departments, government officials, and other relevant healthcare and research organizations.

(c) PHI is a finite set, d number, of health data denoted by $\{phi_1, phi_2, phi_3, ...phi_d\}$. It is an electronic version of a patient's medical data that healthcare providers keep over time. They are protected health information and contain sensitive patient information. PHI must be protected from any unauthorized access, disclosure, or sharing.

(d) P is a finite set of purposes. It indicates the senders' objective in sharing PHI with the receivers. Receivers must use the received PHI for the intended purposes. A finite set of purposes, a p number, can be denoted as $\{p_1, p_2, p_3, ...p_p\}$.

The objective of sharing protected health information specifies the reasons for its disclosure. The recipient must utilize the shared PHI exclusively for its designated purpose. The potential reasons for sharing PHI in this study include, but are not limited to:

(i) ***Treatment:*** Providers or patients need to share PHI with other providers from external hospitals to provide better treatment. Additionally, patients may need to relocate to different

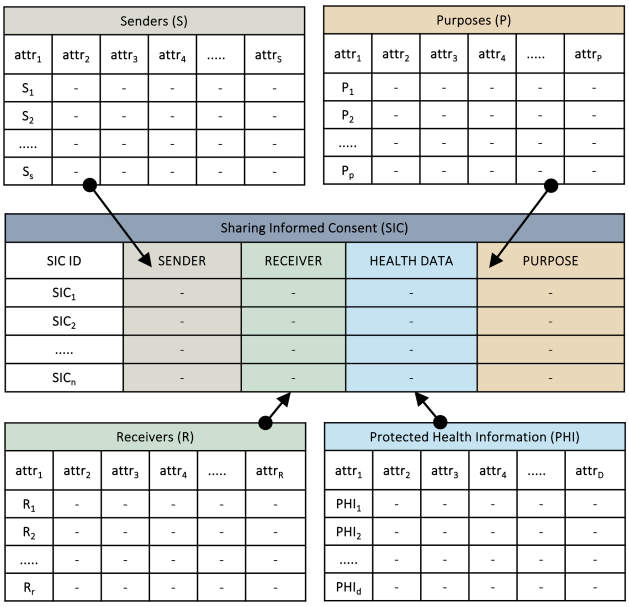

Figure 4.10: Sharing Informed Consent Structure [3]

regions, such as states or countries, due to family relocation, job transfers, or new employment opportunities. Patients need to share or transfer their healthcare data from previous providers to their current one.

(ii) ***Diagnosis:*** Present providers sometimes need more skilled human resources, appropriate machinery, instruments, or sophisticated technology to diagnose disease. However, it is urgent to do so to provide appropriate treatment and services, thereby saving patients' lives or minimizing harm. Patients' health data must be transferred or shared with other providers or labs to complete the diagnosis and develop appropriate treatment plans.

(iii) ***Marketing:*** Healthcare data sharing for marketing purposes involves using patient data to promote healthcare services, products, or initiatives. This can help healthcare providers tailor their services to patient needs, inform patients about new treatments or products, and improve patient engagement. Only the receiver entity can use the shared data as intended and should not share it with other associates for extended business purposes.

(iv) ***Research:*** Sharing PHI for medical research purposes holds significant potential for advancing medical knowledge, leading to breakthroughs in understanding diseases, improving and

developing new treatments, improving healthcare systems and services, and enhancing patient outcomes. Patients' privacy and rights must be respected.

Other purposes may exist, depending on the nature and requirements of the treatment, the patient's condition, and the provider's business policy, among other factors. This study considers only the four purposes mentioned above. After receiving the shared data, the receiver performs the specified operations to complete the task. It is assumed that the receiver cannot share data with other users who lack the patients' permission. More specifically, the receiver's healthcare system does not allow the sharing of PHI by any means, such as printouts, email, or screenshots. However, this paper doesn't provide detailed mechanisms or techniques for preventing data sharing without patients' consent at the receiver end.

4.5.4 Sharing Informed Consent - Smart Contract Deployment

Once a *Patient–Provider Agreement* is finalized, it is securely stored in the PPA repository. Subsequently, an integrity marker, such as the hash value $\mathbb{H}_{PPA_i}$ generated by Algorithm 1, is recorded on the blockchain, along with the corresponding PPA identifier, to enable detection of any unauthorized modifications in the future. These processes are illustrated in *Steps 2 and 3* of Figure 4.11. Afterward, the *Smart Contract Deployment Unit* retrieves all sharing-informed-consent components from the finalized PPA, as shown in *Step 4*. In *Step 5*, the *SCDU* verifies the integrity of these components to ensure that no intentional or accidental alterations have occurred. As a trusted and secure entity, the *SCDU* does not modify the content of the consent; any detected alteration renders the consent invalid.

If the consent components are verified as authentic and unchanged, the *SCDU* proceeds to create and deploy the corresponding smart contract on the blockchain network in *Step 6*. For each patient, a single smart contract stores all consent records associated with that patient. If no such smart contract already exists, the authority deploys a new contract, transfers ownership to the patient, and updates the smart contract address in both the patient profile and the hospital information systems. The contract address serves as the unique identifier for a smart contract on a blockchain network.

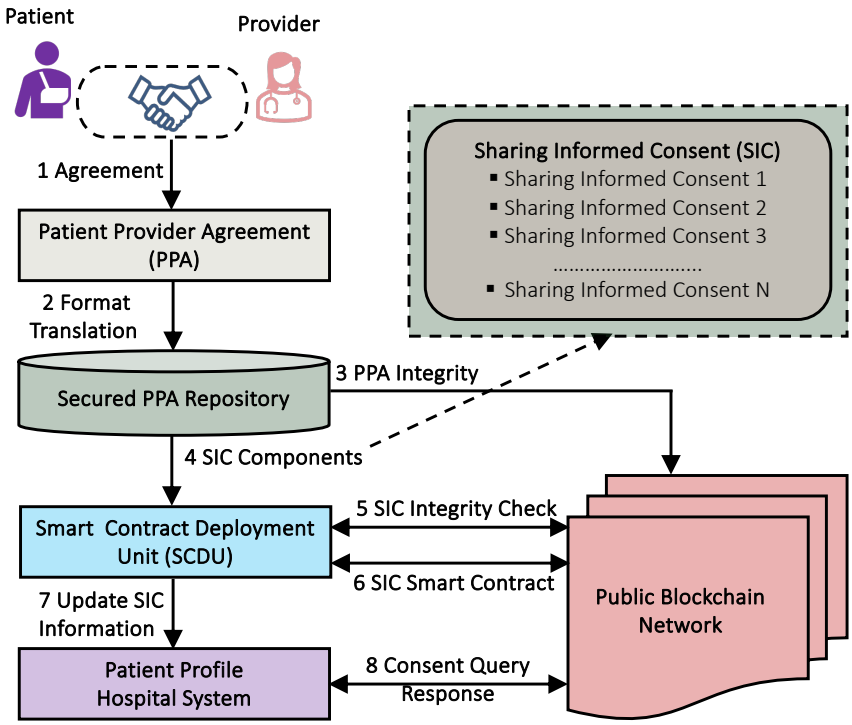

Figure 4.11: Sharing Informed Consent - Smart Contract Deployment Process [3]

Once deployment is completed, the patient profile and hospital systems are updated accordingly in *Step 7*.

This smart contract-based mechanism provides an automated, transparent, and accountable framework for managing informed consent. Once consent records are deployed to or added to the smart contract, they become immutable and cannot be altered. In *Step 8*, authorized users with the required credentials can query the blockchain network directly to obtain informed consent responses. The authorization module interacts with these smart contracts. It integrates their outputs with sender and receiver attributes, purpose specifications, environmental conditions, organizational policies, and regulatory requirements to inform decisions on PHI-sharing requests.

4.5.5 Honest Broker, Applicable Policies, and Industry Best Practices

In addition to patient consent, the proposed approach incorporates relevant security policies and industry best practices before sharing protected health information. For instance, a security policy

might require a data protection mechanism during data transfer between systems. For treatment and diagnosis purposes, encryption is a recommended protection method.

As an industry best practice, the *Advanced Encryption Standard (AES)* is preferred over the *Data Encryption Standard (DES)*. Furthermore, it advises using a robust, lengthy encryption key (*256 bits*) rather than a weaker, shorter one (*64 or 128 bits*). The sender must encrypt the intended PHI using the *AES-256* algorithm while leaving it in the system for treatment and diagnosis. However, this proposed approach does not directly encrypt healthcare data or ensure a strong key size during encryption. Additionally, it does not address key management mechanisms, including creation, storage, sharing, updating, and deletion. It is assumed that key management is performed securely and independently.

Similarly, anonymity is a recommended protection method for marketing and research purposes, where patient identifiers must be removed before sharing. The targeted PHI must be anonymized using proper techniques and tools before sending the data from the host healthcare system to the receiver. The host system indicates where patients' PHI is created or presently stored. Healthcare organizations deploy appropriate encryption and anonymity mechanisms. This study does not directly ensure PHI encryption and anonymity. Instead, this approach relies on an honest broker, a trusted entity that evaluates the encryption algorithm, key size, and the data's anonymity status [143]. After verification, the honest broker certifies or attests to the status, which is recorded in audit trails as proof of policy compliance verification, along with other components, such as informed consent and timestamps.

Depending on the healthcare organization's policies and practices, this broker could be either a human or an automated (non-human) entity. The honest broker's role is confined; it does not share healthcare data with other entities. It also does not analyze data to gain insights about the patient or share those insights. Effectively, it functions as a 'blind' entity, ensuring encryption standards and the anonymity status of the PHI without engaging with the actual data content.

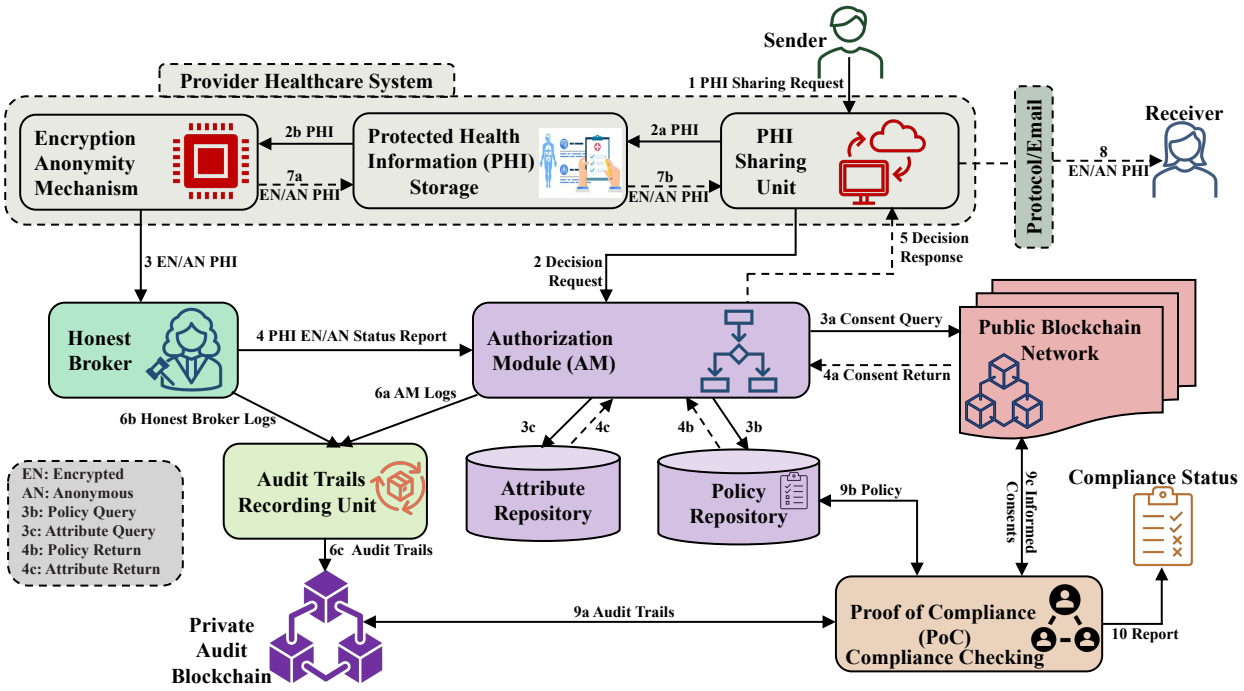

Figure 4.12: Compliance-Based PHI Sharing Authorization Process [3]

4.5.6 PHI Sharing Authorization Process

Consent enforcement ensures that related consents are executed when making decisions about PHI-sharing requests. All consents are stored on the public blockchain network as smart contracts and cannot be enforced until invoked. The *Authorization Module* considers the sharing of informed consent in accordance with applicable policy and the required attributes when making decisions. The attributes may be subject, object, operation, or environmental attributes. The sender must provide the necessary credentials for identification and authentication. Figure 4.12 illustrates the enforcement of informed consent for PHI sharing authorization.

A sender submits a data sharing request to the PHI sharing unit in *Step 1*. The sharing unit forwards the request to the authorization module for a decision in *Step 2*. It also requests that the PHI storage unit send the intended PHI to the protection mechanism unit in *Steps 2a* and *2b*. The honest broker receives encrypted or anonymized data in *Step 3*. After analyzing, it sends a report to AM in *Step 4*. The AM queries the blockchain network via the corresponding smart contract to obtain the sharing-informed consent information for the sharing requests in *Step 3a* and *4a*. It also

makes queries for requests related to applicable policies and required attributes in *Steps 3b* and *3c*. It receives the policy and attributes in *Steps 4b* and *4c*. After evaluating, it makes an authorization decision and sends it to the sharing unit in *Step 5*. If the request is approved, the sharing unit receives encrypted or anonymized data, depending on the purpose, in *Steps 7a* and *7b*. Then, it delivers the intended PHI through email or protocol to the receiver in *Step 8*.

The audit trail recording unit collects logs from AM in *Step 6a* and from the honest broker in *Step 6b*. It combines logs and stores them as an audit trail in *Step 6c* of the Private Audit Blockchain. Chapter 5 discusses block structure and others. The compliance status checking is done in *Steps 9a, 9b*, and *9c* by the *PoC* consensus mechanism. Compliance status reports are produced in *Step 10*. Chapter 6 discusses the required mechanism. For this study, it is assumed that the authorization module remains uncompromised and untampered with. It serves as the reference monitor for access decisions and must be tamper-proof [147]. Also, the communication channel between AU and the smart contract access points or apps is secured from malicious users.

4.5.7 PHI Sharing - Experimental Evaluation

The *Ethereum Virtual Machine (EVM)*-based three blockchain test networks (*Arbitrum*, *Polygon*, and *Optimism*) are chosen for the experiments. We developed and deployed smart contracts to store and retrieve *PPA* integrity and informed consent data in test networks. *Ethereum's Remote Procedure Call (RPC) API* is used to deploy smart contracts and execute transactions on these networks [129]. Utilizing public *RPC* eliminates the need to maintain a blockchain node for contract interaction, assuming minimal resource usage (*CPU, HDD, Bandwidth*) on the local machine. We used the Metamask wallet to sign and authorize transactions using *ETH* and *MATIC* faucet tokens as gas. Healthcare providers may invest in infrastructure, such as blockchain nodes, web interfaces, and mobile applications, to enable seamless service interactions between patients and healthcare systems. Storing informed consent on public blockchains like *Ethereum* incurs direct monetary costs. Patients, insurance companies, and others can share these costs, such as those for doctor visits, medications, and laboratory tests. The following discusses gas consumption and time requirements.

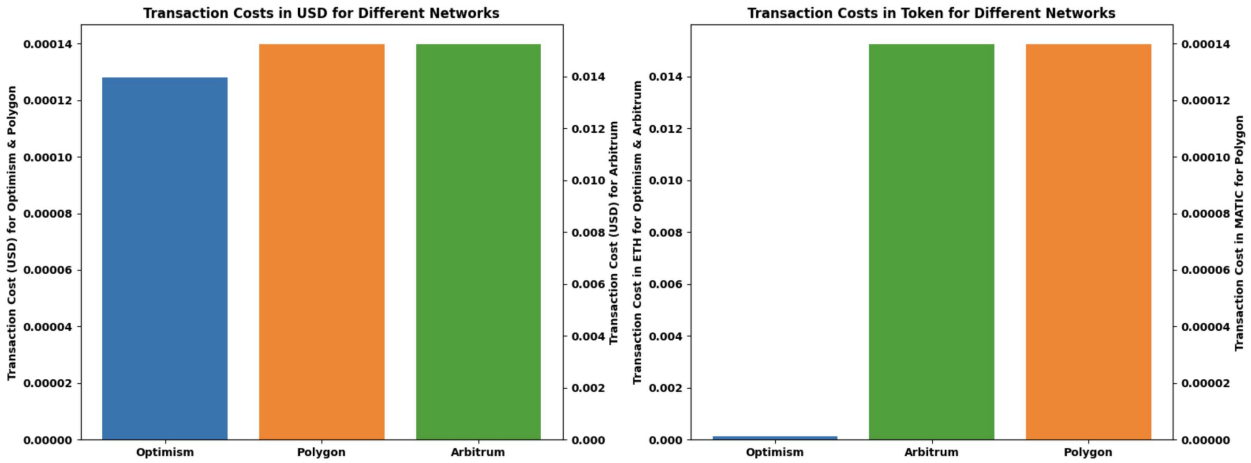

Figure 4.13: PHI Sharing - PPA Integrity Storage Cost [3]

Gas Consumption

Gas is required for all *Ethereum* network activities that involve writing data or modifying the blockchain state. Smart contract deployment and function calls incur costs for writing data to the blockchain network, which are considered in this work. A contract is deployed for each patient separately to manage consent-related queries efficiently. The cost of smart contract deployment is proportional to the code size [148]. This is a one-time cost for a single-contract deployment. The cost of calling a function depends on the number of times it is invoked and on the amount of data that must be stored or modified on the blockchain network. Figure 4.13, 4.14, 4.15, 4.16, and 4.17 show the contract deployment and consent storage costs in gas (token) and USD for three test networks.

Time Requirement

Blockchain-based applications have specific requirements for block data writing and reading times. Writing time includes smart contract deployment and data addition. Table 4.6 shows the writing time for various consent numbers for the test networks. The reading time is the time required to retrieve data from a block in the blockchain ledger. All the read calls of smart contracts are gas-free. Table 4.7 shows the test network's reading time for various consent numbers. The same smart contracts and consents are used for all test networks. Maintaining a node locally reduces network

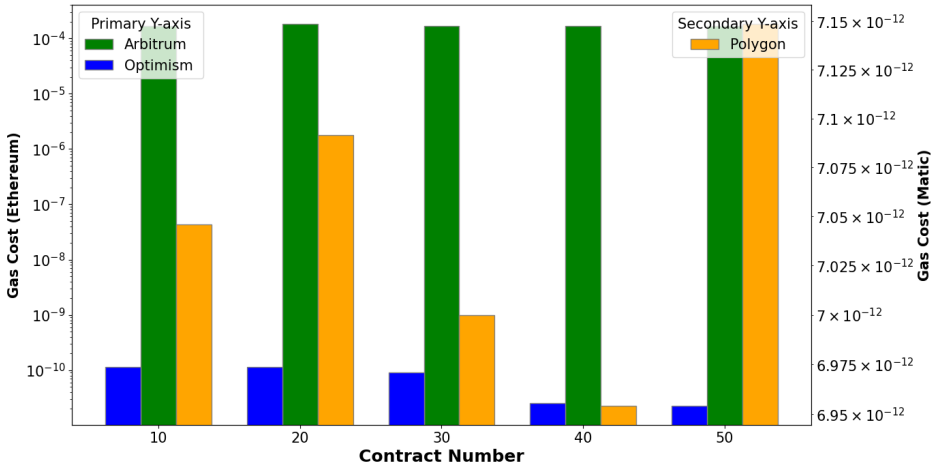

Figure 4.14: PHI Sharing - Smart Contract Deployment Gas Cost [3]

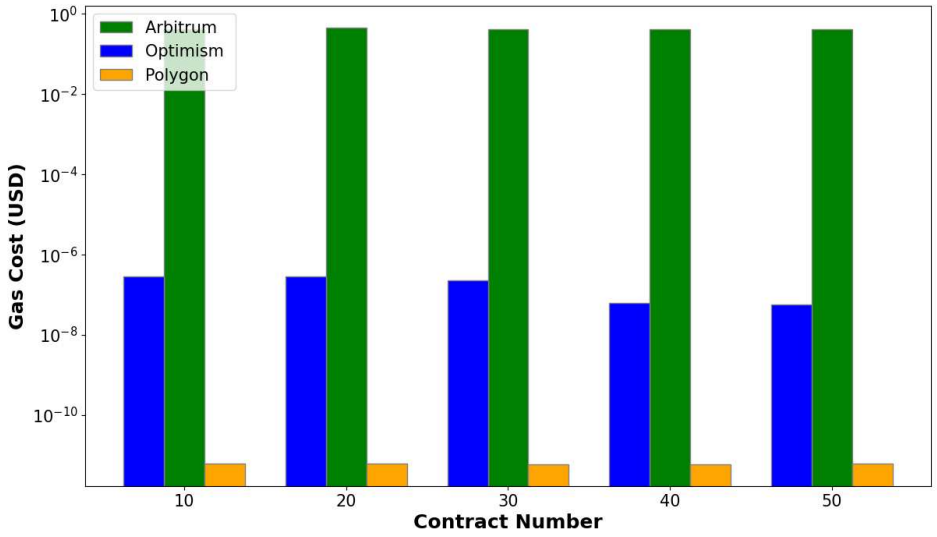

Figure 4.15: PHI Sharing - Smart Contract Deployment USD Cost [3]

reading time, enabling real-time access to block data. The system continuously synchronizes with the blockchain network to update the ledger data. Providers can maintain local nodes to enable faster authorizations.

4.6 Emergency PHI Access Policy Compliance

HIPAA, GDPR, and other data protection laws and regulations mandate patients' consent to access and share their data. They also impose compliance requirements for healthcare organizations. Non-compliance cases or failures to comply carry financial, reputational, business, and other penalties. In emergency medical situations, accessing a patient's protected health information or

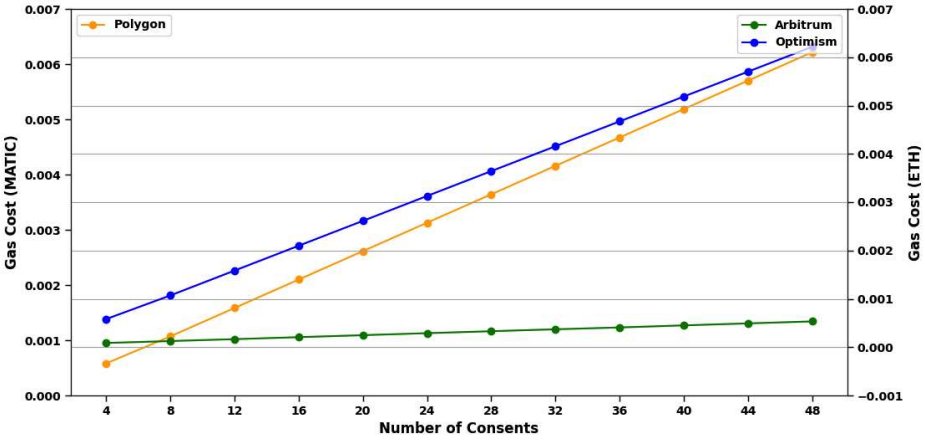

Figure 4.16: Sharing Informed Consent - Storage Gas Cost [3]

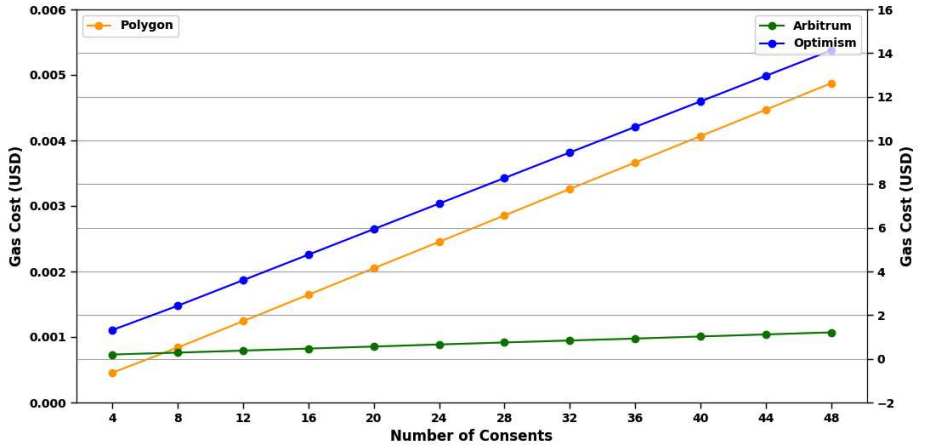

Figure 4.17: Sharing Informed Consent - Storage USD Cost [3]

records can be critical to saving lives, especially when the patient is unconscious or unable to consent. This section addresses the need for a secure, compliant, auditable system for emergency PHI access. We propose a blockchain- and smart contract-based policy compliance framework in which the emergency duty doctor requests access and must obtain approval from the senior in charge, which is recorded via multi-signature transactions. Once access is granted, the patient or their emergency contact is notified. To prevent unauthorized modifications, all actions are captured as immutable audit logs within a private blockchain network. The compliance check employs a novel *Proof of Compliance* consensus mechanism to ensure that all access requests adhere to defined policies. This framework provides transparency, accountability, and security for emergency access to PHI.

Table 4.6: Sharing Informed Consent - Writing Time to Blockchain Network [3]

Consents #	Polygon	Optimism	Arbitrum
4	6.719 Sec	8.459 Sec	6.854 Sec
8	5.961 Sec	7.785 Sec	6.068 Sec
12	5.972 Sec	7.738 Sec	6.338 Sec
16	6.309 Sec	7.762 Sec	6.063 Sec
20	6.085 Sec	8.163 Sec	6.081 Sec
24	6.015 Sec	7.482 Sec	2.476 Sec
28	10.117 Sec	7.718 Sec	6.521 Sec
32	10.041 Sec	8.268 Sec	2.451 Sec
36	10.045 Sec	7.736 Sec	6.662 Sec
40	14.039 Sec	7.797 Sec	2.458 Sec
44	10.048 Sec	7.881 Sec	6.201 Sec
48	10.138 Sec	8.971 Sec	6.174 Sec

4.6.1 Emergency PHI Access Problem Motivation

The digitization of healthcare data brings numerous benefits, including improved access to information, the ability to provide real-time and remote care, and sophisticated services [15]. It enhances patient outcomes by providing healthcare professionals with a comprehensive medical history and supporting coordinated care. Efficiency increases as administrative processes are streamlined, reducing errors and paperwork [149]. As healthcare data becomes increasingly digitized, distributed, and interactive, concerns about patient privacy and the security of healthcare information and systems are growing within the healthcare ecosystem [150]. Various security and privacy regulations are imposed worldwide to protect patient privacy and data security. *HIPAA & HITECH* (USA), *GDPR* (EU, UK), *APPs* (Australia), *PIPEDA* (Canada), *APPI* (Japan), and others are adequate data security and privacy laws. These privacy and data protection laws and regulations typically require data subjects, particularly patients in the healthcare industry, to provide consent to the processing of their data for the intended purposes. Without permission, data should not be collected, processed, used, or shared beyond the mentioned purposes. This is crucial when collecting data to avoid security and privacy violations and lawsuits.

Table 4.7: Sharing Informed Consent - Reading Time from Blockchain Network [3]

Consents #	Polygon	Optimism	Arbitrum
4	0.426 Sec	0.399 Sec	0.234 Sec
8	0.366 Sec	0.423 Sec	0.201 Sec
12	0.337 Sec	0.425 Sec	0.239 Sec
16	0.346 Sec	0.423 Sec	0.259 Sec
20	0.327 Sec	0.442 Sec	0.288 Sec
24	0.344 Sec	0.579 Sec	0.241 Sec
28	0.358 Sec	0.536 Sec	0.221 Sec
32	0.361 Sec	0.495 Sec	0.288 Sec
36	0.401 Sec	0.512 Sec	0.225 Sec
40	0.36 Sec	0.482 Sec	0.206 Sec
44	0.361 Sec	0.462 Sec	0.233 Sec
48	0.522 Sec	0.434 Sec	0.224 Sec

Healthcare providers and other users mainly access patients' healthcare data in three different circumstances: *(i)* accessed by the treatment team members for providing treatment and services and performing business operations; *(ii)* shared with others beyond the treatment team, including enhancing diagnosis and treatment plans through consultations with specialists, research and marketing endeavors, and others; *(iii)* emergency access when a patient is unconscious or insured and admitted in an emergency room in a life-and-death situation. Healthcare providers usually take consent for treatment and sharing purposes. Due to the uncertainty of the emergency, permission is not taken in advance. Also, an emergency may be far from the home or primary care provider. However, obtaining consent from an admitted or injured patient is not possible during an emergency, as the patient is unconscious or incapacitated. It is a life-and-death situation. Healthcare providers may need to bypass traditional consent processes to access PHI for life-saving treatment. The "*break glass*" protocol or emergency access control is used [151]. However, this access must comply with strict policy and regulatory requirements to protect health records, patients' privacy, and accountability.

Security and privacy policy compliance requirements for emergency access include, but are not limited to *(A)* patient must be experiencing a medical emergency and unconscious or unable to give consents to access PHI; *(B)* provider (hence known as *Requester*) must get approval from seniors (hence known as *Approver*) in charge to access PHI, *(C)* seniors in charge must determine

the emergency and give approval; *(D)* PHI access must be done from the emergency room or patient carrying ambulance; *(E)* PHI access activities (audit logs) must be stored and not modified once recorded under any situations; *(F)* compliance review or audit must be done after treatment has been done without any delay according to the applicable policies; *(G)* patient or emergency contact person must be notified about PHI access; *(H)* separation-of-duty must be maintained and enforced strictly to keep functionalities of the requester, approver, audit log unit, and auditors; *(I)* Last but not least, least privileges and need-to-know must be maintained to make sure that the requester can access no less-no more health records to provide treatment and services to contain the situation and make the patient stable. In addition to these requirements, others may be determined by the organization's business nature, regulations, legal jurisdictions, contractual obligations, and other relevant factors.

Current research and practice focus on ensuring compliance with requirements in an isolated, non-time-sensitive manner. The following issues must be addressed for compliance assurance: *(a)* requester and approver must be accountable; *(b)* audit logs must be captured as they happened and protected from modifications under any situations by any users; *(c)* enforcing separation of duty to ensure that not a single entity can manipulate every step; *(d)* maintaining least privilege and need-to-know for protecting healthcare data and patient privacy by not disclosing some PHI locked by the patient; *(e)* assuring that after accessing PHI compliance review must be done quickly to check the compliance status and inform patient or emergency contact personnel without any delays.

This section proposes a policy compliance framework for emergency PHI access to overcome the abovementioned issues and ensure streamlined policy compliance assurance. The proposed approach captures required information, stores it, and performs compliance reviews. A provider or requester submits an emergency access request for an admitted patient. Then, the senior in charge or approver evaluates the patient's condition and determines the criticality of the situation. If it is an absolute emergency, then the approver endorses the request. At this point, both the requester and approver sign the request as a multi-signature transaction using their corresponding private keys. A signed transaction is submitted to the blockchain network. Multi-signature-based blockchain

transactions ensure that no single entity can submit transactions in the network. Emergency PHI access activities are captured and stored in a private audit blockchain to provide an immutable access history for compliance review. Finally, a compliance review process is proposed using a blockchain consensus mechanism called Proof of Compliance. Where a set of independent, decentralized, and distributed audit nodes perform compliance checking using provenance information.

Blockchain technology has inherent properties: security, transparency, and immutability [152]. At its core, it is a distributed ledger technology that records transactions across multiple nodes, ensuring that registered transactions cannot be altered. This feature ensures data integrity once it has been committed to the blockchain and significantly increases the system's fault tolerance and reliability. Integrating multi-signature transactions is essential to the proposed approach, as it establishes a decentralized and immutable record of interactions [153].

To the best of our knowledge, this work is the first to capture and enforce a multi-signature-based emergency PHI access policy compliance assurance framework. This paper makes the following contributions: *(i)* Integrating patient consent into the patient-provider agreement (PPA) and enforcing it while making an emergency PHI access decision. *(ii)* Leveraging *Approver* to evaluate and determine the PHI and access level for the submitted request by the *Requester* to ensure the least privilege and need-to-know basis emergency PHI access. *(iii)* Smart contract-based separation-of-duties enforcement to ensure that *Requester*, *Approver*, *Provenance Unit*, and *Compliance Reviewer* are separate and independent entities. *(iv)* Storing approval request information in the public blockchain using a multi-signature transaction scheme. Thus, the *Requester* and *Approver* cannot deny their actions, making them accountable for assuring compliance. *(v)* Implementing audit log provenance using a private blockchain to provide immutable PHI emergency access activity data. *(vi)* Performing compliance review using a decentralized and distributed consensus mechanism called *Proof of Compliance* to determine the compliance status of every emergency PHI access. *(vii)* Conducting extensive experimental evaluations for the proposed approach on required smart contract deployment, PPA integrity, and informed consent storage and retrieval. *(viii)* Last but not least, performing and analyzing the gas costs, in token and USD, for informed consent and other

required smart contract deployment, storing PPA integrity, and informed consent. Additionally, we are analyzing the time requirements for writing and reading data to/from the blockchain network.

4.6.2 Emergency PHI Access - Proposed Approach Overview

The primary goal is to enforce the necessary consents and policies for emergency access and to capture essential PHI access activity to verify compliance with security and privacy requirements. Proper policy enforcement is crucial for ensuring compliance with provenance records and conducting timely compliance reviews, thereby maintaining a secure and compliant system. For enforcement, this paper considers patient-informed consent, in which the patient locks PHI to restrict access during an emergency. This research leverages multi-signature-based approval of access requests to ensure that PHI is not accessed unnecessarily. The emergency PHI access activities are captured and recorded in a private blockchain network as audit logs to provide provenance and reconstruct events that reflect their actual occurrence. Finally, a blockchain consensus mechanism, *Proof of Compliance*, is employed to examine enforcement actions against the applied policy and informed consent, using provenance data to verify and certify the compliance status.

4.6.3 Emergency Informed Consent (EIC) Structure

Before approval, patients must be informed about the emergency informed consent, specifically which PHI must be restricted from access. Figure 4.18 shows the EIC conceptual structure. Emergency informed consent is formally composed of two tuples:

$$EIC = (PHI, LS)$$

satisfying the following requirements:

(a) PHI is a finite set, d number, of health records denoted by $\{phi_1, phi_2, phi_3, ...phi_d\}$. It is a digital version of a patient's medical history maintained by healthcare providers over time. Classified as protected health information (PHI), it contains sensitive patient details that must be safeguarded against unauthorized access, disclosure, and sharing. Table 4.1 illustrates

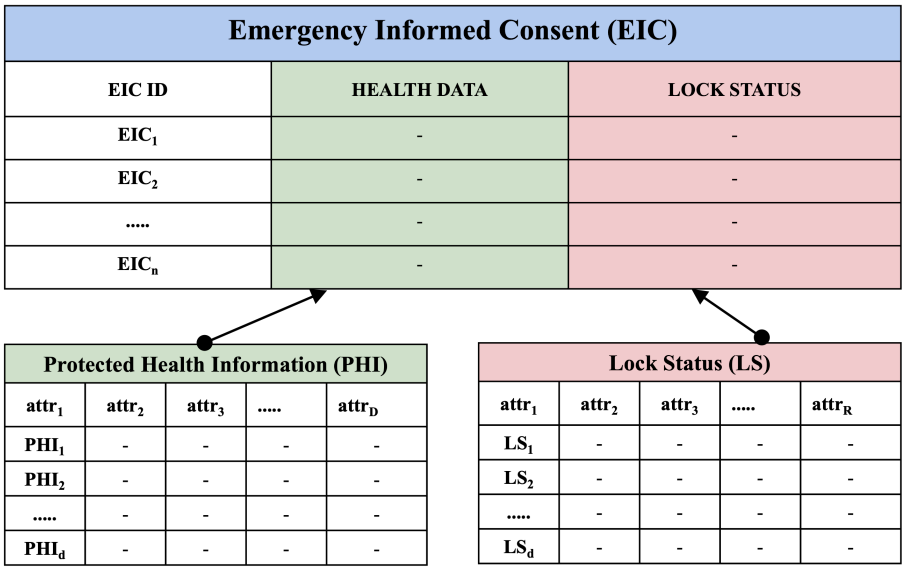

Figure 4.18: Emergency Informed Consent (EIC) Structure [10]

ten types of PHI considered for each patient, including PHI ID, name, and description. In emergencies, healthcare providers access these records to deliver life-saving treatments.

(b) LS is the lock status of the intended PHI with two values: *Locked* and *Unlocked*. A finite set of lock statuses, a d number, can be denoted as $\{ls_1, ls_2, ls_3, ...ls_d\}$. The *Locked* status indicates the PHI cannot be accessed at any moment under any circumstances. The providers cannot access *Locked* PHI during an emergency. At the same time, PHI can be accessed during an emergency if the lock status is *Unlocked*. The patient must consult with the corresponding providers before locking PHI. It should not impose any burden on providing life-saving treatment during an emergency.

There is a one-to-one mapping between each PHI and its lock status: $(phi_1, ls_1), \ldots, (phi_d, ls_d)$. This mapping ensures that patient privacy is respected and health records security is maintained during emergency access.

4.6.4 Emergency Informed Consent - Smart Contract Deployment

Once a *PPA* is established and stored in the repository, all components of the emergency-informed consent are transformed into smart contracts and deployed on the blockchain network. Figure 4.19 illustrates this *EIC* smart contract deployment process. The *Smart Contract Deployment*

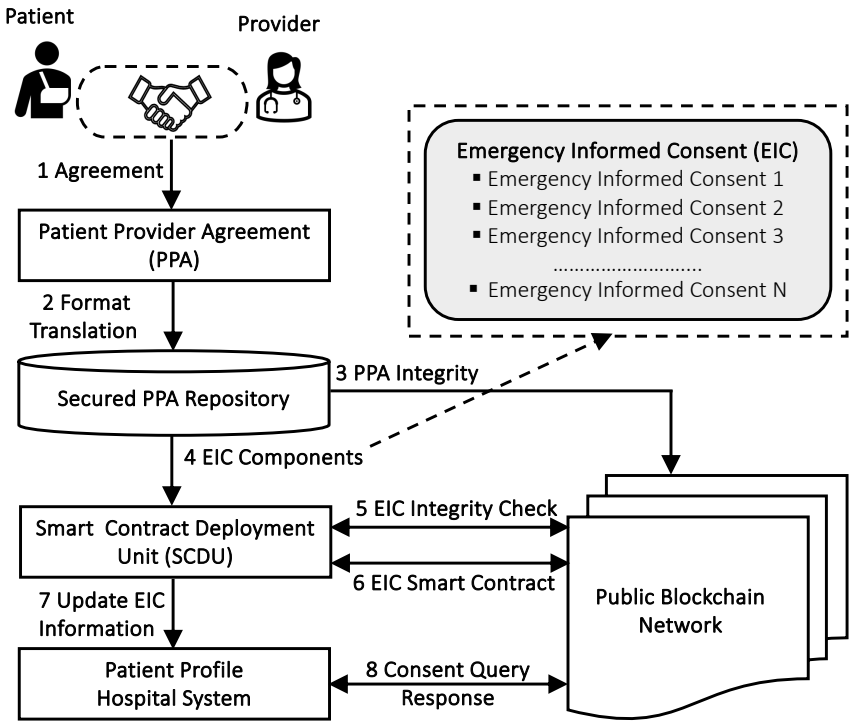

Figure 4.19: Emergency Informed Consent (EIC) Smart Contract Deployment Process [10]

Unit first retrieves all informed consent components from the *PPA*, as described in *Step 4*. In *Step 5*, the integrity of these components is verified to ensure that no intentional or accidental modifications have occurred. As a trusted, secure entity, the *SCDU* guarantees that any alteration to the consent components would invalidate the deployment process.

Once the components are verified as authentic and unmodified, the *SCDU* generates and deploys the corresponding smart contracts onto the blockchain network in *Step 6*. It then updates both the patient profile and the hospital system in *Step 7*. Finally, in *Step 8*, authorized users with appropriate credentials can directly query the blockchain network to obtain informed consent responses. This smart contract-based mechanism provides an automated and trustworthy framework that preserves the integrity, accountability, and immutability of deployed consents. After integration into the blockchain as smart contracts, consent rules cannot be altered, thereby preventing unauthorized modifications. The authorization module subsequently interacts with these smart contracts, combining their outputs with other system components to support emergency PHI access decisions.

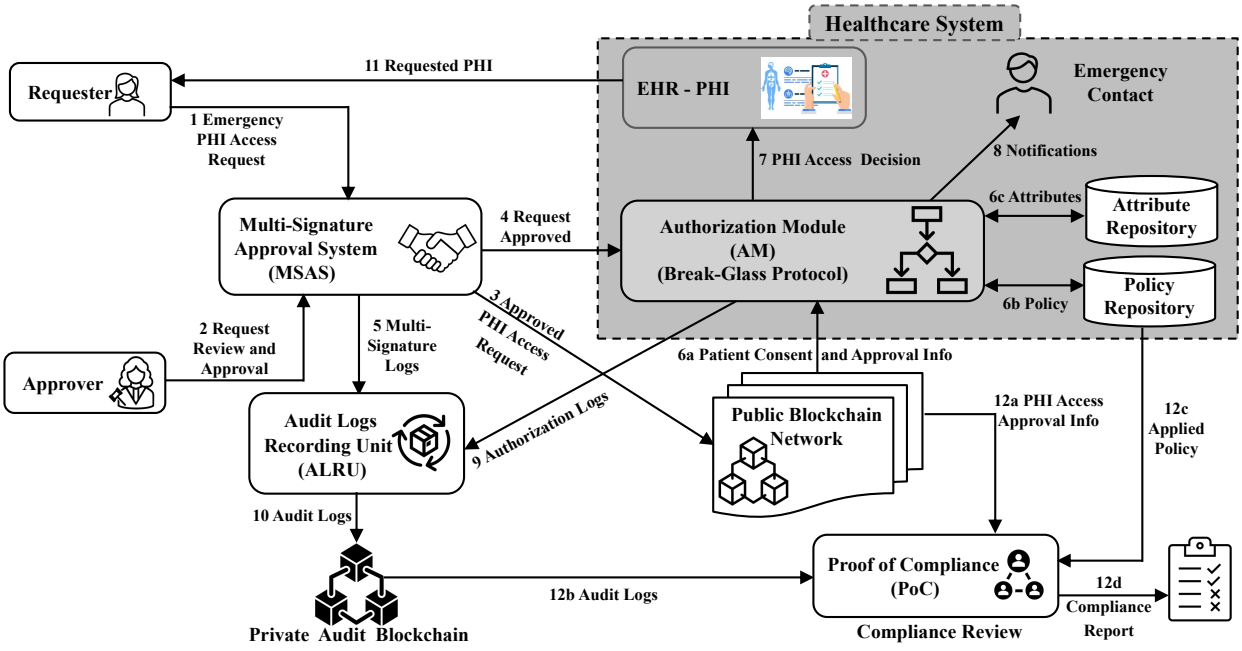

Figure 4.20: Proposed Emergency PHI Access Policy Compliance Assurance Framework [10]

4.6.5 Emergency PHI Access - Authorization Process

Consent enforcement ensures that related consents are executed when making decisions regarding emergency PHI access requests. All consents are stored on the public blockchain network as smart contracts and cannot be enforced until invoked. The *Authorization Module (AM)*, like the *Break-Glass Protocol*, considers emergency-informed consent in accordance with applicable policy and required attributes when making decisions. The attributes may be subject, object, operation, or environmental. The *Requester* must provide the necessary credentials for identification and authentication. Figure 4.20 illustrates the enforcement of informed consent for the emergency PHI access authorization and policy compliance assurance framework.

The *Requester* submits an emergency PHI access request to the *Approver* in *Step 1*. The *Approver* evaluates and determines the urgency of the admitted patient. Then, the *Approver* approves the access requests through the *Multi-Signature Approval System (MSAS)* in *Step 2*. Both *Approver* and *Requester* use their private keys to sign the transaction. In *Step 3*, the signed request is submitted to public blockchain networks, such as *Ethereum*, to be added to their distributed ledgers. Later, this deployed transaction serves as a source of truth, holding the signers accountable. In *Step 4*,

the approved request is forwarded to an emergency *AM*, such as the *Break-Glass Protocol*, for a PHI access authorization decision. The *AM* queries the blockchain network via the corresponding smart contract to obtain emergency-informed consent information and signed request-approval transactions for the submitted access request in *Step 6a*. It also makes queries for requests related to applicable policies and required attributes in *Steps 6b* and *6c*.

After evaluating, it makes an authorization decision and sends it to the *EHR* in *Step 7* and a notification to the patient's emergency contact in *Step 8*. If the access request is approved, the intended PHI is delivered to the *Requester* in *Step 11*. The audit logs recording unit, *ALRU*, collects logs from the *MSAS* in *Step 5* and from the *AM* in *Step 9*. It combines logs and stores them as audit logs in *Step 6c* in Private Audit Blockchain. Chapter 5 discusses block structure and others. The compliance review is done in *Steps 12a, 12b*, and *12c* by the *Proof of Compliance* consensus mechanism. Compliance status reports are produced in *Step 12d*. Chapter 6 discusses the required mechanism. For this study, it is assumed that the authorization module remains uncompromised and untampered with. It serves as the reference monitor for access decisions and must be tamper-proof [147]. Also, the communication channel between *AM* and the smart contract access points or apps is secured from malicious users.

4.6.6 Separation-of-Duty (SoD) Enforcement

There are four significant actors in the proposed approach: (i) the *Requester* who submits the request to access patient data; (ii) the *Approver* who evaluates the situation and determines the level of access required by the *Requester*; (iii) the *Provenance Unit* who maintains all audit logs and applied policies; and (iv) the *Compliance Reviewer* who performs compliance checking to determine the compliance status for every emergency access. These four actors must be distinct entities. No one entity should perform more than one task. Figure 4.21 depicts the *SoD* requirements for emergency PHI access compliance. This proposed approach delegates smart contracts to enforce the separation of duties among those entities to avoid conflicts of interest.

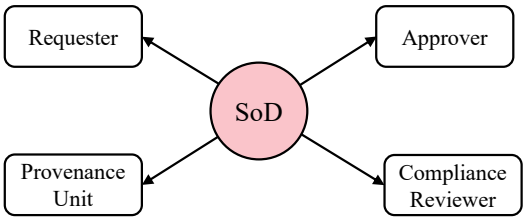

Figure 4.21: SoD Requirements [10]

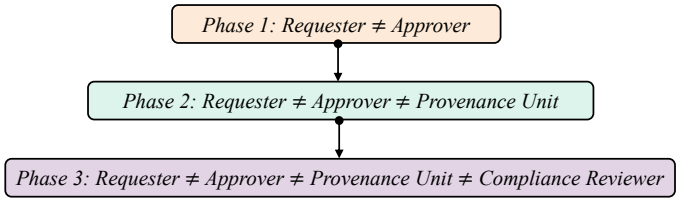

Figure 4.22: Proposed SoD Enforcement [10]

Figure 4.22 shows the *SoD* enforcement approach for the entities that must be separated for various phases. In *Phase 1*, the *Requster* and *Approver* must be different users. The *MSAS* checks and enforces this condition during the request approval process by the *Approver*, as shown in Figure 4.20. In the next *Phase 2*, it is ensured that the *Provenance Unit* is different from the *Requster* and *Approver*. The *ALRU* ensures that while collecting and storing audit logs in the private audit blockchain. Finally, it is ensured that the *Compliance Reviewer* is a separate entity from the *Requster*, *Approver*, and *Provenance Unit* (*Phase 3*). The proposed *Proof of Compliance* maintains the *Phase 3* conditions while performing the compliance review.

4.6.7 Emergency PHI Access Approval

After submitting the emergency access request, the *Approver* evaluates the situation and makes the decision. If conditions demand, the submitted request is approved and forwarded to the authorization module for the final PHI access decision. Both the request and the approval are signed by the *Requester* and *Approver* using their private keys or wallets. The signed transaction is submitted and recorded on the public blockchain to provide an unaltered source of truth for emergency PHI access compliance reviews. This is done through the multi-signature scheme of blockchain technology [153]. Due to the cryptographic properties, both *Requester* and *Approver* cannot deny their actions regarding PHI access.

4.6.8 Emergency PHI Access - Experimental Evaluation

The *Ethereum Virtual Machine (EVM)*-based blockchains are chosen for the proposed approach experiments. It offers a Turing-complete smart contract language, *Solidity*, which enables the

implementation of our model's logic. We developed smart contracts for storing and retrieving informed consent, testing them on test networks: *Ethereum* and *Optimism* to ensure reliability before deployment. Since smart contracts, once deployed, are immutable and errors can incur financial and reputational costs, rigorous testing on these networks is crucial. Ethereum's *Remote Procedure Call (RPC) API* is used to deploy smart contracts on these test networks [129]. Utilizing public *RPC* eliminates the need to maintain a blockchain node for contract interaction, assuming minimal resource usage (*CPU, HDD, Bandwidth*) on the local machine. *Faucet ETH* serves as gas to authorize transactions using the *Metamask* digital wallet [85].

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 in which participants are limited to organizations. This doesn't instill public trust. To avoid this, block IDs and hashes are stored on a public network, such as *Ethereum*, to ensure integrity. Figure 4.23 shows the block integrity storage cost in tokens for two public blockchain networks: *Ethereum and Optimism*. The *USD* costs are depicted in Figure 4.23. *Ethereum* is Layer 1, and the *Optimism* is *Layer 2* [63, 154]. *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, while storing the proof on it. The *Layer 2* solution processes more transactions per second, reducing transaction costs and shortening confirmation times.

Multi-Signature Transaction Cost

The two entities must sign every access request. It costs for each multi-signature operation. Figure 4.24 shows the costs of the *Ethereum* and *Optimism* blockchain networks. The prices fluctuate significantly, with a maximum of $18.26 and a minimum of $1.41 for *Ethereum*, as noted

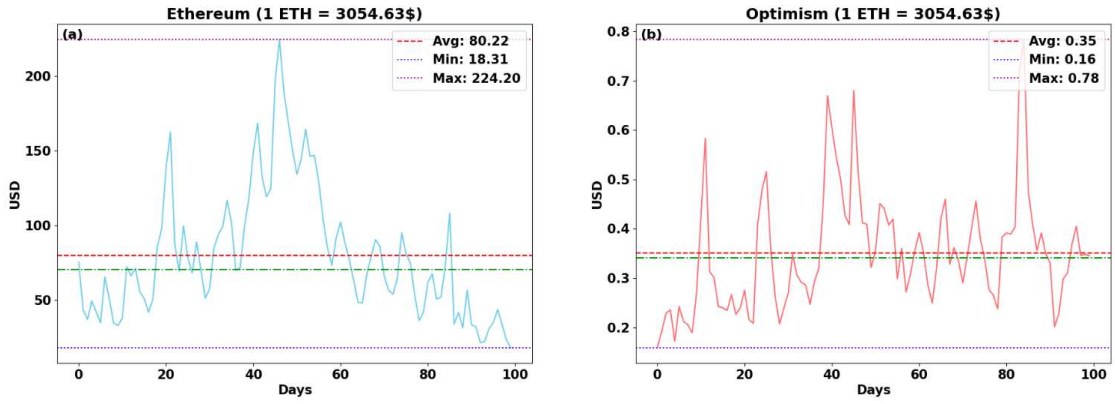

Figure 4.23: Emergency PHI Access - Smart Contract Deployment Cost [10]

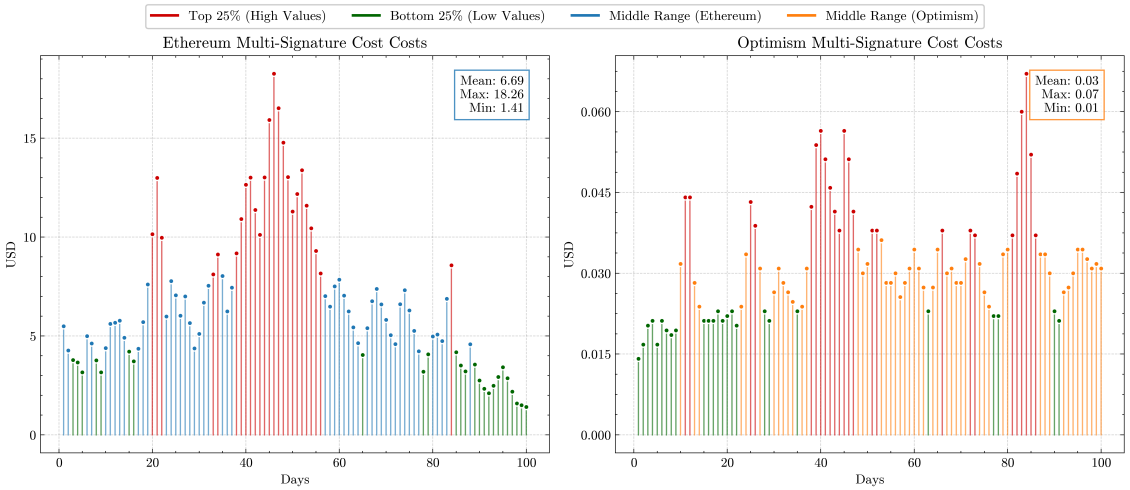

Figure 4.24: Emergency PHI Access - Multi-Signature Cost [10]

in Figure 4.24a. The average transaction cost over the 100 days is approximately $6.69. The graph shows several spikes, suggesting periods of high gas prices, possibly due to network congestion. Figure 4.24b shows the cost for *Optimism*, which is lower than on *Ethereum*, with values ranging from $0.068 to $0.013. The average cost is much lower at $0.03.

Time Requirement

Blockchain-based applications have specific requirements for block data writing and reading times. Writing time includes smart contract deployment and data addition. Table 4.8 shows the writing time for various emergency informed consent numbers for the test networks. The reading time is the time required to fetch data from the block in the blockchain ledger. All the read calls of

Table 4.8: Emergency Informed Consent - Writing Time to Blockchain Network [10]

Consents #	Polygon	Optimism	Arbitrum
4	5.256 Sec	8.167 Sec	4.519 Sec
8	6.329 Sec	8.926 Sec	6.713 Sec
12	6.653 Sec	7.156 Sec	6.907 Sec
16	5.923 Sec	7.692 Sec	4.683 Sec
20	7.465 Sec	8.426 Sec	6.651 Sec
24	5.562 Sec	7.318 Sec	6.098 Sec
28	10.927 Sec	8.925 Sec	2.142 Sec
32	10.518 Sec	8.145 Sec	4.782 Sec
36	10.637 Sec	7.562 Sec	6.872 Sec
40	11.268 Sec	7.498 Sec	4.329 Sec
44	12.519 Sec	7.387 Sec	7.602 Sec
48	13.876 Sec	8.156 Sec	5.274 Sec

smart contracts are gas-free. Table 4.9 shows the test network's reading time for various emergency informed consent numbers. The same smart contracts and consents are used for all test networks. Maintaining a node locally reduces network reading time, enabling real-time access to block data. The system continuously synchronizes with the blockchain network to update the ledger data. Providers can maintain local nodes to enable faster authorizations.

4.7 Informed Consent Management and Administration

This section briefly explains the operations involved in consent administration, including consent creation, alteration, termination, expiration, and archiving. Ensuring these operations are carried out without introducing privilege conflicts, leakages, or incomplete treatment teams is crucial. The most important aspect of these operations is to ensure that they do not disrupt the treatment process. For example, suppose consent from a pharmacy agent is withdrawn. In that case, the agent cannot access or process the patient's prescription to provide medications, which can cause delays in treatment and ultimately lead to life-threatening consequences. The consent owner or patient must invoke consent modification and termination functions. Additionally, consent expiration and archiving operations should be performed automatically as default functions when the conditions are met.

Table 4.9: Emergency Informed Consent - Reading Time from Blockchain Network [10]

Consents #	Polygon	Optimism	Arbitrum
4	0.357 Sec	0.378 Sec	0.265 Sec
8	0.352 Sec	0.329 Sec	0.231 Sec
12	0.467 Sec	0.398 Sec	0.276 Sec
16	0.394 Sec	0.571 Sec	0.246 Sec
20	0.331 Sec	0.603 Sec	0.276 Sec
24	0.354 Sec	0.613 Sec	0.215 Sec
28	0.329 Sec	0.423 Sec	0.234 Sec
32	0.426 Sec	0.612 Sec	0.247 Sec
36	0.353 Sec	0.376 Sec	0.265 Sec
40	0.436 Sec	0.602 Sec	0.291 Sec
44	0.524 Sec	0.421 Sec	0.221 Sec
48	0.462 Sec	0.342 Sec	0.237 Sec

Consent management is a complex, multi-step process that requires careful planning to ensure efficiency and meet all relevant standards and requirements.

4.7.1 Informed Consent Creation

This process involves generating new consent, with complete details and functionalities outlined in Subsections (4.4.2, 4.5.4, and 4.6.4), including the necessary components and their interplay. New consents can be formulated either during or after the *Patient-Provider Agreement* is established to accommodate the addition of new members to the treatment team. However, integrating new consents may lead to conflicts with existing ones, such as an incomplete treatment team. Therefore, a thorough check is required to avoid conflicts or issues before adding consent to the patient's smart contract. After successful verification, the consent is deployed as a smart contract. The procedure is encapsulated in *Algorithm 3*, which details the sequential steps required for successful consent creation.

4.7.2 Informed Consent Alteration

There are times when it's necessary to update a consent for various reasons, such as correcting errors, modifying current users, objects, or conditions, adding new users, entities, or conditions, dropping users, objects, or conditions, and other similar purposes. The old consent is added to

Algorithm 3: Informed Consent Creation [11]

Input: (i) $\mathbb{IC}_{New}$: New Informed Consent, (ii) $\mathbb{R}_{IC}$: Informed Consent Repository
Output: Success or failure status

Consent Creation
if *($\mathbb{R}_{IC}$ + $\mathbb{IC}_{New}$) contains no conflicts* **then**
    /* conflicts mean incomplete treatment team or process, leakage/contradictions */
    if $\mathbb{R}_{IC} \leftarrow (\mathbb{R}_{IC} + \mathbb{IC}_{New})$ == *True* **then**
        return *success:* $\mathbb{IC}_{New}$ *is added to* $\mathbb{R}_{IC}$
        /* $\mathbb{IC}_{New}$ is ready to be executed for authorizations */
    else
        return *error:* $\mathbb{IC}_{New}$ *is not added to* $\mathbb{R}_{IC}$
        /* $\mathbb{IC}_{New}$ must be modified and tested to be added to $\mathbb{R}_{IC}$ */
    end if
else
    return *error: modify/update* $\mathbb{IC}_{New}$
    /* avoid leakage/contradictions */
end if

Algorithm 4: Informed Consent Alteration [11]

Input: (i) $\mathbb{IC}_{Old}$: Old Informed Consent ID, (ii) $\mathbb{IC}_{New}$: New Informed Consent, (iii) $\mathbb{R}_{IC}$: Informed Consent Repository, (iv) $\mathbb{AR}_{IC}$: Informed Consent Archive
Output: Success or failure status

Consent Modification
if *($\mathbb{R}_{IC}$ - $\mathbb{IC}_{Old}$:) contains no conflicts* **then**
    /* leakage/contradictions */
    if *no conflict is in ($\mathbb{R}_{IC}$ - $\mathbb{IC}_{Old}$ + $\mathbb{IC}_{New}$)* **then**
        if $\mathbb{R}_{IC} \leftarrow (\mathbb{R}_{IC} + \mathbb{IC}_{New})$ == *True* **then**
            (i) do $\mathbb{AR}_{IC} \leftarrow (\mathbb{AR}_{IC} + IC_{Old})$
            (ii) add $\mathbb{IC}_{New}$ to patient profile
            return *success:* $\mathbb{IC}_{New}$ *added to* $\mathbb{R}_{IC}$
            /* $\mathbb{IC}_{Old}$ cannot be executed */
            /* $\mathbb{IC}_{New}$ can be executed now */
        else
            return *error:* $\mathbb{IC}_{New}$ *is not added to* $\mathbb{R}_{IC}$
        end if
    else
        return *error: modify/update* $\mathbb{IC}_{New}$
        /* avoid leakage/contradictions */
    end if
else
    return *error:* $\mathbb{IC}_{Old}$ *: cannot be modified*
    /* avoid leakage/contradictions */
end if

the consent archive if any modification occurs. The complete process is described in *Algorithm 4*, including all necessary components and operations. When obtaining or providing consent, there's a risk of inadvertently introducing errors that could lead to unwanted events, including security incidents. Once a mistake is identified, it's crucial to resolve it immediately to avoid adverse incidents or PHI disclosures.

Algorithm 5: Informed Consent Termination [11]

Input: (i) $\mathbb{ID}_{IC}$: Informed Consent ID, (ii) $\mathbb{R}_{IC}$: Informed Consent Repository, (iii) $\mathbb{AR}_{IC}$: Informed Consent Archive
Output: Success or error status

if $\mathbb{ID}_{IC}$ *is in* $\mathbb{R}_{IC}$ **then**
  if *no conflict is in* $\mathbb{R}_{IC} - \mathbb{ID}_{IC}$ **then**
    (i) do $\mathbb{R}_{IC} \leftarrow (\mathbb{R}_{IC} - \mathbb{ID}_{IC})$
    /* delete selected informed consent from repository */
    (ii) do $\mathbb{AR}_{IC} \leftarrow (\mathbb{AR}_{IC} + \mathbb{ID}_{IC})$
    /* add deleted informed consent to archive */
    return *success:* $\mathbb{ID}_{IC}$ is terminated from $\mathbb{R}_{IC}$
    /* $\mathbb{IC}_{Old}$ cannot be executed */
  else
    return *error*
    /* $\mathbb{R}_{IC} - \mathbb{ID}_{IC}$ contains conflict */
  end if
else
  return *error*
  /* $\mathbb{ID}_{IC}$ does not exist in $\mathbb{R}_{IC}$ */
end if

4.7.3 Informed Consent Termination

Consent withdrawal occurs when patients decide to halt their data sharing. Additionally, if consent is erroneously assigned or contains onerous conditions, it can be rescinded by the patient or the overseeing hospital authority. Upon revoking consent, it is imperative to inform all relevant parties. The revoked consent is then documented in an archive to address any subsequent legal or regulatory inquiries. *Algorithm 5* shows the step-by-step instructions for the termination operation. It's important to note that if the revoked consent is critical to ongoing treatment, its removal could result in severe consequences, including disruptions to care, medication availability, or services.

4.7.4 Informed Consent Expiration

Consent may be invalidated if predefined conditions are not met, such as specific dates or access limits. For instance, if a doctor is granted consent to access a patient's data up to five times, this consent automatically expires upon the fifth access. Any attempt to access the data a sixth time would be unauthorized due to the expiration of consent. Consent conditions, including access frequency and other relevant details, are designed to remain valid. *Algorithm 6* presents a set of instructions for the expiration process from initiation to completion. The system must monitor these conditions automatically to prevent delays and oversights, ensuring efficient and accurate consent management.

Algorithm 6: Informed Consent Expiration [11]

Input: (i) $\mathbb{CON}_{IC}$: Informed Consent Conditions, (ii) $\mathbb{R}_{IC}$: Informed Consent Repository, (iii) $\mathbb{AR}_{IC}$: Informed Consent Archive
Output: Success or error status

Consent Expiration
for *con* $\leftarrow \mathbb{CON}_{IC_{Start}}$ **to** $\mathbb{CON}_{IC_{End}}$ **by** 1 **do**
  for $ic \leftarrow \mathbb{R}_{IC_{Start}}$ **to** $\mathbb{R}_{IC_{End}}$ **by** 1 **do**
    if *con is not satisfied by* ic **then**
      (i) do $\mathbb{R}_{IC} \leftarrow (\mathbb{R}_{IC} - ic)$
      /* delete expired informed consent from repository */
      (ii) do $\mathbb{AR}_{IC} \leftarrow (\mathbb{AR}_{IC} + ic)$
      /* add expired informed consent to archive */
    else
    end if
  end for
end for

Algorithm 7: Informed Consent Archiving for Alteration, Termination, and Expiration [11]

Input: (i) $\mathbb{ID}_{IC}$: Informed Consent ID, (ii) $\mathbb{R}_{IC}$: Informed Consent Repository, (iii) $\mathbb{AR}_{IC}$: Informed Consent Archive
Output: Success or error status

Consent Archiving
if *(*$\mathbb{R}_{IC}$ - $\mathbb{ID}_{IC}$ *:) contains no conflicts* **then**
  /* conflicts mean incomplete treatment team/process, leakage/contradictions */
  (i) do $\mathbb{R}_{IC} \leftarrow (\mathbb{R}_{IC} - \mathbb{ID}_{IC})$
  /* delete altered, terminated, and expired informed consent from repository */
  (ii) do $\mathbb{AR}_{IC} \leftarrow (\mathbb{AR}_{IC} + \mathbb{ID}_{IC})$
  /* add altered, terminated, and expired informed consent to archive */
  if $\mathbb{R}_{IC} \leftarrow (\mathbb{R}_{IC} - \mathbb{ID}_{IC})$ && $\mathbb{AR}_{IC} \leftarrow (\mathbb{AR}_{IC} + \mathbb{ID}_{IC})$ == *True* **then**
    return *success:* $\mathbb{ID}_{IC}$ *removed from* $\mathbb{R}_{IC}$ *and added to* $\mathbb{AR}_{IC}$
    /* $\mathbb{ID}_{IC}$ cannot be executed for authorization */
  else
    return *error:* $\mathbb{IC}_{New}$ *is not added to* $\mathbb{AR}_{IC}$
  end if
else
  return *error:* $\mathbb{ID}_{IC}$ *: cannot be removed*
  /* avoid leakage/contradictions */
end if

4.7.5 Informed Consent Archiving

This procedure moves modified, withdrawn, and expired consents into a read-only archive or repository. It ensures that no consent within this database remains active and cannot be used to authorize access to protected health information. The archive's primary objective is to maintain a record of consents to address legal or regulatory queries and facilitate verification of policy compliance, given that certain operations may have been conducted under these consents. Furthermore, it allows patients to view all their historical consents, including any altered, revoked, or expired. The consent archiving process is outlined in *Algorithm 7*.

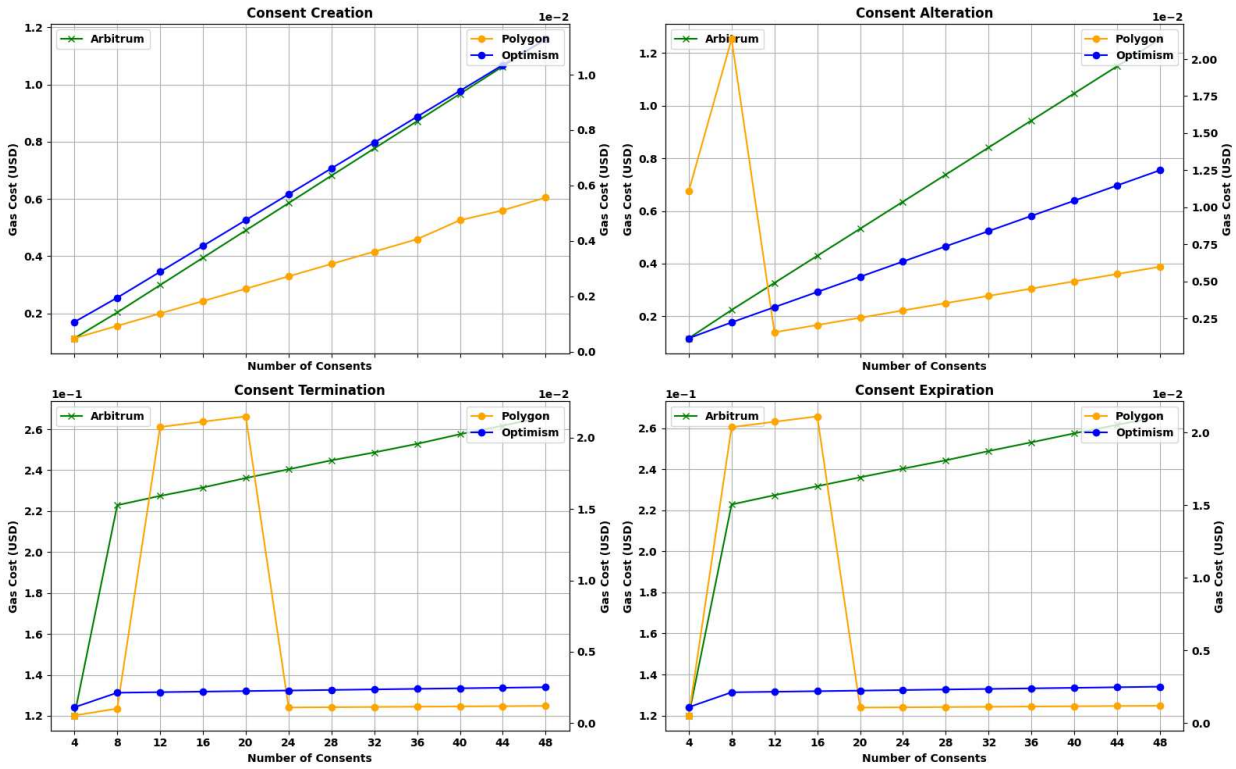

Figure 4.25: Informed Consent Creation, Alteration, Termination, and Expiration Cost [11]

4.7.6 Consent Creation, Alteration, Termination, and Expiration Cost

The creation operation involves writing new consents to the active consent repository. The other operations—alteration, termination, and expiration—require transferring active consents to the read-only archive, thereby changing their status from active to historical. Figure 4.25 illustrates the variation in transaction fees for different operations as the number of consents increases on three test networks. The volatility observed in these graphs can be attributed to network congestion, yet the price differences remain minimal. There is a noticeable, gradual increase in cost correlating with the rise in consent. Using scientific notation on the graph's scales facilitates uniform axis labeling. It provides a coherent point of comparison for vastly different values, with the power denoted at the top for reference.

4.7.7 Consent Administration Operation Time Requirement

Writing time includes smart contract deployment and adding data. A new block is added to the Ethereum main network every 12 seconds on average, ideal for the proposed purposes [131]. So long as there is sufficient space in new blocks, a new transaction would take, on average, no more than 12 seconds. If block congestion occurs, the time required for a transaction to be included in a block may increase. However, users can influence this by paying more gas for faster block confirmation. Given that users may artificially extend the confirmation time of their transactions, this could lead to discrepancies. Table 4.10 depicts the writing-time consent administration operations for the same test networks: alteration, termination, and expiration. These operations require moving consents from the active repository to a read-only archive. In both tables, *Arbitrum* requires less time than the other two networks. This is because of the sequencer design and network congestion management [132]. The same smart contracts and consents are used for all test networks.

The reading time is the time required to retrieve data from a block in the blockchain ledger. All the read calls of smart contracts are gas-free. The reading time for consent administration operations is tabulated in Table 4.11. Maintaining a node locally reduces the time required to read from the network, enabling real-time access to block data. The system continuously synchronizes with the blockchain network to update the ledger data. Hospital authorities can maintain local nodes to expedite authorization decisions.

4.8 Consent Services

The consent service provides patients with concise, clear, and consistent insights into both the given and executed consent, delivered in real time and in an informative manner. Patients need to know to whom they have provided consent, for what specific purposes, involving which resources, and under which conditions. Furthermore, patients should clearly understand how their consent is obtained, including details such as who performs which operations and when. To ensure transparency and accountability, the service provides various assurances regarding the consent provided and the actions taken. This section explores the consent services tailored for patients

Table 4.10: Writing Time for Informed Consent Administration Operation in Seconds [11]

Consents	Alteration Operation			Termination Operation			Expiration Operation		
	Polygon	Optimism	Arbitrum	Polygon	Optimism	Arbitrum	Polygon	Optimism	Arbitrum
4	6.610	7.109	2.597	6.705	6.838	2.574	6.665	7.050	2.597
8	6.670	6.967	2.465	6.769	6.874	2.284	6.592	6.916	2.210
12	6.671	6.962	2.484	6.860	6.979	2.787	6.622	6.842	2.307
16	7.024	2.871	2.240	6.667	6.946	2.349	6.729	6.839	2.485
20	6.926	6.974	2.327	6.826	7.126	2.552	6.697	6.928	2.127
24	6.739	7.066	2.562	6.732	7.053	2.849	6.850	6.875	2.784
28	6.839	7.022	2.486	10.774	2.848	2.304	7.232	2.876	2.418
32	6.797	7.128	2.809	6.853	6.862	2.299	6.581	7.067	2.324
36	6.862	7.176	3.127	6.714	6.839	2.361	6.613	7.266	2.687
40	6.942	7.630	2.533	6.683	6.958	2.602	6.658	7.084	2.414
44	7.000	7.011	2.886	6.680	6.884	2.166	10.655	6.760	2.163
48	6.948	7.195	2.597	6.891	7.036	2.409	6.818	2.217	2.548

Table 4.11: Reading Time for Informed Consent Administration Operation in Seconds [11]

Consents	Alteration Operation			Termination Operation			Expiration Operation		
	Polygon	Optimism	Arbitrum	Polygon	Optimism	Arbitrum	Polygon	Optimism	Arbitrum
4	0.417	0.466	0.482	0.381	0.419	0.401	0.410	0.427	0.395
8	0.426	0.419	0.472	0.411	0.411	0.427	0.398	0.401	0.405
12	0.424	0.418	0.480	0.403	0.438	0.389	0.405	0.408	0.429
16	0.602	0.547	0.462	0.560	0.485	0.399	0.422	0.420	0.399
20	0.479	0.528	0.503	0.551	0.538	0.482	0.463	0.624	0.461
24	0.508	0.714	0.465	0.453	0.482	0.537	0.541	0.574	0.515
28	0.564	0.639	0.566	0.476	0.478	0.481	0.515	0.672	0.467
32	0.632	0.563	0.629	0.514	0.504	0.449	0.493	0.513	0.501
36	0.685	0.657	0.632	0.487	0.495	0.552	0.484	0.555	0.590
40	0.832	0.859	0.674	0.499	0.513	0.811	0.476	0.632	0.601
44	0.890	0.753	0.642	0.495	0.504	0.473	0.528	0.494	0.556
48	1.197	0.838	0.639	0.494	0.501	0.547	0.552	0.515	0.559

within the proposed system, focusing on services oriented towards users, resources, operations, and conditions [110].

Consent Services Mechanism: A consent provenance service mechanism is proposed based on graph databases, as depicted in Figure 4.26. It initiates by collecting comprehensive informed consent information, including consent, related events, execution times, and more, from the public blockchain network (*Step 1*). This data is then stored in a graph database for further processing (*Step 2*). The processing unit retrieves consent-related information upon service requests and generates detailed reports (*Step 3*). These reports provide various service orientations: *(i) user-oriented, (ii) resource-oriented, (iii) operation-oriented,* and *(iv) condition-oriented services*. A trusted and secure *API*, or *Oracle*, facilitates data acquisition from the blockchain network and subsequent storage in the graph database. This setup enables ongoing monitoring of patient-related smart contract activity on the blockchain, capturing data for processing. Utilizing a graph database facilitates consent services by effectively handling complex relationships among patients, consents,

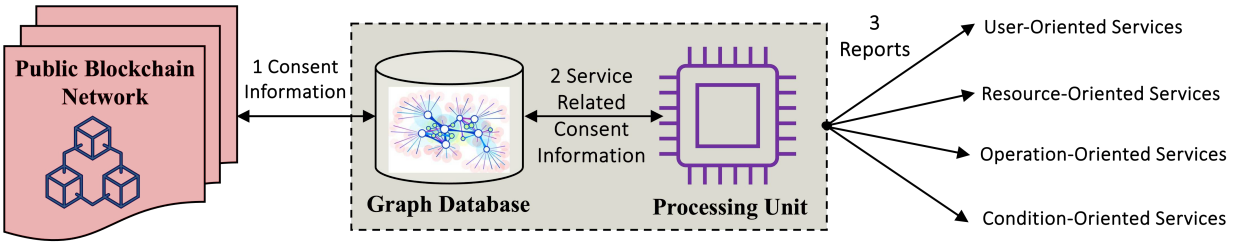

Figure 4.26: Proposed Graph Database Based Consent Service Providing Mechanism [11]

and healthcare events, enabling simplified data retrieval and insightful visualization of consent patterns [155].

User-Oriented Services: Patients can track specific users' consents, viewing a list of resources they've authorized for user operations, along with applicable conditions like access frequency and duration. This enables patients to audit any actions taken with their resources, ensuring transparency. Figure 4.27 illustrates the consents granted by patient $Jordan$ to doctor $David$, covering resources: *Visit Notes, Prescription, Radiology Lab Report, Pathology Lab Report*, and *Immunization History*, detailed with operations and conditions. Furthermore, Figure 4.28 displays the executed consents with operations, frequency, and access timing.

Resource-Oriented Services: Patients may require information on consents granted and executed for specific resources. The object-oriented consent service specifies all permissions, detailing who is authorized to perform which operations and under what conditions. Figure 4.29 presents a sample of such permissions, including the operations and conditions associated with each user and resource. Similarly, Figure 4.30 illustrates the actual usage of these permissions, showing various events with details on who performed what action, when, and other information.

Operation-Oriented Services: This service provides detailed reports on both granted and executed consents for operations such as *(i) read, (ii) write,* and *(iii) update*. While reading operations do not affect data integrity, they may compromise confidentiality in the presence of unauthorized access. Conversely, write and update operations can compromise data integrity. Ensuring that only authorized users and actions can modify data integrity is essential.

Conditions-Oriented Services: Several conditions must be met for consent to be enforced for PHI access authorizations. Patients must be assured that these conditions are thoroughly verified. In

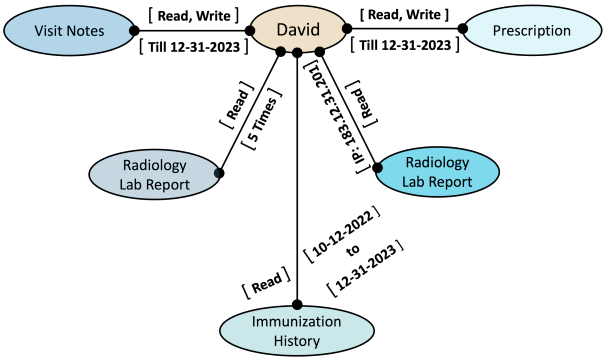

Figure 4.27: User-Oriented Given Consents [11]

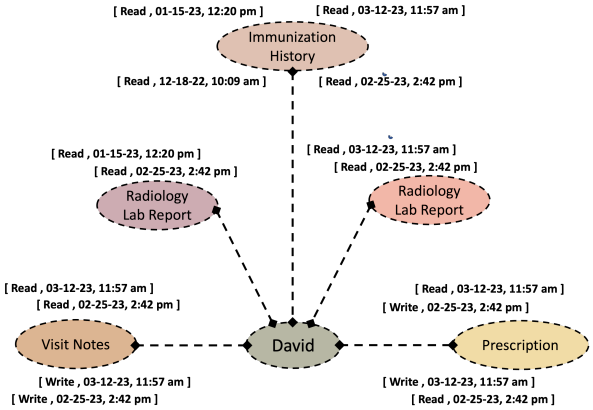

Figure 4.28: User-Oriented Executed Consents [11]

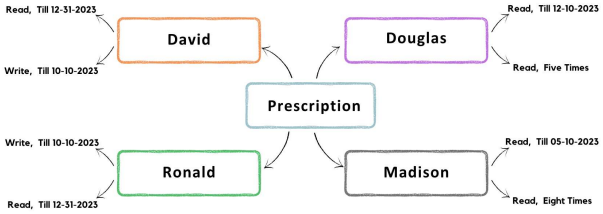

Figure 4.29: Object-Oriented Given Consents [11]

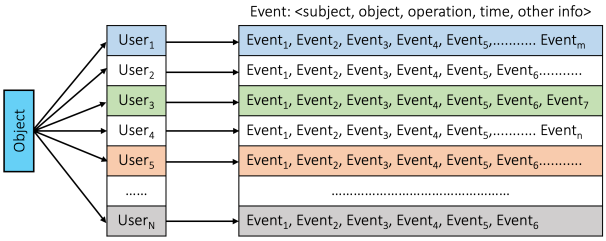

Figure 4.30: Object-Oriented Executed Consents [11]

this service mode, detailed information on both granted and executed consents is provided, with a focus on the associated conditions. This allows comprehensive visibility into how all included conditions are addressed and evaluated for making authorizations.

4.9 Contract-Based Access Control

We propose an extended version of the *Attribute-Based Access Control Model* [156] to integrate the *Patient-Provider Agreement* for authorizations with other applicable policies and attributes. The proposed model is referred to as the *Contract-based Access Control Model*. Integrating PPA into the access control model improves transparency and accountability and facilitates compliance monitoring.

Figure 4.31 contains proposed model components. The solid line with the arrow indicates a request/command, and the dotted line with the arrow means a response/feedback. The dotted line

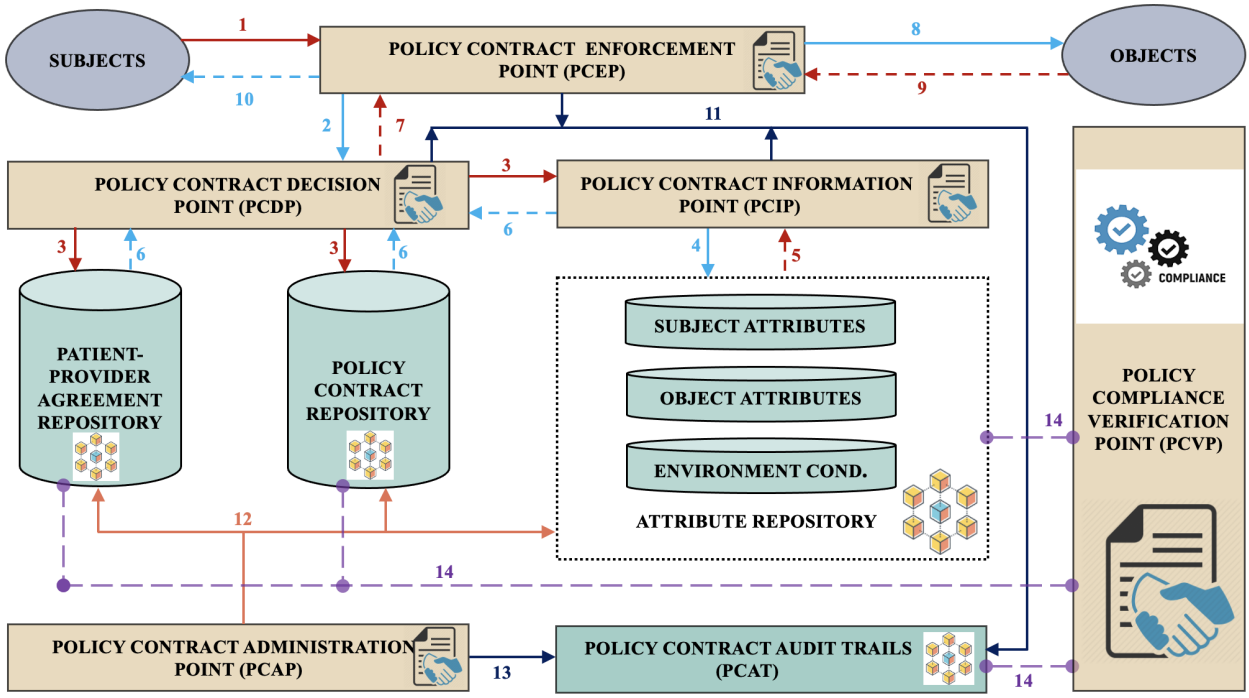

Figure 4.31: Contract-Based Access Control Model [2, 12]

with circles (purple) indicates the request/command and the response/feedback. Their interactions and descriptions are discussed below.

Subject (SB): A subject is a human user or non-person entity (NPE), such as a device that issues access requests to perform operations on objects. Subjects are assigned one or more attributes. A total m number of authorized subjects can be represented as $\{sb_1, sb_2, sb_3, sb_4, ...sb_m\}$.

Subject Attributes (SA): Subject attributes of a subject, such as a name, date of birth, home address, training record, and job function, may, individually or combined, comprise a unique identity that distinguishes that user from all others. A finite number, total n, of subject attributes is defined as $\{sa_1, sa_2, sa_3, sa_4, ...sa_n\}$.

Object (OB): An object can be a resource or requested entity and anything upon which a subject may operate, including data, applications, services, devices, and networks. A finite number, total p, of objects to be protected can be written as $\{ob_1, ob_2, ob_3, ob_4, ...ob_p\}$.

Object Attributes (OA): An object's attributes help to describe and identify it. Attributes include the object name, creator, creation time, and other relevant details. A q number of object attributes can be expressed as $\{oa_1, oa_2, oa_3, oa_4, ...oa_q\}$.

Operation (OP): An operation is an action that can be requested by any subject for any object. Only a subject can be authorized to perform a requested operation on an object if the policy allows it. A finite, total r, of actions to be performed are denoted as $\{op_1, op_2, op_3, op_4, ...op_r\}$.

Environment Condition (EC): Environmental conditions are dynamic factors, independent of subject and object, that may be used as attributes at decision time. They may include location, time, day of the week, threat level, device ID, user IP address, temperature, and other relevant details. A finite, total s, of environmental conditions can be expressed as $\{ec_1, ec_2, ec_3, ec_4, ...ec_s\}$.

Policy Contract Administration Point (PCAP): It provides functionalities for creating, storing, managing, and testing policies, PPAs, SB, OB, and EC attributes.

Attributes Repository (AR): It contains all SB, OB, and EC attributes. Before storage, *PCAP* ensures the authenticity and integrity of the attributes' sources and values, as the repository will later serve as the standard for verifying and comparing the user-provided attributes, informing authorization decisions.

Policy Contract Repository (PCR): The *PCR* contains digital policies (DPs) and metapolicies (MPs) of obligatory policies like organizational policies, regulatory agency policies, and access control policies. A policy specifies the rules for determining whether requested access should be allowed, based on the attributes of the SB, OB, and EC.

Patient-Provider Agreement Repository (PPAR): It contains all valid contracts made by patients and providers. The *Policy Contract Decision Point or PCDP* must execute *PPAs* related to an access request.

Policy Contract Decision Point (PCDP): It computes access decisions by evaluating the applicable policies from *PCR*, PPAs from *PPAR*, and attributes from *AR*.

Policy Contract Information Point (PCIP): The *PCIP* is the retrieval source of the attributes required for policy evaluation to make authorizations by *PCDP*.

Policy Contract Enforcement Point (PCEP): After making authorizations, the *PCDP* forwards decisions to *PCEP*. The *PCEP* can access protected resources and objects through a Resource

Access Point (*RAP*). Multiple *RAP* might exist, but every object is accessible only through a single *RAP*.

Policy Contract Audit Trails (PCAT): It contains the activities happening in the system and related to policy compliance. All activities other than those related to policy compliance are not required for this proposed model. The actions can be adequately reconstructed from *PCAT* to recreate the events performed by *PCEP, PCDP, PCIP, PCAP*, and *SB*.

4.9.1 Achieving Provenance and Compliance

The data access control process for provenance and compliance is summarized in Figure 4.32, and the steps are as follows:

Step 1 and 10: The subject puts the request to *PCEP* in *Step 1* with the required credentials and attributes. The subject receives the object in *Step 10* if an access request is granted. Otherwise, the subject receives the access denial decision.

Step 2 and 7: *PCEP* forwards the subject access request in *Step 2* with the credentials and attributes received from the subject to *PCDP* to make the decision. In *Step 7*, *PCEP* receives the access decision made by *PCDP*.

Step 3, 4, 5, and 6: In *Step 3*, *PCDP* requests *PCIP* to retrieve the attributes of the subject and object and environmental conditions related to the access request. *PCDP* also asks the policy repository to find the applicable policies. It also requests that the PPAR retrieve the PPAs between the patient and the provider for the access request. In *Step 5*, *PCIP* retrieves the attributes from the attribute repository. *PCDP* receives responses in *Step 6*.

Step 8 and 9: If access is granted, then *PCEP* gets the object in these steps.

Step 12: *PCAP* updates the attribute, policy, and patient-provider agreement repository.

Step 11 and 13: In *Step 11*, *PCEP*, *PCDP*, and *PCIP* record the activities as audit trails to *PCAT*. When *PCAP* updates repositories, its activities are recorded in *Step 13*.

Step 14: *PCVP* gets all required information from repositories to certify policy compliance.

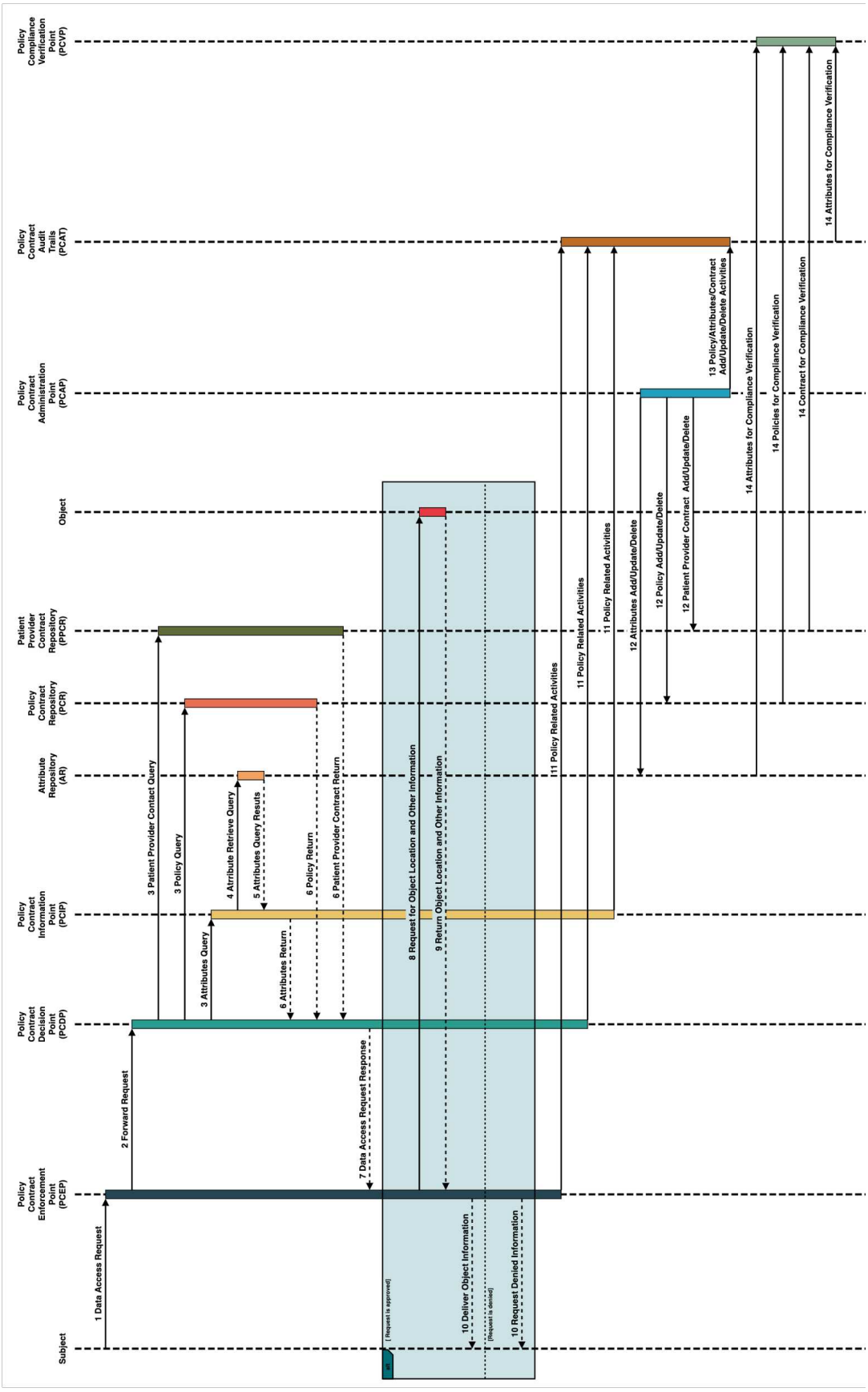

Figure 4.32: Policy Enforcement, Provenance, and Compliance Sequence Diagram [2, 12]

Chapter 5

Policy Provenance

While enforcing policy to ensure security and privacy is essential, it is equally important to maintain provenance to demonstrate policy compliance. However, policy compliance cannot be measured or validated in isolation. An independent auditor performs a policy audit to certify the policy's compliance status using available provenance data. To measure policy compliance, it is essential to maintain the following:

- Policy lineage
- Integrity of policy enforcement activities

Policy lineage comprises all policies on which the authorization module bases its authorization decisions. Enforcement integrity means the events are recorded as they happen. Provenance provides a lineage of policy enforcement activities as they are executed. This section details the mechanisms for provenance to ensure policy lineage and the integrity of audit logs.

5.1 Provenance Requirements

The main requirement for provenance is to maintain or protect data integrity. Provenance data integrity demands maintaining the data and the corresponding time information. Under any circumstances, no one should be able to modify the data or the timestamp. Policies must be preserved as they are executed to make authorization decisions, and when they are executed. Similarly, activity data or audit logs must be stored with timestamps. It is necessary to ensure that no entity can alter the data once it has been captured and recorded. There must also be a mechanism to control access to the audit log. Not every user can access the audit log. This research proposes a blockchain-based provenance mechanism for an assurance framework for healthcare security and privacy policy compliance.

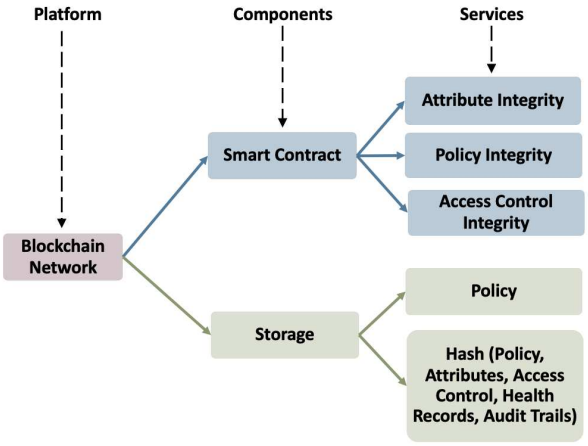

Figure 5.1: Provenance via Blockchain [12]

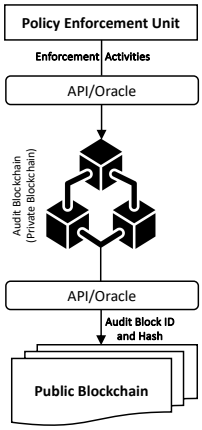

Figure 5.2: Policy Enforcement Audit Logs [2]

5.2 Provenance via Blockchain

The blockchain network acts as a platform that has two major components: *(i) smart contracts* and *(ii) storage capacity*. Figure 5.1 depicts the provenance services via the blockchain network. The smart contract provides integrity services for attributes, policies, and access control mechanisms. The blockchain network stores various types of data, including policy, policy hashes, attributes, access control, health records, audit trails, and other data. The proposed framework captures and deploys informed consent as a smart contract to the public blockchain. It serves as a patient-driven policy governing the access and sharing of protected health information. When the corresponding smart contract is executed to authorize, the network generates activity logs that are stored on the blockchain and timestamped. This provides the policy lineage used later for policy compliance review or audit.

5.3 Policy Enforcement Audit Log

An audit log records information about the user, the operation performed on which health records, the time, and, based on which policy, whether the access request was granted. It may contain additional information required for business requirements and other obligations. An audit log must be sufficient to reconstruct past events and hold users accountable for their actions. To provide audit log provenance, data integrity must be ensured so that it cannot be tampered with

by any users under any conditions. Additionally, controlling the audit log to prevent unauthorized access is necessary to ensure the security of health records and protect patient privacy.

Given the provenance requirements, we propose an audit log storage system based on a private blockchain network. The private blockchain offers the same properties as the public blockchain while providing greater control over access. A private blockchain is a setup in which a closed group of users determines the consensus mechanism and other properties, such as block structure, size, and contents. Figure 5.2 shows the structure of the policy enforcement audit log-capturing and storage unit. All policy enforcement activities are collected and stored as audit logs on the private blockchain network. The private *Ethereum* blockchain network stores audit logs.

5.3.1 Private Block Integrity on Public Blockchain

A private blockchain is a system configured for a select group of participants who establish consensus and specify features such as block structure, block size, and block contents [56]. To safeguard against intentional alteration of audit and compliance blocks, the proposed approach stores the block ID and hash as block-integrity metadata on a public blockchain, such as *Ethereum*, thereby ensuring block integrity. Figure 5.3 illustrates the process of storing block IDs and hashes 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 [157]. An *API/Oracle* is developed and deployed to store and retrieve block-integrity information in the blockchain network. Here, the *API/Oracle* is a secure, trusted, and blind entity that doesn't reveal any information to unauthorized users. Authorized users submit requests to determine the integrity status of any private block. The *API/Oracle* checks whether the requested block has been modified and returns an appropriate response.

A private blockchain network is configured, controlled, and maintained by a set of closed groups or organizations. It is possible to tamper with the ledger by manipulating the consensus mechanism and other configurations. To avoid this, the proposed approach stores the block ID and hash as block integrity metadata on a public blockchain, such as *Ethereum*, to ensure that audit logs are

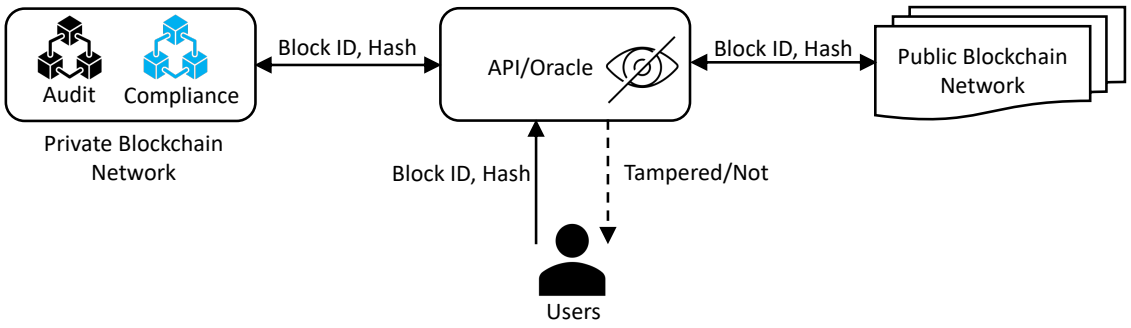

Figure 5.3: Storing Audit Block Integrity on Public Blockchain [4]

not intentionally modified. Later, authorized users can verify the integrity of the audit logs using stored block integrity from the public blockchain. An *API/Oracle* is developed and deployed to interact with a blockchain network from another one. Since smart contracts cannot communicate directly between blockchain networks. An *API/Oracle* is a trusted, blind entity that reads data from the source and transfers it to the destination without modifying or revealing it to other entities. The detailed mechanism of the *API/Oracle* is out of the scope of this paper.

5.3.2 Private Blockchain-Based Audit Log Provenance

Database management systems can store and process policy enforcement activities or audit logs. However, it is required to deploy a separate entity to ensure the integrity of the database that stores the audit logs. Data integrity is a key requirement for provenance compliance. If integrity is not maintained, the entire compliance effort is questionable and not accepted in business processes or legal matters. Depending on an external module to ensure integrity may compromise the objectives. To avoid this, we propose a private blockchain (Ethereum private or enterprise setup) instead of database systems. Since blockchain networks inherently provide integrity properties by cryptographically and chronologically binding blocks in the ledger.

In Chapter 6, a consensus mechanism, *Proof of Compliance (PoC)*, is proposed to verify the compliance status of enforcement activities. A dedicated set of auditor nodes performs compliance-checking operations in accordance with the consensus algorithm. Additionally, the existing smart contract framework can perform various operations to provide services, including all audit logs for a particular user or object, as well as compliance services. To execute the PoC consensus mechanism,

audit logs must be stored on a blockchain. Storing data on public blockchain networks is costly. Moreover, systems generate large volumes of data, and audit logs contain sensitive information about users' actions. We avoid storing audit logs on the public blockchain.

5.4 Provenance - Treatment Team PHI Access

This section describes the provenance mechanisms established to preserve the integrity of policy lineage and ensure the authenticity of audit logs related to the treatment team's access to health information. For effective and accurate policy compliance assessment, it is essential to maintain two key elements: *(i) the consent and policy lineage associated with treatment team access* and *(ii) the PHI access audit logs generated from treatment team activities*.

5.4.1 Treatment Team Consent and Policy Lineage

For treatment purposes, patient consent is collected and recorded on the blockchain via smart contracts. Once deployed, these smart contracts are immutable and cannot be altered. Each informed consent for treatment contains four key components: (i) the user ID of the individual authorized to access the patient's health records, (ii) the health record ID or Protected Health Information (PHI) ID, (iii) the permitted operations, such as reading, writing, updating, or deleting health records, and (iv) the conditions governing access to PHI, including constraints related to time, day, and location.

The blockchain ledger maintains a complete lineage of consent policies associated with the treatment team's access to health information. If a consent requires modification, a new smart contract must be deployed, and this new contract is recorded with its own timestamp. As a result, the blockchain preserves a verifiable history of changes to consent over time. This lineage information can effectively support policy compliance audits, accountability reviews, and access governance verification.

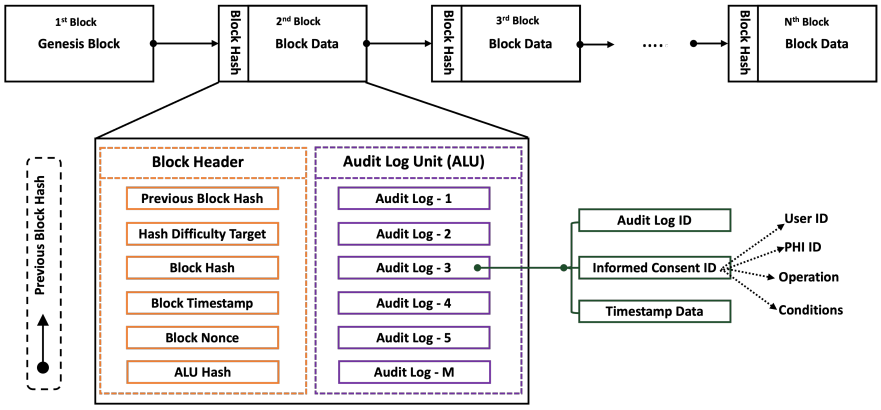

Figure 5.4: Treatment Team PHI Access Audit Log Transaction Structure

5.4.2 Treatment Team PHI Access Audit Logs

Figure 5.4 presents the block structure and components of the private audit blockchain designed to log PHI access by the treatment team. Each block comprises a header containing metadata and a data section containing audit log entries. Each audit log entry consists of three elements: *(i) audit log ID, (ii) informed consent ID,* and *(iii) timestamp data.* The audit log ID uniquely identifies each access event, whereas the informed consent ID refers to the specific consent used to authorize access to the health records. Using this informed consent ID, the associated consent attributes can be derived, including the user ID, PHI ID, operation, and access conditions. The timestamp data indicates when the health record was accessed, thereby enabling traceability and supporting subsequent audit and compliance review activities.

5.5 Provenance - PHI Sharing Beyond Treatment Team

Enforcing an applicable set of policies is crucial, but preserving data provenance to show adherence to these policies is also essential. Nevertheless, policy compliance cannot be quantified or confirmed in isolation. An independent auditor conducts a thorough policy audit to verify compliance with the policy, utilizing the available provenance data to ascertain and certify the policy's compliance status. For an accurate policy compliance assessment, two critical elements must be diligently maintained: *(i) consent and policy lineage* and *(ii) PHI sharing activity audit*

trails. This section describes the detailed provenance mechanisms for preserving the integrity of the policy lineage and ensuring the authenticity of the audit trails.

5.5.1 PHI Sharing Consent and Policy Lineage

Policy lineage involves a comprehensive record of all policies that guide the authorization module's decisions. It's a transparent and traceable record of the policy history and its application in decision-making processes. In this study, informed consent is primarily considered for decision-making. Since all consents are deployed as smart contracts, blockchain networks can create policy lineages. However, this paper does not consider other HIPAA-related policies, such as physical security, provider training, etc [142].

5.5.2 PHI Sharing Activity Audit Logs

Integrity in policy enforcement ensures that events are documented faithfully, reflecting the sequence and nature of actions taken. This authenticity is crucial for transparency and accountability. Provenance plays a key role by providing a detailed, unalterable history of policy enforcement actions as they are carried out, thereby safeguarding against record tampering. The alteration of audit trails or unauthorized access to healthcare data is strictly prohibited to maintain the sanctity of the process. Maintaining the integrity of the audit trail is essential for ensuring policy compliance. If integrity is compromised, checking compliance status to identify both compliant and non-compliant cases is questionable. The blockchain provides these requirements as ledger properties. This work adopts a private blockchain as an audit trail storage system.

Figure 5.5 illustrates the private audit blockchain's block components and structure. Each block comprises a header containing metadata and a data part that stores the audit trail data. Each audit trail has five components: *(i) audit trail ID; (ii) informed consent ID or SIC ID; (iii) honest broker ID; (iv) honest broker report;* and *(v) timestamp data.* The audit trail ID provides unique identifiers; the informed consent ID, or SIC ID, indicates the consent executed to share the intended PHI. From the SIC ID, it is possible to get the components: sender, receiver, PHI, and purpose. The honest broker ID indicates which broker certifies or attests to the intended PHI's protection

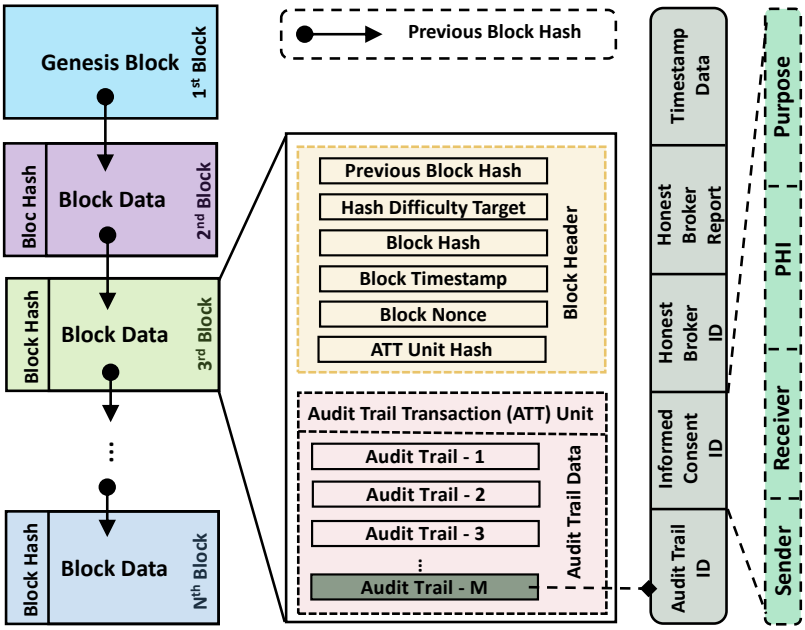

Figure 5.5: PHI Sharing Audit Log Transaction Structure [3]

status (encryption or anonymity). Finally, the timestamp indicates when the sharing authorization is completed. Steps *6a, 6b,* and *6c* in Figure 4.12 show the process of capturing audit trails from the authorization module and honest broker.

Enforcement activity data are collected and stored on a private blockchain, known as an audit blockchain, as immutable records to ensure consent provenance and maintain compliance. The private blockchain network is managed and maintained by an authority, so that read and write permissions are granted to a limited set of participants. In this case, the trust and transparency of the private blockchain are questionable. It doesn't provide a public eye to maintain trust and transparency. Storing audit trails on the public blockchain gives trust and transparency, which is another issue to consider. Firstly, audit trails contain sensitive information, such as user activities, and storing them on a public blockchain creates security and privacy concerns. Secondly, audit trails generate large volumes of data that require substantial storage on the public blockchain. This is not feasible from a business perspective, as it increases operational and treatment costs, as well as service charges.

To address the aforementioned issues, this research uses a private blockchain, known as the private audit blockchain, to store audit trail data. Then, it stores the private audit blockchain block

ID and hash as an integrity check on the public blockchain. Storing block ID and integrity incurs a small cost, providing trust and transparency. Any modifications to private audit blockchain data can be detected by comparing the block's current and stored hashes with those on the public blockchain. Figure 5.3 illustrates the relationship between private and public blockchains, specifically for storing audit block IDs and ensuring integrity in a public blockchain like Ethereum. We have configured a private blockchain using the Ethereum client [158] with the necessary smart contracts and APIs to capture and store audit-trail data on the audit blockchain.

5.6 Provenance - Emergency PHI Access

Enforcing an applicable set of policies is crucial, but preserving data provenance to show adherence to these policies is also essential. Nevertheless, policy compliance cannot be quantified or confirmed in isolation. An independent auditor conducts a thorough policy audit to verify compliance with the policy, utilizing the available provenance data to ascertain and certify the policy's compliance status. For an accurate policy compliance assessment, two critical elements must be diligently maintained: *(i) emergency PHI access request approval* and *(ii) emergency PHI access audit logs*. This section describes detailed provenance mechanisms for preserving the integrity of emergency PHI access request approvals and ensuring the authenticity of audit logs.

5.6.1 Emergency PHI Access Approval

After submitting the emergency access request, the *Approver* evaluates the situation and makes the decision. If conditions demand, the submitted request is approved and forwarded to the authorization module for the final PHI access decision. Both the request and the approval are signed by the *Requester* and *Approver* using their private keys or wallets. The signed transaction is submitted and recorded on the public blockchain to provide an unaltered source of truth for emergency PHI access compliance reviews. This is done through the multi-signature scheme of blockchain technology [153]. Due to the cryptographic properties, both *Requester* and *Approver* cannot deny their actions regarding PHI access.

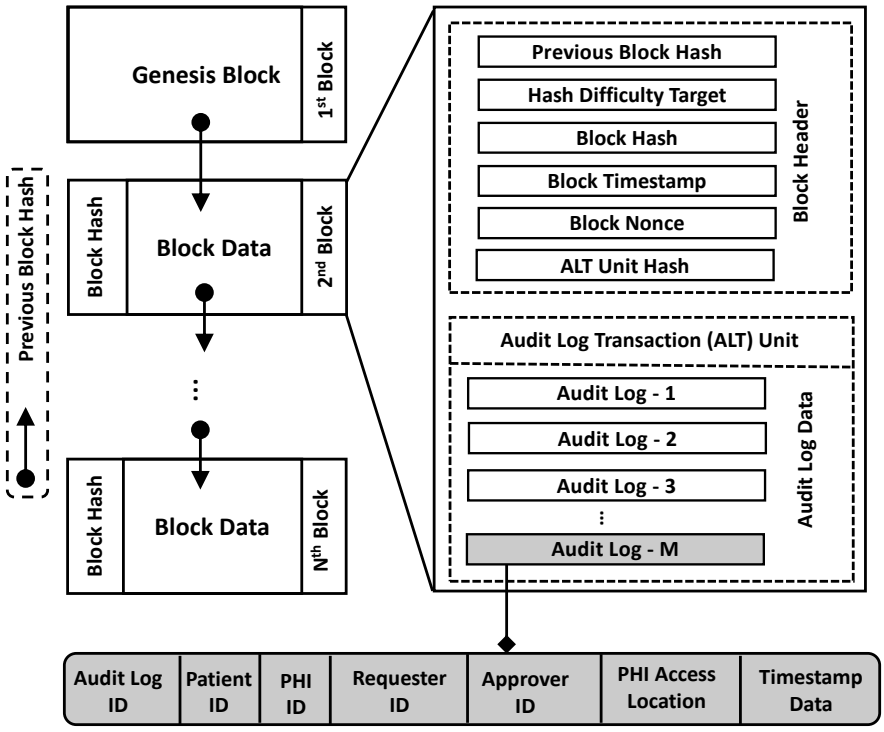

Figure 5.6: Emergency PHI Access Audit Log Structure [10]

5.6.2 Emergency PHI Access Audit Logs

Integrity in policy enforcement ensures that events are documented faithfully, reflecting the sequence and nature of actions taken. This authenticity is crucial for transparency and accountability. Provenance plays a key role by providing a detailed, unalterable history of policy enforcement actions as they are carried out, thereby safeguarding against record tampering. The alteration of audit trails or unauthorized access to healthcare data is strictly prohibited to maintain the sanctity of the process. Maintaining the integrity of the audit trail is essential for ensuring policy compliance. If integrity is compromised, checking compliance status to identify both compliant and non-compliant cases is questionable. The blockchain provides these requirements as ledger properties. This work adopts a private blockchain as an audit log storage system.

Figure 5.6 illustrates the private audit blockchain's block components and structure. Each block comprises a header containing metadata and a data part that stores the audit trail data. Each audit log has seven components: *(i) audit log ID; (ii) patient ID; (iii) PHI ID; (iv) Requester ID; (v) Approver ID; (vi) PHI access location;* and *(vii) timestamp data.*

The audit log ID uniquely identifies each access log, while the patient ID refers to the patient receiving emergency life-saving treatment. The PHI ID indicates the specific health records accessed during treatment, as depicted in Table 4.1. Patients can lock any particular health record in *EIC*. The Requester ID identifies the healthcare provider treating the admitted patient and requires access to the patient's data. The Approver ID corresponds to the person responsible for evaluating and endorsing the access request in light of the current authorization state. These access requests and endorsements are securely recorded on a public blockchain network, such as *Ethereum*, using a multi-signature process, thereby ensuring non-repudiation by the involved parties. The PHI access location identifies the physical setting, such as an emergency room or an ambulance, from which healthcare records are accessed. Finally, the timestamp denotes the time at which the access authorization is performed. Steps *5* and *9* in *Figure 4.20* show the process of capturing audit logs from the *MSAS* and *AM*. The *ALRU* stores audit logs in a private audit blockchain in *Step 10*.

5.7 PoC Transaction and Block Structure

The proposed *Proof of Compliance* or PoC mechanism retrieves audit logs from the audit blockchain, requires informed consent from the public blockchain network, and obtains 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 a private blockchain that stores audit log IDs and their 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*.

5.7.1 Audit Block Transaction Structure

An audit log indicates that a single operation has already occurred in the system. This study considers two types of audit logs, as depicted in Figure 5.7. Figure 5.7(a) shows the audit log for treatment team access. Next, Figure 5.7(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 [159, 160]. This work stores patient-informed consent for treatment in the public blockchain network. The detailed process can be investigated in [13]. 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 4.1 shows the sample health records, categorized by ID, name, and description.
- *User ID:* This unique user ID performs various operations. It is also referred to as the subject, which may be a treatment team member or any other hospital staff member. In this study, we don't consider external users to be members of the treatment team.
- *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* [3]. 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.

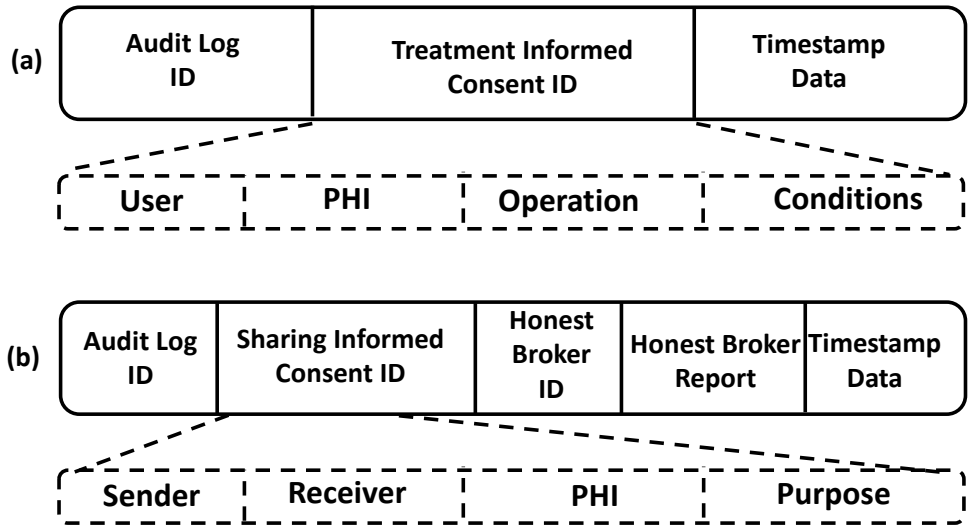

Figure 5.7: Audit Log Transaction Structure [4]

The sender can share specific healthcare data with the receiver, provided the receiver has the patient's permission. The *sender* may be a patient treatment team member or a 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 [143]. After checking, the honest broker certifies or attests to the status, which is recorded in audit trails as proof.

5.7.2 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 5.8. The network participants can determine the number of log records. If the number of records is fixed, the block size will remain constant. Otherwise, the block size would vary. This paper stores a fixed number of log records per block to keep all block sizes the same.

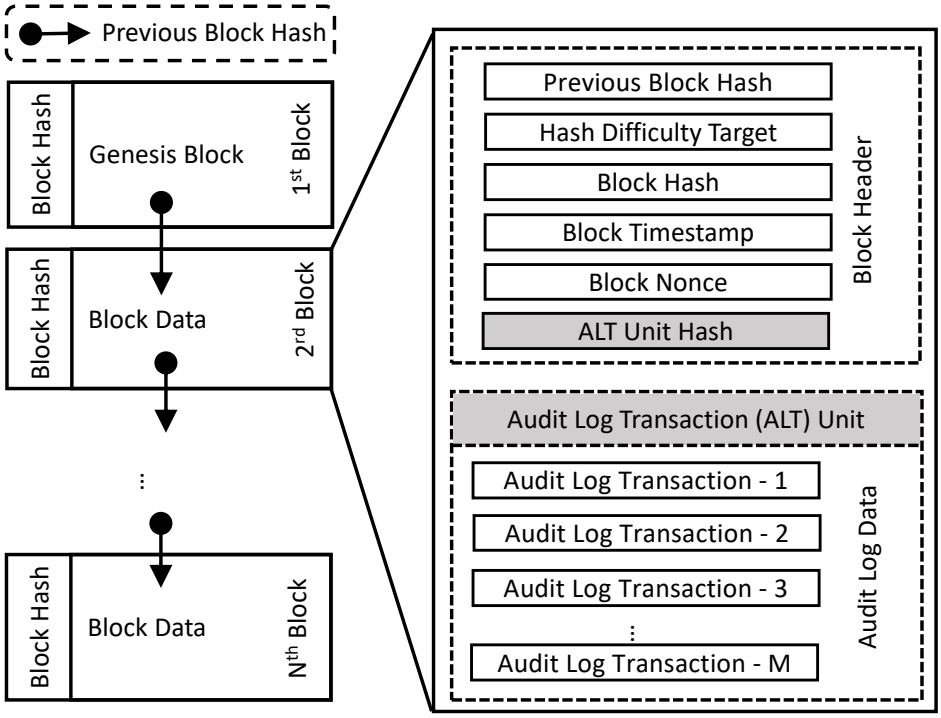

Figure 5.8: Audit Blockchain Block Structure [4]

5.7.3 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 access to or sharing of protected health information. Users access or share PHI only with patients' consent. 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 corrective actions, including employee warnings, training, transfers, and termination. Additionally, organizations should deploy new security systems or update existing ones to 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, and other relevant documents. Additionally, some auditor nodes were unable to determine the compliance status. These cases must be investigated later to resolve the issues.

5.7.4 Compliance Blockchain Block Structure

The compliance block is an organized structure designed to record the compliance status of audit logs securely. Figure 5.10 shows the compliance block structure. Each compliance block includes a unique audit log ID and a corresponding compliance status, 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.

5.8 Private Block and Audit Log Integrity Verification

Not everyone should access the audit logs, as they contain sensitive information about users' executed operations and access to protected health information. Access to audit logs must be controlled and restricted to a specific group of users with privileges consistent with the organization and other applicable policies. However, accessing or checking a user's activity data for various purposes is necessary. Therefore, a process must exist to verify the integrity of a user's audit logs from the audit blockchain without disclosing other users' activity data.

We propose a zero-knowledge-based audit-log integrity verification process, as shown in Figure 5.11. A user provides audit logs or activity data and receives either modified or unmodified responses from the system. Upon receiving a user request, the *Integrity Verifier (IV)* retrieves the block ID and its integrity from the audit blockchain. Then, IV queries the public blockchain to retrieve the block hash for the audit blockchain block ID. If the audit block's integrity and the stored hash on the public blockchain match, then *IV* is reported to the user as unmodified or tampered with. Otherwise, the activity data is tampered with in the audit blockchain.

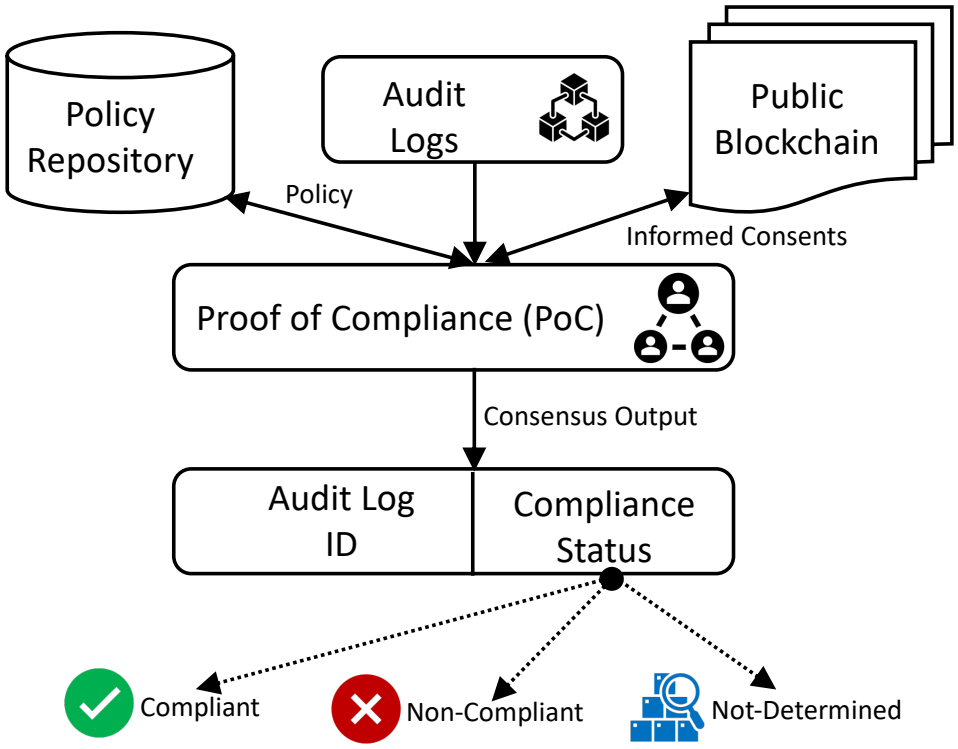

Figure 5.9: Compliance Block Transaction Structure [4]

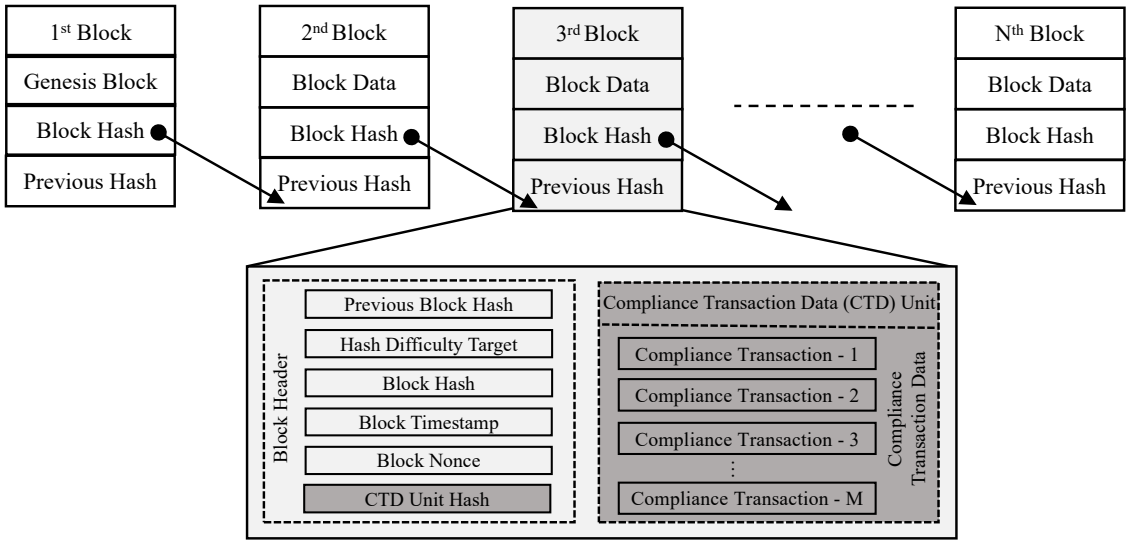

Figure 5.10: Compliance Blockchain Block Structure [4]

Moreover, all blocks in the audit blockchain are added via consensus. A single-bit modification invalidates all the blocks from the tampered block, indicating any modification. Here, the Integrity Verifier acts as a blind, trusted entity, analogous to an *API or Oracle*. It does not disclose data

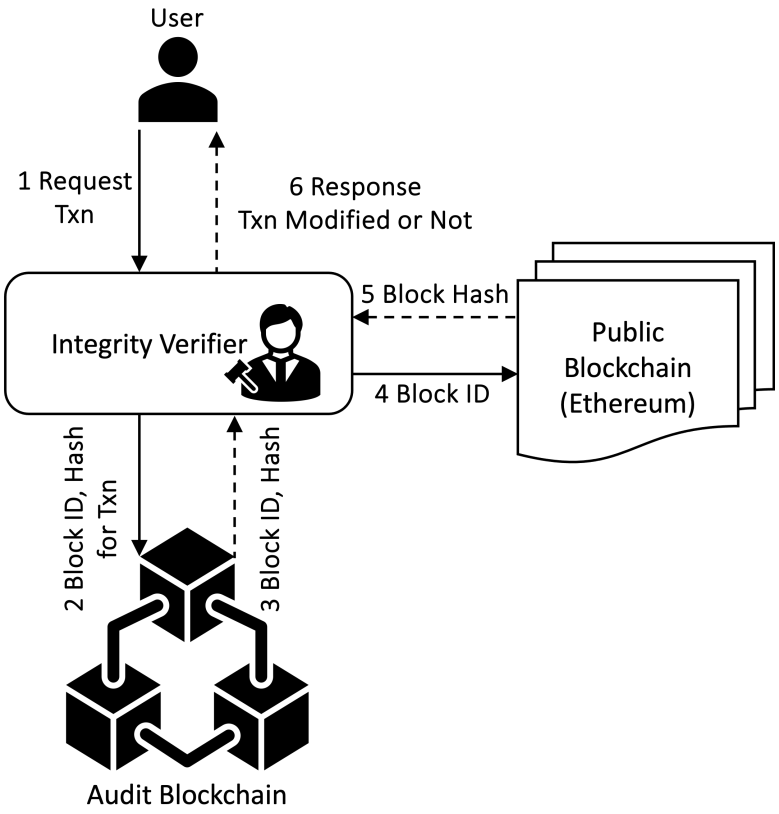

Figure 5.11: Audit Logs Integrity Verification [2]

to other entities and analyzes data for its own interest. It also does not modify any data while processing users' requests.

5.9 Private Audit Blockchain Setup

We have chosen private blockchain infrastructure to manage the provenance of audit logs, specifically an Ethereum private network deployed with the Go Ethereum (geth) client. This approach enhances data security and provides centralized control over policy-provenance activities. The private network employs the *Proof of Authority (PoA)* consensus algorithm, specifically the *Clique* protocol, to mine and validate the audit trails. Additionally, as the *Proof of Compliance* algorithm evolves, modifications to the *Clique* algorithm can be implemented to adapt the block structure to meet specific requirements.

```
INFO [11-30|00:04:45.006] Successfully sealed new block           number=2003 sealhash=ab913e..f93552 hash=0def93..687eda elapsed=5.001s
INFO [11-30|00:04:45.009] Commit new sealing work                 number=2004 sealhash=b28694..b15e07 txs=0 gas=0 fees=0 elapsed=2.191ms
INFO [11-30|00:04:49.408] Looking for peers                       peercount=1 tried=0 static=0
INFO [11-30|00:04:50.005] Successfully sealed new block           number=2004 sealhash=b28694..b15e07 hash=446467..f174f0 elapsed=4.996s
INFO [11-30|00:04:50.008] Commit new sealing work                 number=2005 sealhash=d5135d..dbb4bf txs=0 gas=0 fees=0 elapsed=1.807ms
WARN [11-30|00:04:53.154] Caller gas above allowance, capping     requested=112,915,065 cap=50,000,000
INFO [11-30|00:04:53.188] Setting new local account               address=0x9Ee65e7127516E08AdE560eb63fb8c26016348d7
INFO [11-30|00:04:53.188] Submitted contract creation             hash=0x4117d77443e685b70d9eef36d54cfe495e53641eb124a51c283101199f9f5b57 from=0x9Ee65e7127516E
8AdE560eb63fb8c26016348d7 nonce=67 contract=0x58707DABaC5Cd7AA5d12AB721a1bf003099436c6 value=0
INFO [11-30|00:04:54.014] Commit new sealing work                 number=2005 sealhash=3d1180..e80f66 txs=1 gas=105,687 fees=0.000105687 elapsed=2.270ms
INFO [11-30|00:04:55.003] Successfully sealed new block           number=2005 sealhash=3d1180..e80f66 hash=e9b97c..598d65 elapsed=989.129ms
INFO [11-30|00:04:55.005] Commit new sealing work                 number=2006 sealhash=6f9ba7..42cb72 txs=0 gas=0       fees=0           elapsed=1.643ms
```

Figure 5.12: Private Audit Blockchain Miner Node Operations [2]


```
{
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "muirGlacierBlock": 0,
    "berlinBlock": 0,
    "londonBlock": 0,
    "arrowGlacierBlock": 0,
    "grayGlacierBlock": 0,
    "clique": {
      "period": 5,
      "epoch": 30000
    }
  },
  "difficulty": "1",
  "gasLimit": "800000000",
  "extradata": "0x0000000000000000000000000000000000000000000000000000000000000009Ee65e7
  "alloc": {
    "9Ee65e7127516E08AdE560eb63fb8c26016348d7": { "balance": "1000000000000000000" },
    "8902491ed7Bef40Ed7C447dBbb084A83C04d37f1": { "balance": "1000000000000000000" }
  }
}
```


Figure 5.13: Private Audit Blockchain Genesis Block [2]

Figure 5.12 shows the miner node responsible for the end-to-end transaction handling process. Beginning with the submission of transactions. Once a transaction is submitted, the miner node includes it in a block and then proceeds to mine the block to validate it. Furthermore, the miner node actively broadcasts the mined data to all other nodes in the network, ensuring a synchronized, up-to-date ledger across the entire system.

Figure 5.13 shows the configuration file for the genesis block of our private network. In blockchain technology, a genesis block is the initial block in the chain, and the *JSON* file encapsulates the network's foundational parameters. In this file, the "*chainId*" attribute is set to "*12345*," which denotes the unique identifier assigned to the private network. The "*period*" attribute defines the block time, which determines the interval between consecutive blocks in the blockchain.

Chapter 6

Policy Compliance

Policy enforcement, provenance, and compliance checking are essential and interconnected, providing an overall compliance assurance framework. Required and applicable policies must be enforced through timely and proper enforcement mechanisms. Proper policy enforcement protects healthcare data from unauthorized access and misuse. While enforcing system security policies is essential, maintaining provenance to demonstrate policy compliance is equally important.

All enforcement activities or audit logs must be captured and stored in the system as they occur. Maintaining provenance integrity is essential for measuring policy compliance. Audit logs must also be protected from modification by any means or anyone. The applied or applicable policy lineage must also be maintained to verify compliance. Enforcement activities, audit logs as they are executed, and policy lineage together provide provenance.

Compliance is achieved when the policy is adequately enforced, since all actions must be taken in accordance with the applicable policy. However, this compliance is not measurable or verifiable by itself. An independent entity separate from the enforcement and provenance entity performs a policy audit to certify the policy's compliance status.

The healthcare industry is subject to varying degrees of regulatory oversight, and compliance with these regulations is essential for its operation and growth. The specific regulatory landscape varies across countries and regions, adding complexity to the industry. Non-compliance or compliance failure can lead to various business and legal issues, including medical and business license suspension, employee termination, monetary fines, business restrictions within a particular jurisdiction, loss of business reputation, and patient or client dissatisfaction. These non-compliance cases are most often identified during internal, external, or third-party audits and reviews.

To help healthcare organizations detect early non-compliance issues, this chapter presents a consensus mechanism, *Proof of Compliance* (PoC) [4]. Where a set of distributed, decentralized, and independent auditor nodes verifies the compliance status of any logical operations or accesses

that have already been approved, granted, or executed in the system to access sensitive health records. The PoC auditors work on audit logs and consent-provenance data that are securely stored and maintained on the audit and public blockchains. Chapter 5 presents the detailed proposed mechanism for maintaining a private blockchain-based audit-log provenance system. The *PoC* consensus mechanism helps organizations minimize compliance challenges. Organizations can consider PoC outputs and take further actions to reduce non-compliance cases, thereby avoiding compliance issues and business losses.

6.1 Introduction

Electronic health records (EHRs) have emerged as a cornerstone in modernizing healthcare, offering numerous benefits that enhance efficiency and quality of care [18]. These systems provide immediate and remote access to patient data, a critical feature that streamlines medical decision-making. By transitioning from paper-based systems, EHRs significantly reduce errors and costs commonly associated with manual record-keeping, thereby enhancing patient safety, affordability, and quality of care [17, 19]. One advantage of EHRs is their ability to promote interoperability across healthcare platforms. This interconnectedness enables seamless sharing of patient data among healthcare providers, improving continuity of care and the overall healthcare experience. They enhance clinical cooperation and improve diagnostic accuracy [15, 22].

However, this digital transformation also introduces complex information security and privacy challenges that are critical for maintaining patient trust. To address these challenges, the healthcare industry not only adopts robust security technologies but is also highly regulated, subject to specific laws, privacy standards, policies, and best practices that govern healthcare operations and services [161]. Examples of these are the General Data Protection Regulation (*GDPR*) in Europe [162], 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 [33]. 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, including access to and amendment of 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 it challenging to attract and retain customers and employees.

Unfortunately, even then, unauthorized access to and disclosure of health data remain prevalent in healthcare settings, heightening security and privacy concerns. For example, Table 1.1 shows the number of compliance complaints received by the U.S. Department of Health and Human Services (HHS) Office for Civil Rights (OCR) [1]. 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.

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 promptly 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 verifying compliance with audit logs. Audit logs are stored in a private blockchain network called *Audit Blockchain*. Where a set of independent auditor nodes performs compliance verification using the *PoC* blockchain consensus mechanism in a decentralized, distributed manner to determine the compliance status as *compliant*, *non-compliant*, or *not-determined*. After determining the compliance status, the audit log ID and the compliance status are stored in another private blockchain network, the *Compliance Blockchain* (Figure 6.1). Private blockchain block ID and hash for integrity are stored on the public blockchain. Involved entities can verify the integrity of private blockchain data against the public network, since any modification to the private blockchain alters its integrity.

The assumptions and scope of this approach 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 accurately and on time from the healthcare system 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 constitutes a policy violation. (e) Lastly, only logical activities are considered for compliance verification, such as patient electronic health records, physical location access, and other relevant operational processes. Based on these assumptions, this paper focuses on maintaining provenance and on compliance checking 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 specifically designed to ensure that all transactions and their corresponding blocks 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 are constantly evolving, so providers must stay current with the latest requirements.

In the evolving landscape of blockchain technology, where the integrity and security of distributed systems are paramount, consensus mechanisms are essential for maintaining network consensus 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 that bridges the gap between blockchains' autonomous, trustless nature and the stringent compliance demands of modern-day applications.

6.2 Consensus-Based Policy Compliance Review

The primary function of the blockchain consensus mechanism is to reach consensus on a set of transactions or data within a decentralized, distributed ledger. Each node must agree on and maintain the same set of transactions for a given block. To do this, a set of tasks must be done. Primary tasks include (i) collecting client transactions and storing them in the transaction memory pool for selection in 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 originate from the claimed users and have sufficient account balances to process transactions and transfer balances. (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 compete to be selected as block proposers, miners, or validators, or as forgers, to perform the aforementioned tasks. However, there is only one vacancy per 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 employ various mechanisms to select the block proposer, miner, validator, or forger to perform the seven (7) tasks mentioned above. For instance, the PoS uses validators' stakes and ages, whereas the PoW uses computational capacity to select the block proposer.

Compliance checking ensures that transactions or operations are executed in accordance with 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 responsible for compliance checking must be separate from those that perform operations or track 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, vulnerable to attacks, lacking transparency, dependent on external entities, and resulting in increased costs, etc. [163, 164]. To address these issues, a decentralized, distributed process is necessary to conduct compliance reviews against relevant 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. Compliance checking encompasses functionalities beyond those of the current consensus

mechanism. Compliance-checking unique processes (using provenance to verify compliance status) necessitates a new consensus mechanism that complements the functionalities of existing consensus mechanisms, as outlined in the seven points above. To address this, this paper proposes a compliance mechanism, *Proof of Compliance*, to validate compliance status in a decentralized and distributed manner. In a scenario where a set of independent auditor nodes performs compliance checking of audit logs using policy lineages, without a central or single entity.

6.3 Proposed Approach Overview

The proposed *Proof of Compliance* consensus mechanism for the healthcare industry would provide a means to ensure the compliance status of all activities and transactions granted and executed within the healthcare system. The compliance status can be *(i) compliant*, *(ii) non-compliant*, 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 6.1 depicts the proposed approach, whereas Figure 5.7 shows the Txn structure. The private blockchain block ID and hash, which ensure integrity, are stored on the public blockchain. This allows involved entities to verify the integrity of private blockchain data against the public network, since any modification to the private block alters its hash. Figure 6.1 depicts the proposed approach.

The following sections discuss the detailed *Proof of Compliance* consensus mechanism, the functionality of participant nodes, the audit log and compliance transaction and block structures, the private block ID and integrity storage process on the public blockchain network, policy compliance services, and initial experimental outcomes.

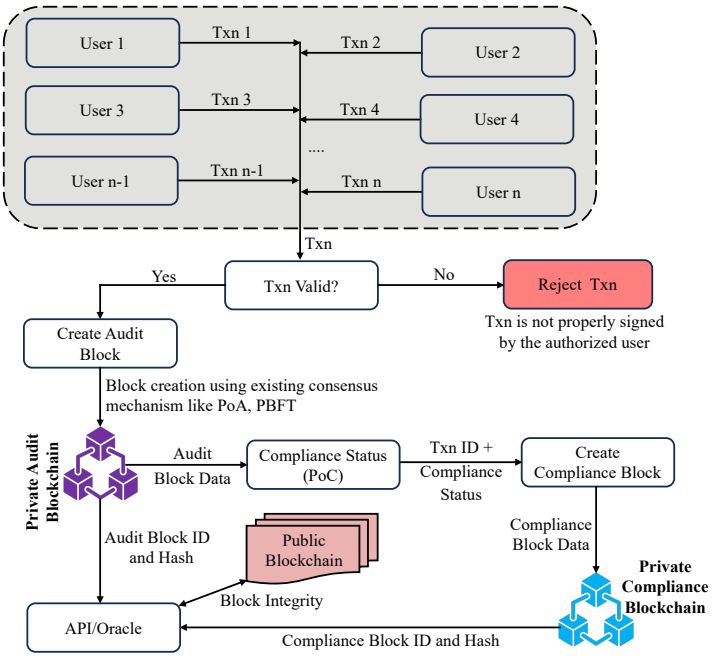

Figure 6.1: Proof of Compliance Process Overview [4]

6.4 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 [165]. 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 the 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.

In healthcare, major data protection laws and regulatory agencies, such as the *GDPR* and HIPAA, mandate patient consent for the collection, storage, processing, sharing, and other operations. The authors of [3, 13] proposed patient-consent-based *PHI* access for treatment team members and

sharing beyond the treatment team for marketing, research, advanced diagnosis, and consultation with another provider. This work focuses on consent-based policy compliance criteria and validation approaches. However, the proposed *PoC* mechanism applies to any set of policy compliance criteria defined by the organizations.

6.4.1 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.

This work assumes that a single consent governs each access request to simplify compliance checking. In practice, multiple consents influence the decision-making process for a data access

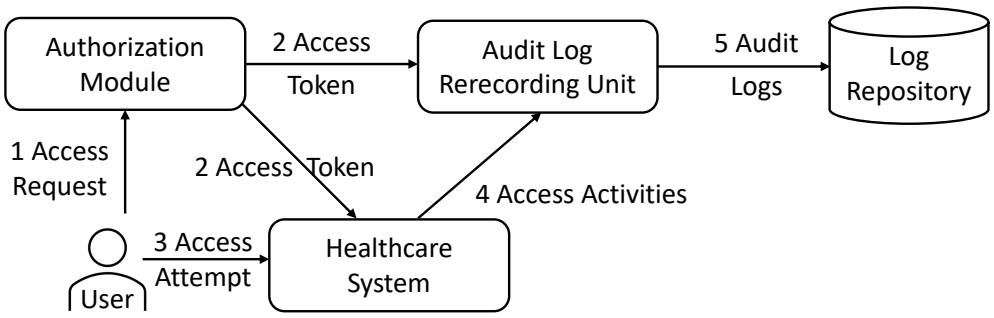

Figure 6.2: Audit Log Capture [4]

request, where a single log entry could correspond to several consents. However, in this study, each audit log entry is associated with a single consent, resulting in a one-to-one mapping between access requests and consents.

6.4.2 Access Token and Audit Log Capture

Only approved access requests 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 can access health records, even with approved requests. Figure 6.2 illustrates the audit log capture process using the *Access Token* defined below. Upon receiving a user request, the authorization module evaluates it 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. Both use time information to grant PHI access and to record audit logs. An audit log is also created and stored in the log repository if no access is made.

Definition 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:

- $\mathbb{R}_i$ *denotes the unique Request ID associated with the user's access request,*
- t_{start} *represents the Access Start Time when the access is first permitted; users cannot access health records before this time,*

- t_{end} *represents the Access End Time when the access expires. After this time, users will no longer be able to 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.

6.4.3 Compliance Status Verification

Definition 2. [Compliance Criteria ζ] *The compliance function ζ maps each audit log entry to its corresponding executed consent, verifying that an authorized consent underpins 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.*

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

6.5 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 in the proposed approach. In addition to these nodes, the *patient* gives consent, and those are deployed to the public blockchain as detailed in [3, 13]. The client nodes perform system activities, which are captured in audit logs and stored on the private audit blockchain. The following discusses the tasks of the participant nodes, message communication, and the transaction flows between the nodes. Algorithm 8 shows the steps of the PoC process.

Algorithm 8: Proof of Compliance (PoC) Consensus Mechanism [4]

Input : (i) list of transactions ($Txns$) and (ii) set of policy $Plcy$
Output :
(i) list of accepted/rejected transactions ($Txns$)
(ii) list of transactions that are policy compliance

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

(i) Orderer Node: It performs all transactions and consents to 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 the block ID and hash are stored previously for integrity. If no modifications are made, the audit logs and required consents are transmitted to the auditor nodes to conduct the compliance review.

(iii) Auditor Nodes: These nodes are responsible for checking the compliance requirements for regulations and other applicable bodies. Auditor nodes can include hospitals, local and state governments, the federal government, regulatory agencies, insurance companies, business associates, accreditation bodies, independent auditors, and others arising from contractual obligations. Each node functions as an honest entity, analyzing provenance data, audit trails, and applicable policies to determine the compliance status of activities. 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 [166]. 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 [167]. 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 either messages are eventually delivered correctly to all participants, or all participants abort the messages without side effects. However, this paper does not provide a detailed functional mechanism for message communication.

Figure 6.3 shows the sequence diagram of the transaction flow. 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. Consent is also required for the compli-

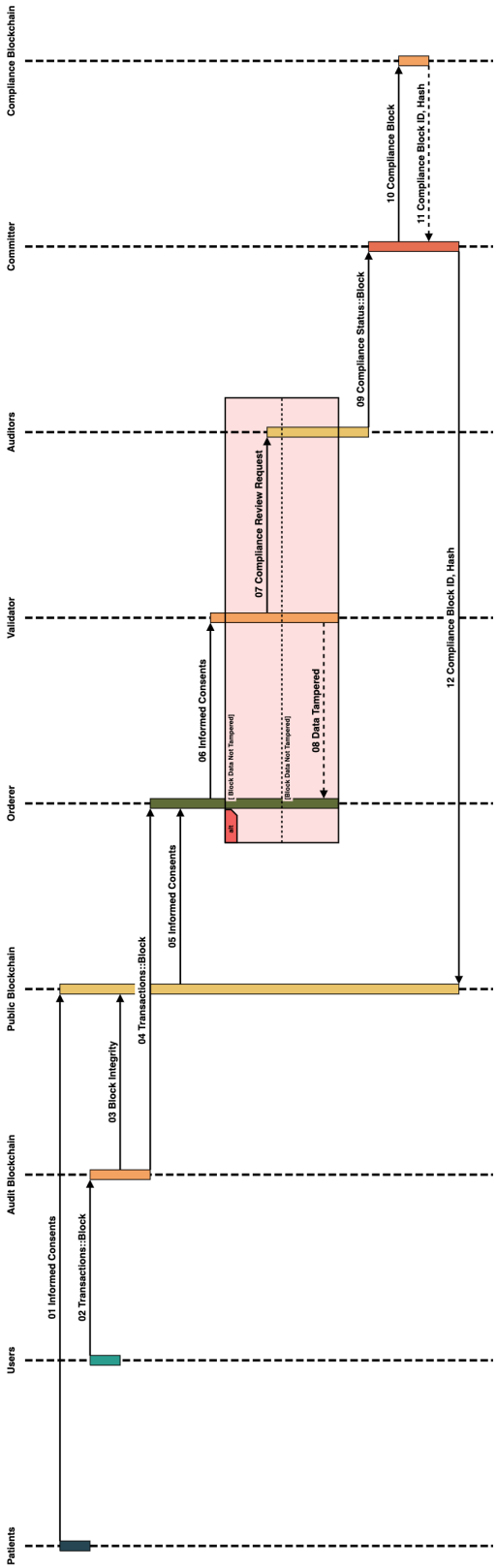

Figure 6.3: Proof of Compliance (PoC) Transaction Flow [4]

ance review. Client nodes submit transactions for validation and compliance checks before adding them to the ledger. The client nodes use their private keys to digitally sign all transactions.

(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. All public keys are published publicly to every node of the network. 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 compliant. Otherwise, the transaction is marked as non-compliant. 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 blockchain. After writing to the ledger, the compliance block ID and hash are recorded on the public blockchain for later verification.

6.6 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 6.4 shows the decision-making process.

6.6.1 Decision Counting Threshold

Suppose a total of 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 s responses. There may be cases in which auditors' responses are lost due to connectivity issues, power failures, intentional non-submission of results, the auditor node being

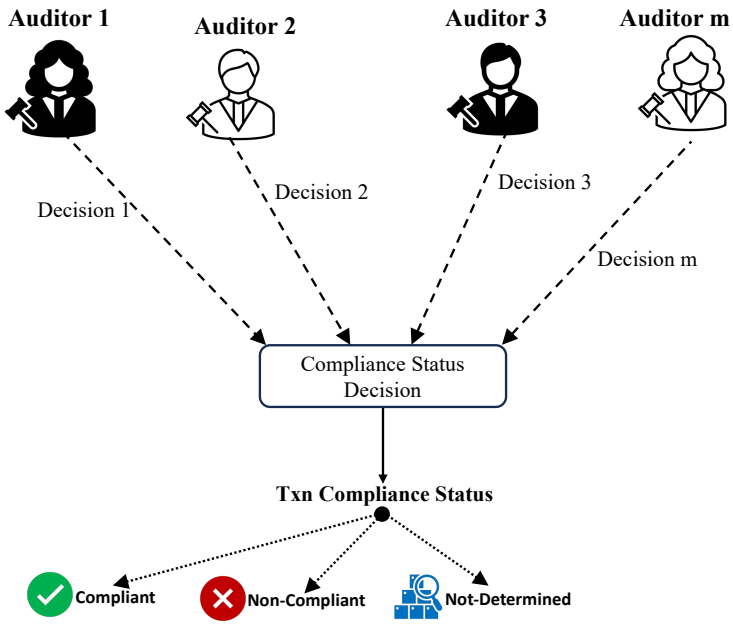

Figure 6.4: Proof of Compliance (PoC) Decision Mechanism [4]

offline, or, after the process starts, the auditor node going offline due to a system error, among others [168]. 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. As *"Not-Determined"* cases must be further investigated to check the reasons or any violations.

6.6.2 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} 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 on the mandatory participation of the auditors, A_R, the above requirements are waived.

6.6.3 Auditors and Decisions

Let m denote the total number of auditor nodes. The following information is given.

- 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: $(a_1, d_1), (a_2, d_2), \ldots, (a_m, d_m)$. This mapping enables us to jointly analyze the decisions and apply the *PoC decision-combining algorithm*.
- Let W be a set of weights defined as $W = \{w_1, w_2, w_3, \ldots, w_m\}$, where w_i represents the weight of an individual auditor node a_i. The final compliance decision is derived from the majority rule over the decisions.

Purpose of Weighting: Weighting allows for differentiated influence among auditors. For instance, an auditor from a regulatory agency may carry greater weight due to their authority, whereas an internal auditor may have a standard weight. This flexibility is beneficial in settings in which some nodes bear greater responsibility for compliance oversight. 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 business requirements, legal jurisdictions, regulatory mandates, industry-standard best practices, contractual obligations, and other factors.

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 computes the weighted counts for each decision type (*Compliant, Non-Compliant, Not-Determined*) and determines the final compliance status according to the 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.

6.6.4 Decision Counting and Combining Process with Weight

In this scope, all the auditor nodes don't bear the same weight values, where $w_1 \neq w_2 \neq w_3 \neq \ldots \neq w_m$, indicating that they don't have an equal impact on the decision. The weight value depends on the auditor's nature and business requirements.

Decision Counts with Weight: The total counts for each type of decision with weight are calculated as follows:

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

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

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

Where $\delta(.)$ is an indicator function that equals *1* if the inside condition is satisfied and *0* otherwise.

Decision-Combining Process with Weight: After counting each decision type, the final decision is made by majority vote. The *Weighted Not-Determined* dictates to others if they are equal to it. The distinct combinations are given in Table 6.1. The final decision $\mathbb{D}_{\mathbb{W}f_{inal}}$ can then be 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 $\mathbb{C}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}}$.

6.6.5 Decision Counting and Combining Process without Weight

In this setting, all auditor nodes have the same weight, where $w_1 = w_2 = w_3 = \cdots = w_m$, indicating that they have equal impact on the decision.

Decision Counts without Weight: The total counts for each type of decision are calculated as follows:

Table 6.1: Proof of Compliance (PoC) Decision Combining Scope with Weight [4]

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	$\mathbb{U}_{\mathbb{W}} > \mathbb{N}_{\mathbb{W}} > \mathbb{C}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$
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	$\mathbb{C}_{\mathbb{W}} = \mathbb{N}_{\mathbb{W}} = \mathbb{U}_{\mathbb{W}}$	$\mathbb{U}_{\mathbb{W}}$

Table 6.2: Proof of Compliance (PoC) Decision Combining Scope without Weight [4]

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	$\mathbb{C} = \mathbb{N} = \mathbb{U}$	$\mathbb{U}$

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

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

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

Where $\delta(.)$ is an indicator function that equals *1* if the inside condition is true and *0* otherwise.

Decision-Combining Process without Weight: After counting, the final decision is made by majority vote. The *Not-Determined* dictates to others if they are equal to it. The distinct combinations are given in Table 6.2. 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 $\mathbb{C} = \mathbb{N}$.

6.7 PoC Participant Incentive

The proposed approach does not provide direct incentives for participating auditor nodes. Auditor nodes engage in the PoC consensus mechanism as part of their core responsibilities, fulfilling compliance requirements mandated by their respective organizations or regulatory agencies. Participation in the compliance evaluation process is restricted to auditor nodes that meet specific, pre-established criteria, ensuring that only authorized and qualified entities are involved. This

restriction maintains the integrity of the compliance process and prevents unauthorized access. Furthermore, the information used to determine compliance status is carefully managed to avoid compromising security or privacy.

Xiao et al. [169] identified five critical components of a blockchain consensus protocol: *(i) block proposal, (ii) information propagation, (iii) block validation, (iv) block finalization,* and *(v) incentive mechanism*. Though the five components provide essential functionalities for blockchain consensus, a new blockchain consensus protocol proposal may not cover all of them. For instance, we consider the *PoC* for a permission network in which several organizations must approve the auditor nodes. The incentive mechanism is indispensable for participant nodes in permissionless blockchain networks. This work aims to extend PoC to public networks and to incorporate incentive mechanisms in future work.

6.7.1 Healthcare Data Security and Patient Privacy

The auditor nodes verify the compliance status of healthcare workers' access activities. They must access the deployed and applicable policies, along with their corresponding audit trails. In this situation, the audit nodes do not access any protected health information directly or indirectly. However, activity logs can contain sensitive information about health professionals and patients. For example, learning about a patient's ongoing treatment from the audit trails is possible. If a specialist doctor accesses a patient's health data, it can be concluded that the doctor is treating the patient. In the proposed *Proof of Concept (PoC)* consensus mechanism, the auditor nodes serve as secure, trusted, and blind entities. They don't reveal data to other unauthorized users. They don't analyze data to learn about providers and patients. They do not modify or tamper with data or PoC decisions. They do not store data intentionally. They analyze data solely for compliance status decisions. The complete protocols for them to act as secure, trusted, and blind entities are currently not addressed in this paper. They are our future research directions.

6.7.2 Blockchain Data as Court Evidence

In June 2018, a landmark court judgment affirmed for the first time that electronic data stored on a blockchain could be considered valid electronic evidence. Subsequently, in a separate case in 2019, the court extended this recognition to the authenticity and integrity of electronic evidence stored on a blockchain and evidence generated by the blockchain itself [170]. The Blockchain Technology Act, House Bill 3575, passed by the House of Representatives on May 29, 2019, and enacted in January 2020, governs the utilization of blockchain technology in transactions and legal proceedings. The Act sets forth regulations, imposes limitations, and defines terms, including blockchain and electronic records. According to the Act, using a blockchain to create, store, or verify a smart contract, record, or signature does not undermine its legal effect or enforceability.

Moreover, electronic evidence on a blockchain is deemed sufficient when the law mandates written records. Similarly, for situations requiring a signature, an electronically recorded signature on a blockchain or blockchain evidence confirming a person's intent to provide a signature is considered satisfactory [171]. Hence, implementing our prototype using blockchain for smart contract storage complies with US law. The hospital can submit complaints about policy violations against offenders, supported by evidence from our blockchain framework.

6.8 Experimental Evaluation

This section provides experimental evaluation 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, accompanied by the necessary data, figures, and tables.

Environment Setup

To implement the blockchain model, we use *Node.js* to develop server-side programs for the roles of *Client*, *Orderer*, *Auditor*, and *Committer* in our network. 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 that mimics healthcare transaction activities and audit logs. This data tests the blockchain system's performance and reliability. Nodes were encapsulated in *Docker* containers to ensure isolated, consistent environments, thereby simplifying dependency management and network communications [172]. We used *Go Ethereum Docker* image *version 1.13.15* for our private network setup. This setup effectively imitates a distributed ledger. 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*.

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 in which participants are limited to organizations. This doesn't instill public trust. To avoid this, block IDs and hashes are stored on a public network, such as *Ethereum*, to ensure integrity. Figure 6.5 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 6.6. The first two networks are *Layer 1*, and the third network is *Layer 2* [63, 154]. *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, while storing the proof on it. The *Layer 2* solution processes more transactions per second, reducing transaction costs and shortening confirmation times.

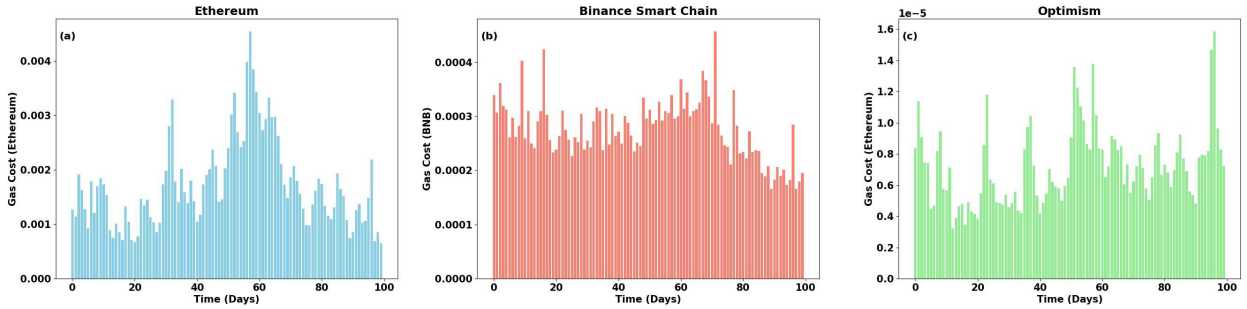

Figure 6.5: Private Block ID and Integrity Storage Token Cost—Public Blockchain Networks [4]

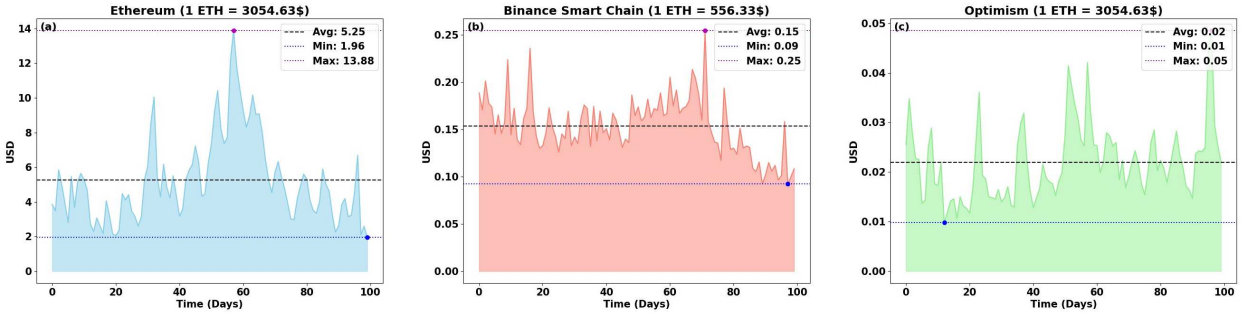

Figure 6.6: Private Block ID and Integrity Storage USD Cost—Public Blockchain Networks [4]

Time Requirements

Before processing any data from the private network, the block integrity is verified from the public network. This requires access to a public blockchain network to read the stored private blockchain block IDs and hashes. This study leverages *Ethereum's Remote Procedure Call (RPC) API* to deploy smart contracts and execute transactions on these networks [129]. Writing and reading time requirements are measured for three networks: *Ethereum, Binance Smart Chain, and Optimism*. Table 6.3 shows ten (10) writing time requirements. Table 6.4 shows ten (10) reading time requirements. Additional time is required because transactions traverse the API servers. Maintaining a local public blockchain node provides real-time access to block data, reducing the time spent reading it. The system continuously synchronizes with the blockchain network to update the ledger data. Providers can maintain local nodes to enable faster integrity verification. However, time differences are not accounted for in this study.

Table 6.3: Writing Time to Public Blockchain Networks [4]

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 6.4: Reading Time from Public Blockchain Networks [4]

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*

Compliance Block Construction Time

After performing compliance checking and block finalization, it is time to confirm the compliance block. The *Auditor* nodes are responsible for compliance checking and for making final decisions on the compliance status. The *Committer* nodes finalize blocks by writing the compliance block to the compliance blockchain ledger. It does not include the time required by the *Orderer* nodes to fetch audit logs from the private audit blockchain, obtain informed consent from the public blockchain network, and retrieve applicable policies from the policy repository. Figure 6.7(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.

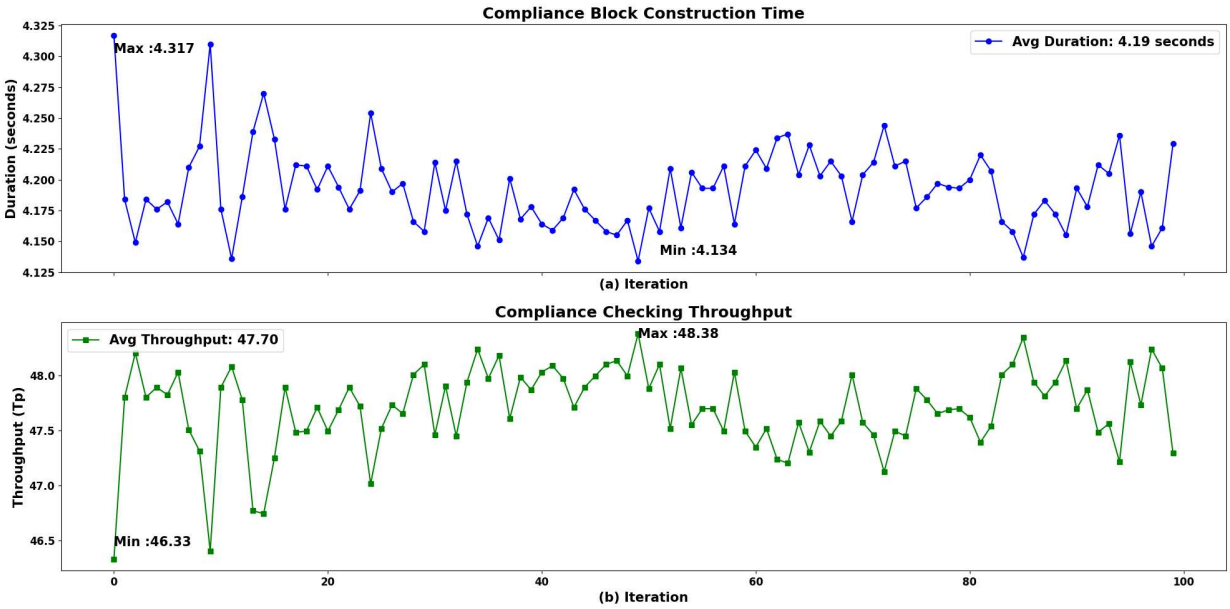

Figure 6.7: (a) Compliance Block Construction Time (b) Compliance Checking Throughput [4]

Compliance Checking Throughput

It is the number of transactions per second that can be processed after all required operations are performed. For the *Proof of Compliance* consensus mechanism, the operations performed are compliance checking and finalizing the compliance block by the *Auditor* and *Committer* nodes. Figure 6.7(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.

6.9 Policy Compliance Services

Policy compliance services provide the status of healthcare providers' adherence to the required security and privacy policies to protect sensitive healthcare data. Services also indicate how healthcare providers maintain the integrity of healthcare operations to provide patient treatments and essential healthcare services. Figure 6.8 illustrates the proposed mechanism for providing policy compliance services. Users can access default service reports containing all audit logs and their corresponding compliance status. Additionally, they can submit customized queries to receive reports. User- or request-related data is fetched from the *audit blockchain* and the *compliance*

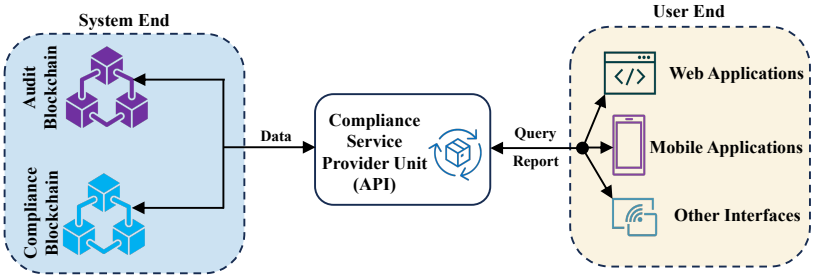

Figure 6.8: Proposed Policy Compliance Services Mechanism

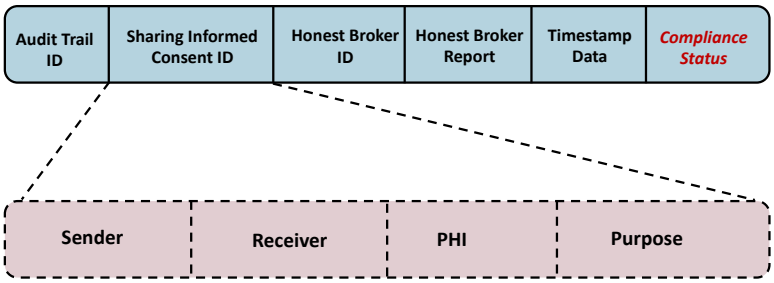

Figure 6.9: Policy Compliance Service Report Structure

blockchain. A trusted entity or an API-based *Compliance Service Provider Unit* intercepts, processes service-related requests, and produces and delivers reports to the requested users. Figure 6.9 depicts the policy compliance service report structure. At the end of each audit log type, a compliance status is added and reported to the corresponding users. The details about audit logs are discussed in Figure 5.5. Policy compliance services for patients and users are discussed below.

Audit Log ID	Treatment Informed Consent ID	User	PHI	Operation	Conditions	Timestamp	Status
T5351	TIC52752	PR8573	PHI5900	Read	C1 C2 C3 C4 C5 C6	4/19/2024, 4:06:12 PM	Non-determined
T5352	TIC36507	PHR6187	PHI1879	Write	C1 C2 C3 C4	4/19/2024, 4:06:12 PM	Compliant
T5353	TIC40636	ICA3540	PHI4049	Update	C1 C2 C3 C4 C5	4/19/2024, 4:06:12 PM	Compliant
T5354	TIC73352	PR1977	PHI3805	Read	C1 C2 C3 C4 C5 C6	4/19/2024, 4:06:12 PM	Compliant
T5355	TIC64084	PHR8069	PHI9749	Write	C1 C2 C3 C4 C5 C6 C7	4/19/2024, 4:06:12 PM	Compliant
T5356	TIC46434	ICA1289	PHI3769	Update	C1 C2 C3 C4 C5 C6	4/19/2024, 4:06:12 PM	Non-compliant
T5357	TIC95689	PR5882	PHI2183	Read	C1 C2 C3 C4	4/19/2024, 4:06:12 PM	Compliant
T5358	TIC31098	PHR5533	PHI8110	Write	C1 C2 C3 C4 C5 C6	4/19/2024, 4:06:12 PM	Compliant
T5359	TIC84090	ICA3385	PHI7849	Update	C1 C2 C3 C4 C5 C6	4/19/2024, 4:06:12 PM	Compliant
T5360	TIC6451	PR1919	PHI1649	Read	C1 C2 C3 C4	4/19/2024, 4:06:12 PM	Compliant
T5361	TIC536	PHR5767	PHI5016	Write	C1 C2 C3 C4 C5	4/19/2024, 4:06:12 PM	Compliant
T5362	TIC90668	ICA4928	PHI5950	Update	C1 C2 C3 C4 C5	4/19/2024, 4:06:12 PM	Compliant

Figure 6.10: Policy Compliance Service Sample Report

6.9.1 Services for Patients

Patients must know who accessed their healthcare data, when, and under what circumstances. They also need to know the compliance status of each access. As all audit logs and compliance status are provided in a timely and adequate manner, patients can be confident in the security and privacy of their healthcare data. If a patient discovers that their health records were accessed without their consent, they may file an official complaint.

6.9.2 Services for Users

Here, users access patients' health information, healthcare facilities, and other objects protected by the security and privacy policy. These users may be members of the treatment team, such as doctors, nurses, support staff, radiology and pathology lab technicians, billing officers, pharmacists, insurance company agents, and patients' emergency contact personnel. Others may include providers and lab technicians from external hospitals, where patients seek advanced care and sophisticated diagnostics that primary care providers cannot provide. For example, a doctor, *David*, can check the number of patients and the health records he accesses over a specific time period, as well as their compliance status. David can learn about non-compliant cases in which he has violated security and privacy policies from compliance reports. Then, he can collaborate with the relevant authority to enhance his knowledge, skills, and performance, thereby preventing further policy violations. This retroactive process enables health workers such as David to enhance their awareness and skills in protecting sensitive patient data by enforcing security and privacy policies.

Policy compliance services increase trust among patients, providers, and other involved entities by ensuring transparency through blockchain-based audit logs and PoC-based compliance status. These services can be provided to various entities, including insurance companies, local, state, and federal governments, regulatory agencies, and international data protection agencies. These entities can access services if policy and regulatory requirements mandate their involvement. However, at this point, this paper considers only the patients and users who access health records and healthcare facilities. In future communications, other entities will be added for the services.

6.10 Chapter Conclusion

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 to address compliance requirements heavily influenced by regulations. Compliance with security and privacy policies and regulations reflects the extent to which healthcare organizations adhere to these policies and regulations. It also shows the integrity of the operations in handling sensitive patients' health records. This mechanism not only enhances 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 aim to formally verify the *Proof of Compliance* consensus mechanism. 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, dynamically aligns with evolving regulatory requirements, and fosters trust, thereby facilitating wider adoption in the healthcare industry. Such verification is essential to confirm the security and effectiveness of PoC systems before deployment in sensitive and critical healthcare application environments.

Chapter 7

Conclusion and Future Directions

This chapter summarizes the dissertation's findings, highlighting the key results, benefits, and challenges encountered throughout the study. It also discusses the study's limitations and presents concluding remarks along with future research directions that collectively encapsulate the core contributions and broader implications of this research.

7.1 Conclusion

Policy compliance and provenance are critical for ensuring healthcare organizations adhere to security and privacy policies and measures designed to protect sensitive healthcare data and patient privacy. Timely enforcement of these policies and conditions during authorization processes is equally essential. This proposed work emphasizes the significance of patient-provider agreements by integrating them with existing policies, enhancing their effectiveness. It extends the traditional attribute-based access control model with a blockchain-based smart contract framework to ensure robust policy enforcement, provenance, and compliance.

Introducing the Proof of Compliance (PoC) consensus mechanism facilitates compliance verification through independent, distributed, and decentralized auditor nodes. This approach enhances transparency, assurance, and accountability for all stakeholders in the healthcare ecosystem. Leveraging the inherent immutability and transparency of blockchain technology, smart contracts can automate and enforce policy rules, thereby strengthening trust, operational efficiency, and accountability across healthcare operations.

7.2 Future Research Directions

Many promising areas for research, development, and refinement remain to be explored. The key directions are briefly discussed below, each offering opportunities to enhance the proposed

framework's capabilities and maturity. Together, these efforts will pave the way toward a more intelligent, transparent, and resilient healthcare compliance ecosystem.

- *Blockchain-Based Healthcare Data Access:* Future work will focus on developing a private blockchain-based solution to enable patients' secure access to their healthcare data while maintaining rigorous compliance with privacy and regulatory requirements. In this model, each patient will have controlled access to their data as collected and managed by healthcare providers. The approach will also ensure that healthcare data is not collected, processed, or shared beyond its intended and authorized purposes, thereby strengthening trust, transparency, and accountability in data management.
- *Framework Integration Test:* As the proposed framework evolves, the enforcement, provenance, and compliance verification components have been designed and evaluated individually. In the next phase, all these components will be integrated into a unified system to validate their interactive functionalities and ensure that the overall policy compliance framework operates as intended. Scripted healthcare providers will simulate healthcare operations using synthetically generated patient data, during which provenance information will be automatically captured, and a real-time compliance review will be performed to assess the system's effectiveness and accuracy.
- *Compliance Framework Validation:* Future research will focus on validating the proposed policy compliance framework through both formal verification and empirical studies. Formal validation techniques, such as model checking and theorem proving, will be employed to ensure the logical correctness, consistency, and completeness of the framework's enforcement and compliance mechanisms. In parallel, empirical evaluations will be conducted using real or synthetically generated healthcare datasets to assess the framework's practical performance, scalability, and reliability in real-world conditions. The combined use of formal and empirical validation will strengthen confidence in the framework's robustness, ensuring it meets both theoretical soundness and practical applicability requirements.

- *Real-World Implementation and Validation:* The next phase involves collaborating with healthcare institutions to implement this blockchain-based architecture in real-world scenarios. This includes evaluating its efficacy, applicability, and regulatory acceptance among key stakeholders, including healthcare providers, patients, business associates, regulatory agencies, insurance companies, and others. By doing so, the proposed model can be fine-tuned to meet practical and regulatory requirements, ensuring its successful integration into the healthcare domain.
- *AI/ML-Based Compliance:* Future research can focus on integrating *Artificial Intelligence (AI)* and *Machine Learning (ML)* techniques into the proposed policy compliance framework to enable intelligent detection, prediction, and analysis of policy violation patterns. This integration would support proactive decision-making, adaptive policy enforcement, and continuous improvement of compliance effectiveness in dynamic healthcare environments.

Bibliography

[1] Office for Civil Rights (OCR). HIPAA Enforcement, May 2008. Last Modified: 2021-06-28T08:59:34-0400.

[2] Md Al Amin, Hemanth Tummala, Seshamalini Mohan, and Indrajit Ray. Healthcare policy compliance: A blockchain smart contract-based approach. *arXiv preprint arXiv:2312.10214*, 2023.

[3] Md Al Amin, Hemanth Tummala, Rushabh Shah, and Indrajit Ray. 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*, pages 211–223. INSTICC, SciTePress, 2024.

[4] Md Al Amin, Hemanth Tummala, Rushabh Shah, and Indrajit Ray. 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, page 119–130, New York, NY, USA, 2025. Association for Computing Machinery.

[5] Natalia Chaudhry and Muhammad Murtaza Yousaf. Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities. In *2018 12th international conference on open source systems and technologies (ICOSST)*, pages 54–63. IEEE, 2018.

[6] Mingyue Xie, Jun Liu, Shuyu Chen, and Mingwei Lin. A survey on blockchain consensus mechanism: research overview, current advances and future directions. *International Journal of Intelligent Computing and Cybernetics*, 16(2):314–340, 2023.

[7] Ashok Kumar Yadav, Karan Singh, Ali H Amin, Laila Almutairi, Theyab R Alsenani, and Ali Ahmadian. A comparative study on consensus mechanism with security threats and future scopes: Blockchain. *Computer Communications*, 201:102–115, 2023.

[8] Bishakh Chandra Ghosh, Tanay Bhartia, Sourav Kanti Addya, and Sandip Chakraborty. Leveraging public-private blockchain interoperability for closed consortium interfacing. In *IEEE INFOCOM 2021-IEEE conference on computer communications*, pages 1–10. IEEE, 2021.

[9] Sonia Singh, Arun Kumar, and Mamta Kathuria. Understanding the public, private and consortium consensus algorithms in blockchain technology. *International Journal of Blockchains and Cryptocurrencies*, 3(3):269–288, 2022.

[10] Md Al Amin, Rushabh Shah, Hemanth Tummala, and Indrajit Ray. Did you break the glass properly? a policy compliance framework for protected health information (phi) emergency access. In *Proceedings of the 22nd International Conference on Security and Cryptography - SECRYPT*, pages 195–208. INSTICC, SciTePress, 2025.

[11] Md Al Amin, Hemanth Tummala, Rushabh Shah, and Indrajit Ray. Empowering patients for disease diagnosis and clinical treatment: A smart contract-enabled informed consent strategy. In Pierangela Samarati and Sabrina De Capitani di Vimercati, editors, *Security and Cryptography*, pages 49–74, Cham, 2026. Springer Nature Switzerland.

[12] Md Al Amin, Amani Altarawneh, Sumantra Sarkar, and Indrajit Ray. Blockchain smart contracts for policy compliance: A healthcare perspective. In *2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pages 1–6. IEEE, 2023.

[13] Md Al Amin, Amani Altarawneh, and Indrajit Ray. 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*, pages 159–170. INSTICC, SciTePress, 2023.

[14] Nir Menachemi and Robert G Brooks. Reviewing the benefits and costs of electronic health records and associated patient safety technologies. *Journal of medical systems*, 30(3):159–168, 2006.

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

[16] Abdul Kader Saiod, Darelle van Greunen, and Alida Veldsman. Electronic health records: benefits and challenges for data quality. In *Handbook of Large-Scale Distributed Computing in Smart Healthcare*, pages 123–156. Springer, 2017.

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

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

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

[20] Shahid Munir Shah and Rizwan Ahmed Khan. Secondary use of electronic health record: Opportunities and challenges. *IEEE access*, 8:136947–136965, 2020.

[21] Subrata Acharya, Brian Coats, Arpit Saluja, and Dale Fuller. Secure electronic health record exchange: achieving the meaningful use objectives. In *2013 46th Hawaii International Conference on System Sciences*, pages 2555–2564. IEEE, 2013.

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

[23] E Cherif and M Mzoughi. Electronic health record adopters: a typology based on patients' privacy concerns and perceived benefits. *Public Health*, 207:46–53, 2022.

[24] Ahmad Al-Marsy, Pankaj Chaudhary, and James Allen Rodger. A model for examining challenges and opportunities in use of cloud computing for health information systems. *Applied System Innovation*, 4(1):15, 2021.

[25] Arshdeep Bahga and Vijay K Madisetti. A cloud-based approach for interoperable electronic health records (ehrs). *IEEE journal of biomedical and health informatics*, 17(5):894–906, 2013.

[26] Hari Krishna Karri, Ayesha Begum, and Lina George. Optimizing healthcare efficiency: The role of artificial intelligence in medical records management. *International Journal of Engineering and Management Research*, 14(6):55–67, 2024.

[27] Adekunle Oyeyemi Adeniyi, Jeremiah Olawumi Arowoogun, Rawlings Chidi, Chioma Anthonia Okolo, and Oloruntoba Babawarun. The impact of electronic health records on patient care and outcomes: A comprehensive review. *World Journal of Advanced Research and Reviews*, 21(2):1446–1455, 2024.

[28] Abhishek Vyas, Satheesh Abimannan, and Ren-Hung Hwang. Sensitive healthcare data: Privacy and security issues and proposed solutions. *Emerging Technologies for Healthcare: Internet of Things and Deep Learning Models*, pages 93–127, 2021.

[29] David Grande, Xochitl Luna Marti, Rachel Feuerstein-Simon, Raina M Merchant, David A Asch, Ashley Lewson, and Carolyn C Cannuscio. Health policy and privacy challenges associated with digital technology. *JAMA network open*, 3(7):e208285–e208285, 2020.

[30] Neethu Mathai, MF Shiratudin, and F Sohel. Electronic health record management: expectations, issues, and challenges. *Journal of Health & Medical Informatics*, 8(3):1–5, 2017.

[31] Ronald Bayer, John Santelli, and Robert Klitzman. New challenges for electronic health records: confidentiality and access to sensitive health information about parents and adolescents. *Jama*, 313(1):29–30, 2015.

[32] Farhad Fatehi, Farkhondeh Hassandoust, Ryan KL Ko, and Saeed Akhlaghpour. General data protection regulation (gdpr) in healthcare: Hot topics and research fronts. In *Medical Informatics in Europe Conference (MIE) 2020*, pages 1118–1122. IOS Press, 2020.

[33] Scholas Mbonihankuye, Athanase Nkunzimana, and Ange Ndagijimana. Healthcare data security technology: Hipaa compliance. *Wireless communications and mobile computing*, 2019(1):1927495, 2019.

[34] Patrick Cheong-Iao Pang, Dana McKay, Shanton Chang, Qingyu Chen, Xiuzhen Zhang, and Lishan Cui. Privacy concerns of the australian my health record: Implications for other large-scale opt-out personal health records. *Information Processing & Management*, 57(6):102364, 2020.

[35] Gina M Berg, Taylor Shupsky, and Kevin Morales. Resident indentified violations of usability heuristic principles in local electronic health records. *Kansas Journal of Medicine*, 13:84, 2020.

[36] Waldemar W Koczkodaj, Jolanta Masiak, Mirosław Mazurek, Dominik Strzałka, and Pavel F Zabrodskii. Massive health record breaches evidenced by the office for civil rights data. *Iranian Journal of Public Health*, 48(2):278, 2019.

[37] Wullianallur Raghupathi, Viju Raghupathi, and Aditya Saharia. Analyzing health data breaches: A visual analytics approach. *AppliedMath*, 3(1):175–199, 2023.

[38] Shruti Patne and Deepika Kanyal. The evaluating compliance and monitoring practices: A comprehensive review of auditing surgical safety checklists against regulatory standards and guidelines. *Multidisciplinary Reviews*, 8(3):2025068–2025068, 2025.

[39] Sumantra Sarkar, Anthony Vance, Balasubramaniam Ramesh, Menelaos Demestihas, and Daniel Thomas Wu. The influence of professional subculture on information security policy violations: A field study in a healthcare context. *Information Systems Research*, 31(4):1240–1259, 2020.

[40] Peter Garpenby and Ann-Charlotte Nedlund. The patient as a policy problem: Ambiguous perceptions of a critical interface in healthcare. *Health*, 26(6):681–701, 2022.

[41] Luke Fowler. How to implement policy: Coping with ambiguity and uncertainty. *Public Administration*, 99(3):581–597, 2021.

[42] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*, pages 557–564. Ieee, 2017.

[43] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. *Ieee Access*, 7:22328–22370, 2019.

[44] Jie Xu, Cong Wang, and Xiaohua Jia. A survey of blockchain consensus protocols. *ACM Computing Surveys*, 55(13s):1–35, 2023.

[45] Wubing Chen, Zhiying Xu, Shuyu Shi, Yang Zhao, and Jun Zhao. A survey of blockchain applications in different domains. In *Proceedings of the 2018 international conference on blockchain technology and application*, pages 17–21, 2018.

[46] Erikson Júlio De Aguiar, Bruno S Faiçal, Bhaskar Krishnamachari, and Jó Ueyama. A survey of blockchain-based strategies for healthcare. *ACm Computing surveys (CsUr)*, 53(2):1–27, 2020.

[47] Wattana Viriyasitavat and Danupol Hoonsopon. Blockchain characteristics and consensus in modern business processes. *Journal of Industrial Information Integration*, 13:32–39, 2019.

[48] Weiqin Zou, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Xia, Yang Feng, Zhenyu Chen, and Baowen Xu. Smart contract development: Challenges and opportunities. *IEEE transactions on software engineering*, 47(10):2084–2106, 2019.

[49] Bhabendu Kumar Mohanta, Soumyashree S Panda, and Debasish Jena. An overview of smart contract and use cases in blockchain technology. In *2018 9th international conference on computing, communication and networking technologies (ICCCNT)*, pages 1–4. IEEE, 2018.

[50] Shafaq Naheed Khan, Faiza Loukil, Chirine Ghedira-Guegan, Elhadj Benkhelifa, and Anoud Bani-Hani. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications*, 14(5):2901–2925, 2021.

[51] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, 2020.

[52] Bahareh Lashkari and Petr Musilek. A comprehensive review of blockchain consensus mechanisms. *IEEE access*, 9:43620–43652, 2021.

[53] Ammar Al-Ashmori, Shuib Bin Basri, PDD Dominic, Luiz Fernando Capretz, Amgad Muneer, Abdullateef Oluwagbemiga Balogun, Abdul Rehman Gilal, and Rao Faizan Ali. Classifications of sustainable factors in blockchain adoption: a literature review and bibliometric analysis. *Sustainability*, 14(9):5176, 2022.

[54] Aparna Singh, Jaya Sinha, Tanu Shree, and Surbhi Sharma. Exploring the spectrum of blockchain: Private, public, consortium, and hybrid and their applications. In *Navigating the Blockchain Revolution: Decentralization, Finance, and Beyond*, pages 217–242. Bentham Science Publishers, 2025.

[55] Rebecca Yang, Ron Wakefield, Sainan Lyu, Sajani Jayasuriya, Fengling Han, Xun Yi, Xuechao Yang, Gayashan Amarasinghe, and Shiping Chen. Public and private blockchain

in construction business process and information integration. *Automation in construction*, 118:103276, 2020.

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

[57] Ghassan Al-Sumaidaee, Rami Alkhudary, Zeljko Zilic, and Andraws Swidan. Performance analysis of a private blockchain network built on hyperledger fabric for healthcare. *Information Processing & Management*, 60(2):103160, 2023.

[58] Xiaotong Chen, Songlin He, Linfu Sun, Yangxin Zheng, and Chase Q Wu. A survey of consortium blockchain and its applications. *Cryptography*, 8(2):12, 2024.

[59] Weiquan Ni, Xumin Huang, Junxing Zhang, and Rong Yu. Healchain: A decentralized data management system for mobile healthcare using consortium blockchain. In *2019 Chinese Control Conference (CCC)*, pages 6333–6338. IEEE, 2019.

[60] Han Song, Zhongche Qu, and Yihao Wei. Advancing blockchain scalability: An introduction to layer 1 and layer 2 solutions. In *2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, pages 71–76. IEEE, 2024.

[61] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. Sok: Sharding on blockchain. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 41–61, 2019.

[62] Louis Tremblay Thibault, Tom Sarry, and Abdelhakim Senhaji Hafid. Blockchain scaling using rollups: A comprehensive survey. *IEEE Access*, 10:93039–93054, 2022.

[63] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. 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*, pages 201–226. Springer, 2020.

[64] Mahen Mandal, Mohd Sameen Chishti, and Amit Banerjee. Investigating layer-2 scalability solutions for blockchain applications. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 710–717. IEEE, 2023.

[65] Joseph Poon and Vitalik Buterin. Plasma: Scalable autonomous smart contracts. *White paper*, pages 1–47, 2017.

[66] Wenbing Zhao and Xiong Luo. Layer 2 blockchains: An introduction. In *Proceedings of the 2024 7th Artificial Intelligence and Cloud Computing Conference*, pages 449–456, 2024.

[67] Mindaugas Juodis, Ernestas Filatovas, and Remigijus Paulavičius. Overview and empirical analysis of wealth decentralization in blockchain networks. *ICT Express*, 10(2):380–386, 2024.

[68] Rui Zhang, Rui Xue, and Ling Liu. Security and privacy on blockchain. *ACM Computing Surveys (CSUR)*, 52(3):1–34, 2019.

[69] Amrita Jyoti and RK Chauhan. A blockchain and smart contract-based data provenance collection and storing in cloud environment. *Wireless Networks*, 28(4):1541–1562, 2022.

[70] Amritraj Singh, Reza M Parizi, Qi Zhang, Kim-Kwang Raymond Choo, and Ali Dehghantanha. Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Computers & Security*, 88:101654, 2020.

[71] Weilin Zheng, Zibin Zheng, Xiangping Chen, Kemian Dai, Peishan Li, and Renfei Chen. Nutbaas: A blockchain-as-a-service platform. *Ieee Access*, 7:134422–134433, 2019.

[72] Ruben Fernando Cuadros Mieses, Adolfo Jorge Prado Ventocilla, and Edwin Jorge Montes Eskenazy. From trust to code: A comparative performance analysis of ethereum and polygon for decentralized university research funding. In *2025 IEEE XXXII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–6. IEEE, 2025.

[73] Xigao Li, Anurag Yepuri, and Nick Nikiforakis. Double and nothing: Understanding and detecting cryptocurrency giveaway scams. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2023.

[74] Frank Hofmann, Simone Wurster, Eyal Ron, and Moritz Böhmecke-Schwafert. The immutability concept of blockchains and benefits of early standardization. In *2017 ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*, pages 1–8. IEEE, 2017.

[75] Christian Badertscher, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. Bitcoin as a transaction ledger: A composable treatment: C. badertscher et al. *Journal of Cryptology*, 37(2):18, 2024.

[76] Sergei Tikhomirov. Ethereum: state of knowledge and research perspectives. In *International symposium on foundations and practice of security*, pages 206–221. Springer, 2017.

[77] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. Blockchain based access control. In *IFIP international conference on distributed applications and interoperable systems*, pages 206–220. Springer, 2017.

[78] Jongbeen Han, Mansub Song, Hyeonsang Eom, and Yongseok Son. An efficient multi-signature wallet in blockchain using bloom filter. In *Proceedings of the 36th annual ACM symposium on applied computing*, pages 273–281, 2021.

[79] Yue Xiao, Peng Zhang, and Yuhong Liu. Secure and efficient multi-signature schemes for fabric: An enterprise blockchain platform. *IEEE Transactions on Information Forensics and Security*, 16:1782–1794, 2020.

[80] Anubha Jain and Emmanuel S Pilli. Libmultisig—a multisignature transaction library for bitcoin. In *International Conference on Intelligent Systems and Security*, pages 367–381. Springer, 2024.

[81] Susmita Mondal, Mehak Shafi, Sumeet Gupta, and Sachin Kumar Gupta. Blockchain based secure architecture for electronic healthcare record management. *GMSARN Int. J*, 16(4):413–426, 2022.

[82] Md Al Amin, Rushabh Shah, Hemanth Tummala, and Indrajit Ray. Utilizing blockchain and smart contracts for enhanced fraud prevention and minimization in health insurance through multi-signature claim processing. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pages 1–9. IEEE, 2024.

[83] Ehsan Nowroozi, Seyedsadra Seyedshoari, Yassine Mekdad, Erkay Savaş, and Mauro Conti. Cryptocurrency wallets: assessment and security. In *Blockchain for Cybersecurity in Cyber-Physical Systems*, pages 1–19. Springer, 2022.

[84] Sabine Houy, Philipp Schmid, and Alexandre Bartel. Security aspects of cryptocurrency wallets—a systematic literature review. *ACM Computing Surveys*, 56(1):1–31, 2023.

[85] Wei-Meng Lee. Using the metamask crypto-wallet. In *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, pages 111–144. Springer, 2023.

[86] Andleeb Khan, Parma Nand, Bharat Bhushan, Alaa Ali Hameed, and Akhtar Jamil. A review of blockchain based decentralised authentication solutions and their improvement through metamask. In *2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, pages 1–5. IEEE, 2024.

[87] Matheus VX Ferreira, Daniel J Moroz, David C Parkes, and Mitchell Stern. Dynamic posted-price mechanisms for the blockchain transaction-fee market. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pages 86–99, 2021.

[88] Lin Zhang, Brian Lee, Yuhang Ye, and Yuansong Qiao. Ethereum transaction performance evaluation using test-nets. In *European Conference on Parallel Processing*, pages 179–190. Springer, 2019.

[89] Reval Prabhu Puneeth and Govindaswamy Parthasarathy. Seamless data exchange: advancing healthcare with cross-chain interoperability in blockchain for electronic health records. *International Journal of Advanced Computer Science and Applications*, 14(10), 2023.

[90] Office for Civil Rights (OCR). Hipaa home, Aug 2023.

[91] Mohammed Shuaib, Shadab Alam, Mohammad Shabbir Alam, and Mohammad Shahnawaz Nasir. Compliance with hipaa and gdpr in blockchain-based electronic health record. *Materials Today: Proceedings*, 2021.

[92] Yangheran Piao, Kai Ye, and Xiaohui Cui. A data sharing scheme for gdpr-compliance based on consortium blockchain. *Future Internet*, 13(8):217, 2021.

[93] Zhou Wu, Andrew B Williams, and Debbie Perouli. Dependable public ledger for policy compliance, a blockchain based approach. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1891–1900. IEEE, 2019.

[94] AKM Bahalul Haque, Bilal Naqvi, AKM Najmul Islam, and Sami Hyrynsalmi. Towards a gdpr-compliant blockchain-based covid vaccination passport. *Applied Sciences*, 11(13):6132, 2021.

[95] Anton Hasselgren, Paul Kengfai Wan, Margareth Horn, Katina Kralevska, Danilo Gligoroski, and Arild Faxvaag. Gdpr compliance for blockchain applications in healthcare. *arXiv preprint arXiv:2009.12913*, 2020.

[96] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd international conference on open and big data (OBD)*, pages 25–30. IEEE, 2016.

[97] Zhe Xiao, Zengxiang Li, Yong Liu, Ling Feng, Weiwen Zhang, Thanarit Lertwuthikarn, and Rick Siow Mong Goh. Emrshare: A cross-organizational medical data sharing and management framework using permissioned blockchain. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 998–1003. IEEE, 2018.

[98] Peng Zhang, Jules White, Douglas C Schmidt, Gunther Lenz, and S Trent Rosenbloom. Fhirchain: applying blockchain to securely and scalably share clinical data. *Computational and structural biotechnology journal*, 16:267–278, 2018.

[99] Jens-Andreas Hanssen Rensaa, Danilo Gligoroski, Katina Kralevska, Anton Hasselgren, and Arild Faxvaag. Verifymed-a blockchain platform for transparent trust in virtualized healthcare: Proof-of-concept. In *Proceedings of the 2nd International Electronics Communication Conference*, pages 73–80, 2020.

[100] Ayesha Shahnaz, Usman Qamar, and Ayesha Khalid. Using blockchain for electronic health records. *IEEE access*, 7:147782–147795, 2019.

[101] André Henrique Mayer, Cristiano André da Costa, and Rodrigo da Rosa Righi. Electronic health records in a blockchain: A systematic review. *Health informatics journal*, 26(2):1273–1288, 2020.

[102] Hao Wang and Yujiao Song. Secure cloud-based ehr system using attribute-based cryptosystem and blockchain. *Journal of medical systems*, 42(8):152, 2018.

[103] Thomas Ploug and Søren Holm. Pharmaceutical information systems and possible implementations of informed consent-developing an heuristic. *BMC Medical Ethics*, 13(1):1–12, 2012.

[104] Qi Xia, Emmanuel Boateng Sifah, Abla Smahi, Sandro Amofa, and Xiaosong Zhang. Bbds: Blockchain-based data sharing for electronic medical records in cloud environments. *Information*, 8(2):44, 2017.

[105] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):218, 2016.

[106] Guy Zyskind, Oz Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184. IEEE, 2015.

[107] Kai Fan, Shangyang Wang, Yanhui Ren, Hui Li, and Yintang Yang. Medblock: Efficient and secure medical data sharing via blockchain. *Journal of medical systems*, 42(8):136, 2018.

[108] Dara Tith, Joong-Sun Lee, Hiroyuki Suzuki, WMAB Wijesundara, Naoko Taira, Takashi Obi, and Nagaaki Ohyama. Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology. *Healthcare Informatics Research*, 26(4):265–273, 2020.

[109] James Cunningham, Nigel Davies, Sarah Devaney, Søren Holm, Mike Harding, Victoria Neumann, and John Ainsworth. Non-fungible tokens as a mechanism for representing patient consent. *Studies in Health Technology and Informatics*, 294:382–386, 2022.

[110] Faisal Albalwy, Andrew Brass, Angela Davies, et al. A blockchain-based dynamic consent architecture to support clinical genomic data sharing (consentchain): Proof-of-concept study. *JMIR medical informatics*, 9(11):e27816, 2021.

[111] Mira Shah, Chao Li, Ming Sheng, Yong Zhang, and Chunxiao Xing. Crowdmed: A blockchain-based approach to consent management for health data sharing. In *Smart Health: International Conference, ICSH 2019, Shenzhen, China, July 1–2, 2019, Proceedings 7*, pages 345–356. Springer, 2019.

[112] Yan Zhuang, Lincoln R Sheets, Yin-Wu Chen, Zon-Yin Shae, Jeffrey JP Tsai, and Chi-Ren Shyu. A patient-centric health information exchange framework using blockchain technology. *IEEE journal of biomedical and health informatics*, 24(8):2169–2176, 2020.

[113] May Alhajri, Ahmad Salehi Shahraki, and Carsten Rudolph. Privacy of fitness applications and consent management in blockchain. *Proceedings of the 2022 Australasian Computer Science Week*, pages 65–73, 2022.

[114] Sandro Amofa, Emmanuel Boateng Sifah, O-B Kwame, Smahi Abla, Qi Xia, James C Gee, and Jianbin Gao. A blockchain-based architecture framework for secure sharing of

personal health data. In *2018 IEEE 20th international conference on e-Health networking, applications and services (Healthcom)*, pages 1–6. IEEE, 2018.

[115] Eugenio Balistri, Francesco Casellato, Carlo Giannelli, and Cesare Stefanelli. Blockhealth: Blockchain-based secure and peer-to-peer health information sharing with data protection and right to be forgotten. *ICT Express*, 7(3):308–315, 2021.

[116] Bingqing Shen, Jingzhi Guo, and Yilong Yang. Medchain: Efficient healthcare data sharing via blockchain. *Applied sciences*, 9(6):1207, 2019.

[117] Yang Yang, Ximeng Liu, and Robert H Deng. Lightweight break-glass access control system for healthcare internet-of-things. *IEEE Transactions on Industrial Informatics*, 14(8):3610–3617, 2017.

[118] Melissa Loos. Break-glass access control systems in medical devices. *RTDS, WS 2020, Institute of Distributed Systems, Ulm University*, 2020. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

[119] Vidyadhar Aski, Vijaypal Singh Dhaka, and Anubha Parashar. An attribute-based break-glass access control framework for medical emergencies. In *Innovations in Computational Intelligence and Computer Vision: Proceedings of ICICV 2020*, pages 587–595. Springer, 2021.

[120] Sigrid Schefer-Wenzl, Helena Bukvova, and Mark Strembeck. A review of delegation and break-glass models for flexible access control management. In *Proceedings of the International Conference on Security and Trust Management*, pages 1–12. University of Applied Sciences Campus Vienna and WU Vienna, Austria, Springer, 2013.

[121] Dries Van Bael, Shirin Kalantari, Andreas Put, and Bart De Decker. A context-aware break glass access control system for iot environments. In *7th International Conference on Internet of Things: Systems, Management, and Security (IOTSMS)*, pages 20–27. IEEE, 2020.

[122] José A García-Berná, Raimel Sobrino-Duque, Juan M Carrillo de Gea, Joaquín Nicolás, and José L Fernández-Alemán. Automated workflow for usability audits in the phr realm. *International Journal of Environmental Research and Public Health*, 19(15):8947, 2022.

[123] Jovan Stevovic, Fabio Casati, Bilal Farraj, Jun Li, Hamid R. Motahari-Nezhad, and Giampaolo Armellin. Compliance aware cross-organization medical record sharing. *IEEE Symposium on Integrated Network Management*, 2023.

[124] Dae-young Kim, Lavanya Elluri, and Karuna P. Joshi. Trusted compliance enforcement framework for sharing health big data. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 4715–4724, Baltimore, MD, USA, 2021. IEEE.

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

[126] Shezon Saleem Mohammed Abdul. Navigating blockchain's twin challenges: Scalability and regulatory compliance. *Blockchains*, 2(3):265–298, 2024.

[127] Joseph V Pergolizzi, Frederick A Curro, Nanada Col, Mary Papa Ghods, Don Vena, Robert Taylor, Frederick Naftolin, and Jo Ann LeQuang. A multicentre evaluation of an opioid patient–provider agreement. *Postgraduate medical journal*, 93(1104):613–617, 2017.

[128] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, 800(162):1–54, 2013.

[129] Shinhae Kim and Sungjae Hwang. 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*, pages 1333–1344, 2023.

[130] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[131] Giuseppe Antonio Pierro and Henrique Rocha. The influence factors on ethereum transaction fees. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 24–31. IEEE, 2019.

[132] Harry Kalodner, Steven Goldfeder, Xiaoqi Chen, S Matthew Weinberg, and Edward W Felten. Arbitrum: Scalable, private smart contracts. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1353–1370, 2018.

[133] Clemens Scott Kruse, Michael Mileski, Alekhya Ganta Vijaykumar, Sneha Vishnampet Viswanathan, Ujwala Suskandla, and Yazhini Chidambaram. Impact of electronic health records on long-term care facilities: systematic review. *JMIR medical informatics*, 5(3):e7958, 2017.

[134] Shona Kalkman, Johannes van Delden, Amitava Banerjee, Benoît Tyl, Menno Mostert, and Ghislaine van Thiel. Patients' and public views and attitudes towards the sharing of health data for research: a narrative review of the empirical evidence. *Journal of medical ethics*, 48(1):3–13, 2022.

[135] Gill Haddow, Ann Bruce, Shiva Sathanandam, and Jeremy C Wyatt. 'nothing is really safe': a focus group study on the processes of anonymizing and sharing of health data for research purposes. *Journal of evaluation in clinical practice*, 17(6):1140–1146, 2011.

[136] Edmond Li, Jonathan Clarke, Ana Luisa Neves, Hutan Ashrafian, and Ara Darzi. Protocol: Electronic health records, interoperability and patient safety in health systems of high-income countries: A systematic review protocol. *BMJ Open*, 11(7), 2021.

[137] Samuel D Lustgarten, Yunkyoung L Garrison, Morgan T Sinnard, and Anthony WP Flynn. Digital privacy in mental healthcare: current issues and recommendations for technology use. *Current opinion in psychology*, 36:25–31, 2020.

[138] Elizabeth Hutchings, Max Loomes, Phyllis Butow, and Frances M Boyle. A systematic literature review of attitudes towards secondary use and sharing of health administrative and clinical trial data: a focus on consent. *Systematic Reviews*, 10:1–44, 2021.

[139] Stefan Timmermans. The engaged patient: The relevance of patient–physician communication for twenty-first-century health. *Journal of Health and Social Behavior*, 61(3):259–273, 2020.

[140] Antonio Lopez Martinez, Manuel Gil Pérez, and Antonio Ruiz-Martínez. A comprehensive review of the state-of-the-art on security and privacy issues in healthcare. *ACM Computing Surveys*, 55(12):1–38, 2023.

[141] Malak Aljabri, Maryam Aldossary, Noor Al-Homeed, Bushra Alhetelah, Malek Althubiany, Ohoud Alotaibi, and Sara Alsaqer. Testing and exploiting tools to improve owasp top ten security vulnerabilities detection. In *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 797–803. IEEE, 2022.

[142] Kyusuk Chung, Dalsang Chung, and YangHee Joo. Overview of administrative simplification provisions of hipaa. *Journal of medical systems*, 30:51–55, 2006.

[143] Mauro Lemus Alarcon, Minh Nguyen, Saptarshi Debroy, Naga Ramya Bhamidipati, Prasad Calyam, and Abu Mosa. 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)*, pages 201–206. IEEE, 2021.

[144] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[145] Tuan-Vinh Le and Chien-Lung Hsu. A systematic literature review of blockchain technology: Security properties, applications and challenges. *Journal of Internet Technology*, 22(4):789–802, 2021.

[146] Shangping Wang, Dongyi Li, Yaling Zhang, and Juanjuan Chen. Smart contract-based product traceability system in the supply chain scenario. *IEEE Access*, 7:115122–115133, 2019.

[147] Dieudonne Mulamba and Indrajit Ray. Resilient reference monitor for distributed access control via moving target defense. In *Data and Applications Security and Privacy XXXI: 31st Annual IFIP WG 11.3 Conference, DBSec 2017, Philadelphia, PA, USA, July 19-21, 2017, Proceedings 31*, pages 20–40. Springer, 2017.

[148] Elvira Albert, Jesús Correas, Pablo Gordillo, Guillermo Román-Díez, and Albert Rubio. Gasol: Gas analysis and optimization for ethereum smart contracts. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 118–125. Springer, 2020.

[149] Nir Menachemi and Taleah H Collum. Benefits and drawbacks of electronic health record systems. *Risk management and healthcare policy*, pages 47–55, 2011.

[150] José Luis Fernández-Alemán, Inmaculada Carrión Señor, Pedro Ángel Oliver Lozoya, and Ambrosio Toval. Security and privacy in electronic health records: A systematic literature review. *Journal of biomedical informatics*, 46(3):541–562, 2013.

[151] Ana Ferreira, Ricardo Cruz-Correia, Luis Antunes, Pedro Farinha, E Oliveira-Palhares, David W Chadwick, and Altamiro Costa-Pereira. How to break access control in a controlled manner. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pages 847–854. IEEE, 2006.

[152] Daniel Conte de Leon, Antonius Q Stalick, Ananth A Jillepalli, Michael A Haney, and Frederick T Sheldon. Blockchain: properties and misconceptions. *Asia Pacific Journal of Innovation and Entrepreneurship*, 11(3):286–300, 2017.

[153] Nurzhan Zhumabekuly Aitzhan and Davor Svetinovic. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE transactions on dependable and secure computing*, 15(5):840–852, 2016.

[154] Ankit Gangwal, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. A survey of layer-two blockchain protocols. *Journal of Network and Computer Applications*, 209:103539, 2023.

[155] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph databases: new opportunities for connected data*. " O'Reilly Media, Inc.", 2015.

[156] Vincent C Hu, D Richard Kuhn, David F Ferraiolo, and Jeffrey Voas. Attribute-based access control. *Computer*, 48(2):85–88, 2015.

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

[158] Cyril Naves Samuel, Severine Glock, François Verdier, and Patricia Guitton-Ouhamou. Choice of ethereum clients for private blockchain: Assessment from proof of authority perspective. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–5. IEEE, 2021.

[159] Divakaran Liginlal, Inkook Sim, Lara Khansa, and Paul Fearn. Hipaa privacy rule compliance: An interpretive study using norman's action theory. *Computers & Security*, 31(2):206–220, 2012.

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

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

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

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

[164] John Ugoani and Grace Iyi Ibeenwo. External audit process failures: Unethical practices and business demise. *Business, Management and Economics Research*, 8(1):1–11, 2022.

[165] Puzant Balozian and Dorothy Leidner. 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):11–43, 2017.

[166] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J Knottenbelt. Sok: Communication across distributed ledgers. In *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part II 25*, pages 3–36. Springer, 2021.

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

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

[169] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, 22(2):1432–1465, 2020.

[170] Hong Wu and Guan Zheng. Electronic evidence in the blockchain era: New rules on authenticity and integrity. *Computer Law & Security Review*, 36:105401, 2020.

[171] The Interaction between Blockchain Evidence and Courts – A cross-jurisdictional analysis — blockgeeks.com. https://blockgeeks.com/guides/blockchain-evidence/. [Accessed 25-11-2023].

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