Blockchain-based Patient Centric Healthcare Architecture: Secure Medical Data Sharing

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

DOCTOR OF PHILOSOPHY

Submitted by

DEEPAK KUMAR MISHRA

(2K21/PHDCO/03)

Under the supervision of

DR. PAWAN SINGH MEHRA

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

February, 2026

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Deepak Kumar Mishra (2K21/PHDCO/03) Research Scholar (Department of Computer Science and Engineering), hereby declare that the thesis titled **"Blockchain-based Patient Centric Healthcare Architecture: Secure Medical Data Sharing"**, which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, in partial fulfillment of the requirement for the award of Doctorate of Philosophy is original and not copied from any source without proper citation. This work has not previously formed the basis for awarding any Degree, Diploma, Associateship, Fellowship or other title or recognition.

Place: Delhi Deepak Kumar Mishra

Date:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that the work entitled **"Blockchain-based Patient Centric Healthcare Architecture: Secure Medical Data Sharing"**, which is being submitted by **Deepak Kumar Mishra, Roll No. 2K21/PHDCO/03** to the Department of Computer Science and Engineering, Delhi Technological University for the award of the degree of Doctor of Philosophy is a record bona fide research work carried out by him under my supervision and guidance. He has fulfilled the requirements for the submission of this thesis. The contents of this thesis, in whole or in parts, have not been submitted for any other degree or diploma.

Place: Delhi — Dr. Pawan Singh Mehra

Date: — **SUPERVISOR**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

Be grateful when things go good and graceful when they go bad; as it is said, I feel immensely grateful when it comes to acknowledging my thanks to those who helped me complete my thesis successfully. At the onset, I bowed my head to the almighty God, who led me on this path, as everything has been possible with his blessings.

I take great pleasure in expressing my gratitude with profound respect to my revered supervisor, **Dr. Pawan Singh Mehra**, Assistant Professor, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, for his invaluable guidance, close supervision, untiring, ever-ready help and discussions throughout my Ph.D. work. I am indebted to him for his constructive criticism, exemplary patience, perseverance, motivation, and immense knowledge, which have gone a long way towards completing this thesis. His constant inspiration gave me considerable impetus to achieve this milestone. I could not have imagined having a better supervisor and mentor for my Ph.D. study.

I extend my gratitude to **Prof.(Dr.) Anil Singh Parihar** , Head, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, for his support and guidance in carrying out this research work.

I also extend my heartiest thanks to the **Departmental Research Committee** for monitoring the progress and providing priceless suggestions encouraging me to widen my research from various perspectives.

I am also deeply grateful to all the members of the Department of Computer Science and Engineering, Delhi Technological University, for their constant help and providing all the necessary research resources.

I am also thankful to all academic, administrative and technical staff of Delhi Technological University for providing me with the university's resources and the necessary laboratory and research facilities for carrying out this work throughout my candidature.

A special thanks to my family members for their love, encouragement, patience and trust. With God's grace, this work allows me to make my parents, **Mr. Vinod Kumar Mishra** and **Mrs. Neeraj Mishra**, feel proud. I am also thankful to my wife, **Mrs. Malya Mishra** and my daughters, **Mahika, Vamika**. Their prayers for me were what sustained me in reaching this milestone.

Place: Delhi

Deepak Kumar Mishra

2K21/PHDCO/03

Abstract

The rapid digitization of the healthcare sector is bringing forward new and serious concerns regarding patient confidentiality, patient data protection and integrity, data integration/interoperability and patient consent management. The traditional healthcare architecture models that have been used for decades are centralized, fragmented and insecure and are no longer able to meet the changing demands and the associated risk factors such as identity theft, violations of confidentiality and inefficiency in managing data. Addressing these concerns, this research systematically explores patient-centric blockchain-based healthcare architectures that provide secure, transparent, decentralized and patient-controlled data management.

With an extensive literature review, this thesis identifies present research trends and gaps in blockchain-based healthcare solutions, showing the need for better security measures and consent-management systems with higher interoperability standards. This thesis proposes DiabeticChain, a novel blockchain-based solution made for managing diabetic healthcare data, improving patient control, data security and privacy using smart contracts, dynamic consent management and decentralized storage solutions. The proposed architecture integrates HL7/FHIR for semantic interoperability, modern cryptography for confidentiality and fine-grained access control, and a scalable Layer-2 blockchain plus decentralized storage to support real-time healthcare scenarios.

Additionally, TotalSol, a novel multi-layer static analysis framework, is introduced to detect vulnerabilities in Ethereum-based smart contracts. The tool enhances the robustness and reliability of blockchain applications, by pointing critical security flaws. Because the proposed healthcare architecture relies on Ethereum smart contracts to enforce consent and access policies, TotalSol is developed to analyse these contracts and detect vulnerabilities before deployment. Using Hyperledger Caliper and comparative studies, our experiments show that DiabeticChain outperforms the selected blockchain-based healthcare frameworks on the evaluated performance metrics (throughput and latency). Fur-

thermore, the system considers ethical factors like patient data ownership and compliance with major regulations such as GDPR and HIPAA, combining its technical advancements with legal and ethical healthcare standards.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Abbreviation	Description	Abbreviation	Description
ABE	Attribute-Based Encryption	AES	Advanced Encryption Standard
AMSC	Access Management Smart Contract	API	Application Programming Interface
APSC	Access Policy Smart Contract	AST	Abstract Syntax Tree
CID	Content Identifier	CGM	Continuous Glucose Monitor
CPU	Central Processing Unit	DApp	Decentralized Application
DAO	Decentralized Autonomous Organization	DID	Decentralized Identifier
DLT	Distributed Ledger Technology	DoS	Denial of Service
EHR	Electronic Health Record	EMR	Electronic Medical Record
EOSIO	Public-permissioned blockchain platform	FBG	Fasting Blood Glucose
FHIR	Fast Healthcare Interoperability Resources	GCM	Galois/Counter Mode
GDPR	General Data Protection Regulation	GPU	Graphics Processing Unit
HIPAA	Health Insurance Portability and Accountability Act	HL7	Health Level Seven (standards body)
IPFS	InterPlanetary File System	IoMT	Internet of Medical Things
IoT	Internet of Things	JSON	JavaScript Object Notation
MDSC	Metadata Smart Contract	PBFT	Practical Byzantine Fault Tolerance

Abbreviation	Description	Abbreviation	Description
PoA	Proof of Authority	PoB	Proof of Burn
PoET	Proof of Elapsed Time	PoS	Proof of Stake
PoW	Proof of Work	PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RAM	Random Access Memory	R1CS	Rank-1 Constraint System
RBE	Role-Based Encryption	RPC	Remote Procedure Call
ROR	Real-or-Random (security model)	RSA	Rivest–Shamir–Adleman
SGX	Software Guard Extensions (Intel)	SUT	System Under Test
TPS	Transactions per second	UI	User Interface
USC	User Smart Contract	VM	Virtual Machine
W3C	World Wide Web Consortium	ZKP	Zero-Knowledge Proof
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge		

Chapter 1

Introduction

1.1 Overview

Healthcare is becoming increasingly patient-centered. People should be able to view, use, and share their health data in ways that are private and secure. In practice, however, custody and control usually sit with central organizations such as hospitals, laboratories, and software vendors. Electronic Health Records (EHRs) made information easier to store, search, and transport, but they did not change the underlying power relationship. Institutions still decide who can access patient information and how it can be used, while long-standing concerns about privacy, confidentiality, and security remain [1–3].

These risks are not theoretical. In February 2023, Lehigh Valley Health Network (LVHN) disclosed a BlackCat/ALPHV ransomware attack. When LVHN did not pay, attackers released sensitive oncology images and documents, showing how centrally held clinical data can be used against patients [4]. Earlier, on 23 November 2022, the All India Institute of Medical Sciences (AIIMS), New Delhi, suffered a severe ransomware attack that forced a manual fallback. Most e-Hospital functions returned after about two weeks, and five physical servers running the e-Hospital application were impacted [5]. Beyond operational disruption, incidents like these erode public trust and increase the human cost of institutional custody of personal health data, including identity theft, coercion, and stigma.

Patients also move through multiple providers and settings over time, including hospitals, clinics, and physician offices. As a result, a single person's health record is often scattered across EHR systems, laboratories, pharmacies, and registries, and the data can become misaligned. Sharing information across organizations is brittle and still relies on manual steps. Each exchange often triggers new consent paperwork or custom agreements, which increases delays and the likelihood of mistakes [1, 2].

At the same time, networked sensors and consumer devices now generate large volumes of high-frequency health signals that have real clinical value. In diabetes care, continuous glucose monitors (CGMs), insulin pumps, smart pens, and wearable devices produce rich time-series data. When combined with laboratory results such as HbA1c and medication information, these streams can support safer medication titration, reduce hypoglycemia risk, and improve outcomes [6, 7]. Yet these signals are also highly sensitive. If they are mishandled, the consequences can be long-lasting for individuals and families, including discrimination and social stigma [8].

Taken together, security breaches, operational silos, and governance challenges motivate a shift away from institution-based custodianship toward truly patient-centered stewardship.

A patient-centered platform must deliver three capabilities at once:

1. patient-authenticated control over who can access which data and when,
2. responsible exchange across organizations with an auditable, tamper-evident record,
3. interoperability that spans both clinical systems and patient-generated data, without sacrificing privacy.

Traditional hub-and-spoke models struggle to provide all three at scale in real-world environments [9, 10].

In this work, we explore a careful application of blockchain to provide the consent, provenance, and audit layer for patient control, while keeping Protected Health Information (PHI) off-chain. When used appropriately, distributed ledgers reduce dependence on any single institution, allow consent to be expressed as machine-readable policy, and support end-to-end traceability of access events [11–14]. Crucially, we do not place PHI on the ledger. Instead, PHI remains encrypted in secure off-chain storage. The chain stores verifiable pointers, consent status, provenance proofs, and time-limited access tokens. This approach preserves an immutable audit trail while still supporting corrections, redactions, and dynamic policy changes, including revocation [15, 16].

Diabetes is a classic example of a lifelong, data-rich disease. Effective and safe care depends on integrating longitudinal EHR information, including comorbidities, medications, diagnoses, and labs, with device streams from CGMs and insulin delivery systems. Evidence and guidelines increasingly support CGM use to improve glycaemic outcomes and reduce hypoglycaemia in people with type 1 diabetes and some type 2 diabetes populations [6, 7]. We use diabetes as a recurring example because it forces the system to address practical integration hurdles across clinical and consumer data. It also highlights everyday fine-grained consent requirements, such as sharing CGM trends but not raw minute-level traces. Finally, it reflects the latency and reliability constraints encountered by clinicians and patients in routine care.

Our system treats patients as cryptographic principals who can issue fine-grained, time-bounded permissions for specific datasets and time intervals. Smart contracts encode consent into actionable policies. Off-chain services enforce those policies by releasing only the minimum necessary information, or by providing computed aggregates, to authorized requesters. Policy updates and access events are recorded immutably, enabling both real-time visibility and post hoc auditing [11, 12]. For interoperability, we translate clinical payloads into HL7/FHIR resources when feasible, so that consented exchanges map onto widely used data models and APIs [9, 10]. The result is a consent-and-audit layer that addresses key weaknesses of traditional designs, including single points of failure, invisible data flows, and untraceable disclosures, while remaining compatible with privacy principles such as data minimisation and purpose limitation, and with operational needs such as low latency, reliability, and ease of integration.

The following sections present the motivation and research gaps, define the research objectives, outline the main contributions, and provide an overview of the thesis structure.

1.2 Research Motivation

Accelerated digitization has augmented both the worth and the risk of medical information. Legacy, institution-based systems continue to find it hard to ensure end-to-end

integrity, safe sharing and nuanced consent [17, 18], while diverse providers and environments disintegrate records and make longitudinal decision-making challenging [12, 19]. Interoperability among these silos continues to be an ongoing hindrance [9]. At the same time, wearables and sensors create high-frequency data that might enhance personalization and results but have uneven governance and integration [20, 21]. Since health data frequently carries very sensitive properties, inappropriate handling can cause irreparable harm in the form of stigma, discrimination and loss of privacy [8].

The risks are highlighted by recent ransomware attacks: Lehigh Valley Health Network's BlackCat/ALPHV attack, followed by publication of confidential clinical content [4, 22], and the AIIMS New Delhi hack, which compelled a multi-week switch to manual handling [5]. In addition to disruptions to operations, these incidents undermine confidence and showcase the vulnerability of centralized custody.

This thesis is based on the need to develop data sharing models that support the requirements of both verifiable access for patients and their caregivers as well as accountable data exchanges with a tamper-evident history of all exchanges across organizational boundaries. Additionally, this thesis supports the development of data sharing models that will allow for practical interoperability across all clinical systems including data collected from patients. Blockchain presents a promising basis, decentralization, immutability and auditability, when deployed cautiously with protected health information stored off-chain and the ledger reserved for verifiable pointers, consent state and provenance proofs [23, 24]. We target diabetes care as a challenging testbed, seeking to show safe, accountable and interoperable sharing that is practicable in live clinical settings.

1.3 Research Gaps

A survey of patient-centric, blockchain-based healthcare identifies four enduring gaps that correspond to issues raised in previous research.

RG1: End-to-end architecture (security and scale and interoperability, all together): Most proposals tackle one aspect—secure storage, consent, or exchange, but few provide all three properties together under realistic load and across heterogeneous systems and standards. Cross-organizational interoperability at scale and clinically informed evaluation remain sparse evidences [13, 25–30].

RG2: Healthcare-specific smart-contract assurance: Platform selection (e.g., enterprise Ethereum clients and oracle designs) is addressed generally, but strong, health-data–specific assurance for on-chain logic is underdeveloped. Threat models for consent bugs, revocation edge cases, and misuse of external data sources (oracles) are seldom formalised for clinical workflows [31, 32].

RG3: Operational consent identification and management: Fine-grained, dynamic consent—who/what/why/when—with time limits, minimum-necessary disclosure, provenance, and revocation is variably specified and enforced. Mechanisms to record, verify, and audit consent between institutions and devices are still nascent [33–35].

RG4: Compliance, governance, and real-world evaluation: Compliance with healthcare regulation and governance (ownership frameworks, audit trails, data minimisation) is sometimes claimed but not evidenced. Most systems have no deployment evidence beyond small pilots or laboratory environments, with poor reporting on usability, reliability, and integration overheads [27, 33, 36].

These deficits underpin our patient-centric architecture and assessment agenda: verifiable control to patients, accountable exchange across boundaries, and pragmatic inter-

operability covering EHR and patient-generated data.

1.4 Research Objectives

The objective of this thesis is to identify the feasibility of blockchain for secure patient-centric medical data sharing (e.g. diabetic data), focusing on security, privacy, interoperability, scalability, consent management and vulnerability mitigation to allow safe and effective integration into healthcare systems. This objective can be achieved at several levels as follows:

1. **RO1:** To conduct a systematic literature survey on blockchain based patient centric healthcare systems.

2. **RO2:** To develop a model for managing patients' consent preferences and assessing their confidence in the consent architecture.

3. **RO3:** To develop a blockchain based model for security and privacy in patient centric healthcare data.

4. **RO4:** To develop an interoperable and scalable architecture for a blockchain based patient centric healthcare system.

5. **RO5:** To develop a method/model for vulnerability detection in Ethereum based smart contracts.

These research objectives are addressed across the thesis chapters as follows. RO1 is fulfilled through the systematic literature survey presented in Chapter 3. RO2 and RO3 are addressed through the design and evaluation of DiabeticChain in Chapter 4. RO4 is achieved via the enhanced DiabeticChain architecture described in Chapter 5. Finally, RO5 is addressed through the TotalSol vulnerability-detection framework presented in Chapter 6.

1.5 Contributions

This section provides a comprehensive overview of the major contributions of this thesis.

1. **Exploring Blockchain-based Patient-Centric Healthcare: A Comprehensive Review**
   This thesis provides a detailed systematic literature review (SLR) of the current scenario of blockchain and smart contract technologies for patient centric healthcare applications, which is given in Chapter 3. In particular, the following have been analyzed:

   a. Current existing blockchain platforms, types, consensus algorithms, smart contracts and IT tools that have been employed for developing blockchain based patient centric healthcare applications;

   b. State-of-the-art blockchain use cases in the patient-centric healthcare field;

c. The benefits and challenges of the use of blockchain in the patient-centric healthcare field, with emphasis on data security, privacy, interoperability, and scalability;

d. The technical challenges and barriers impeding the sharing.

2. **DiabeticChain: A Novel Blockchain Approach for Patient-Centric Diabetic Data Management**
   This thesis introduces "DiabeticChain", a patient-centric, permissioned architecture for safe sharing of diabetic records using smart contracts, addressing the interoperability and consent-management gaps. Confidential clinical information stays off-chain and verifiable metadata and consent statuses stay on-chain to enable traceability and auditing. The design and analysis are described in Chapter 4. Specifically, the following have been designed and analyzed:

   a. Hybrid on/off-chain design and workflow; off-chain storage of protected health information with on-chain references, consent state and provenance to reduce exposure while maintaining accountability;

   b. Smart-contract components and consent operations; contracts for user/role management, policy specification, metadata registration, access request/approval, revocation and audit logging;

   c. Deployment and implementation stack; contract testing and development standard tooling, and an oracle/off-chain data service path to settle authorised requests;

   d. Performance and security assessment; latency/throughput benchmarking in rising loads, verifications for data-minimisation, revocation trends and audit trails with tamper-evident support.

3. **Enhancing Interoperability and Scalability in DiabeticChain: An Integrative Architecture Using IPFS, Polygon, and HL7/FHIR**
   This thesis extends DiabeticChain with a standards-compliant, privacy-enforcing design that scales execution and facilitates semantic interoperability, filling the lack of scalable, performance-evaluated blockchain architectures for healthcare. IPFS yields content-addressed storage, Polygon high-throughput execution, and HL7/FHIR mappings standardize data exchange between heterogeneous systems. The extended design is described in Chapter 5. In particular:

   a. Layered architecture and stack; clear separation of application, interoperability, smart-contract, storage and security layers to support modular evolution and deployment;

   b. Privacy-preserving consent enforcement via zk-SNARKs (Groth16); off-chain proof generation with on-chain verification to authorize access without revealing sensitive consent attributes;

   c. Hybrid encryption and IPFS confidentiality; encrypt records prior to persistence (AES with RSA/CP-ABE for key protection), store ciphertext on IPFS, and anchor only CIDs with minimal policy/consent state on-chain;

   d. Interoperability and evaluation stance; HL7/FHIR mapping and validation pipeline for heterogeneous clinical sources, plus benchmarking and design choices enabling auditability, purpose limitation and data minimisation.

4. **TotalSol: A Multi-Layer Static Analysis Method for Vulnerability Detection in Ethereum-based Smart Contracts**
   This thesis develops "TotalSol", a multi-layer static-analysis framework that orchestrates complementary analyzers to improve vulnerability coverage and reduce false negatives when auditing Solidity smart contracts at scale, supporting secure deployment of smart-contract logic in healthcare systems (Chapter 6). Specifically:

   a. Unified, extensible framework; a modular runner with a shared input/interface to execute multiple analyzers over the same corpus and scale to large contract sets;

   b. Multi-analyzer ensemble; parallel use of complementary tools (e.g., Slither, Aderyn, SmartCheck) with a consolidation step to reconcile outputs and strengthen first-pass detection;

   c. Mainnet-derived dataset pipeline; automated retrieval of verified contracts, JSON-to-`.sol` parsing, pragma/version fixes, compilation checks and filtering to produce a clean benchmark set;

   d. Healthcare assurance and impact; auditing the DiabeticChain contracts (e.g., registration, consent-policy and access-control modules) and producing one deduplicated report that surfaces high-impact issues such as reentrancy, arithmetic and access-control weaknesses.

1.6 Outline of the Thesis

The organization of this thesis is as follows.

The visual representation of the thesis structure is shown in Figure 1.1 and outline of the chapters is as follows:

Chapter 2 presents the foundational knowledge important for understanding the technologies used in the thesis. It explains the core technical concepts, including the blockchain, algorithms, tools and internal architecture used. This chapter covers all preliminary knowledge required to understand the following chapters in the thesis.

Chapter 3 presents a systematic literature review of blockchain-based, patient-centric healthcare architectures. Based on four research questions—security and privacy, consent management, interoperability and regulatory compliance, it identifies key gaps and outlines future directions toward scalable, privacy-preserving architectures that strengthen patient-controlled data sharing.

Chapter 4 introduces "DiabeticChain", a permissioned, patient-centric blockchain architecture that restores patient data ownership and enables secure sharing via smart contracts. A hybrid storage design keeps confidential diabetic data in an off-chain secure database while recording only metadata on-chain, preserving integrity, confidentiality and authenticity.

Chapter 5 illustrates the enhanced DiabeticChain architecture by adding Polygon (a layer-2 Ethereum scaling solution) for large-scale smart contract execution, IPFS for

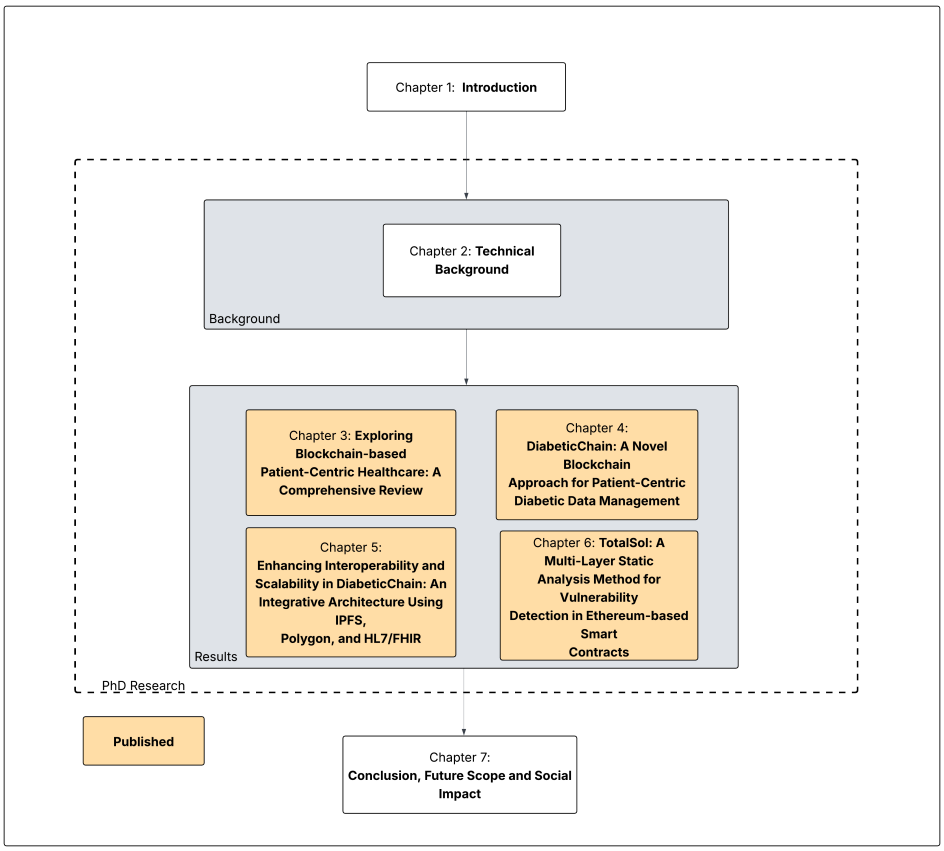


Figure 1.1: Overall Structure of the Thesis

tamper-resistant and decentralized storage, HL7/FHIR for semantic interoperability, and zk-SNARK enabled consent management for privacy-preserving access control. Using hybrid encryption (AES-256-GCM + ABE/RSA), it provides confidentiality from source to destination and stores FHIR resources off-chain with content identifiers stored on-chain.

Chapter 6 presents "TotalSol", a novel analytical tool designed to identify vulnerabilities within smart contract systems based upon Ethereum. TotalSol employs multi-classification techniques to identify multiple vulnerabilities that may be present in smart contracts. It analyzes the smart contracts used in our healthcare architecture to improve their security. Chapter 6 describes many of the most common types of vulnerabilities (reentrancy, integer overflows or underflows, etc.) as well as provides a comprehensive review of the effectiveness of TotalSol in improving overall blockchain security.

Chapter 7 summarizes the findings of this dissertation and discusses the main contributions it has made toward increasing security, privacy, interoperability and scalability within a blockchain-based model of healthcare. It also discusses the importance and social implications of the proposed solutions as well as some of the limitations of the proposed solutions and provides direction for future research.

Chapter 2

Preliminaries

This chapter introduces the foundational tools and technologies that are used in the following chapters of this thesis. It is designed to provide background knowledge for better understanding the implementation details and evaluations that follow.

2.1 Overview of Tools and Technologies

This chapter explains and discusses the major tools and technologies that form the technical basis for the research work covered in the following chapters of this thesis. Due to the interdisciplinary and applied nature of blockchain technologies in healthcare, it is important to have a good understanding of these tools to understand the implementation and evaluation details that follow. In particular, we introduce an overview of necessary technologies such as smart contract development platforms, blockchain deployment strategies, oracle integrations, decentralized and off-chain storage solutions, performance evaluation tools, and interoperability standards. Each section briefly outlines the technical relevance and functional contribution of these tools towards the achievement of goals associated with secure, efficient, and interoperable management of healthcare data in blockchain-based healthcare systems.

2.2 Background of Blockchain

Blockchain technology represents a decentralized and distributed database system that maintains a secure and unalterable record of transactions. Satoshi Nakamoto first introduced it in 2008 as a peer-to-peer electronic cash system [37]. Blockchain consists of a series of blocks, each of which ensures the chronological order of transactions by including a timestamp and a link to preceding blocks [38]. Figure 2.1 illustrates a series of interconnected blocks in a blockchain. Each block consists of a header and a body. The body contains a Merkle tree with hashes of all the data nodes. The header includes a timestamp, a nonce, a hash pointer to the prior block, a merkle root, a transaction count, and a version number, establishing the basis for the blockchain's structure. Due to its ability to provide a platform for direct interaction between parties involved in the process, and thus remove the necessity for an intermediary, this technology has gained a lot of interest [21]. In addition, the properties of blockchain including decentralization, immutability, and cryptographic consensus, have generated a wide array of possible applications in areas including finance, education, voting, and healthcare.

A blockchain provides assurance of the safe storage of transactions and tracking of

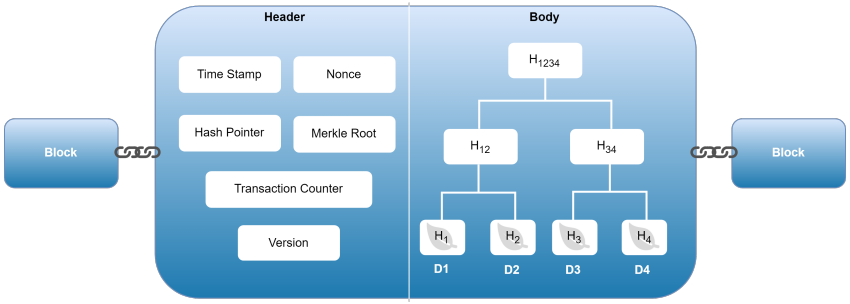


Figure 2.1: Structure of the Blockchain

assets due to its status as a shared and immutable ledger [39]. Nodes which have been authorized verify and authenticate transactions within the system, therefore, no need exists for an intermediate entity to be trusted [40]. Due to its tamper-proof characteristics, blockchain is well suited to manage and ensure the integrity of patient health data [39, 41].

The operation of a blockchain is based on a distributed ledger and a peer-to-peer network [42]. Each transaction is recorded in a block in the distributed ledger, and every node in the network holds a replica (a duplicate) of the shared ledger locally to prevent unauthorized modification of the data [42]. Each transaction must undergo validation and verification through consensus algorithms before it can enter the network; thereby, ensuring that the data is accurate. A transaction once it has entered into a blockchain, becomes permanent and unalterable, thus establishing an unchangeable history of transactions and any alteration attempts will establish an identifiable footprint [39].

2.3 Architecture of Blockchain

The architecture of a blockchain includes two primary components: a distributed database and a network of peer-to-peer participants. Figure 2.2 illustrates the architecture of a blockchain as a peer-to-peer network and how it differs from the traditional centralized network.

- Distributed Database – The transactions within the blockchain are recorded in a ledger accessible to all participating nodes, eliminating the possibility of some malicious node intervening and manipulating the records. Every node has an identical local copy of that shared ledger to ensure the data integrity. To append a new transaction to a block in the blockchain, it is first verified and authenticated by all nodes using a consensus algorithm, then appended to the network only if majority of the nodes validate the transaction. Once this change is made it cannot be reverted. No single node has the authority to manipulate the data existing in the blockchain.

- Peer to peer network – In a blockchain network, the nodes are connected instead of one central node, making the entire network decentralized and every transaction is authorized before adding it to the network, making the network authentic [25]. Every node has the equal power to consume the services of the network and agree upon them via a consensus algorithm. Peer-to-peer networks eliminate the problem of a single point of failure due to absence of a central node, resulting in a more secure network [26].

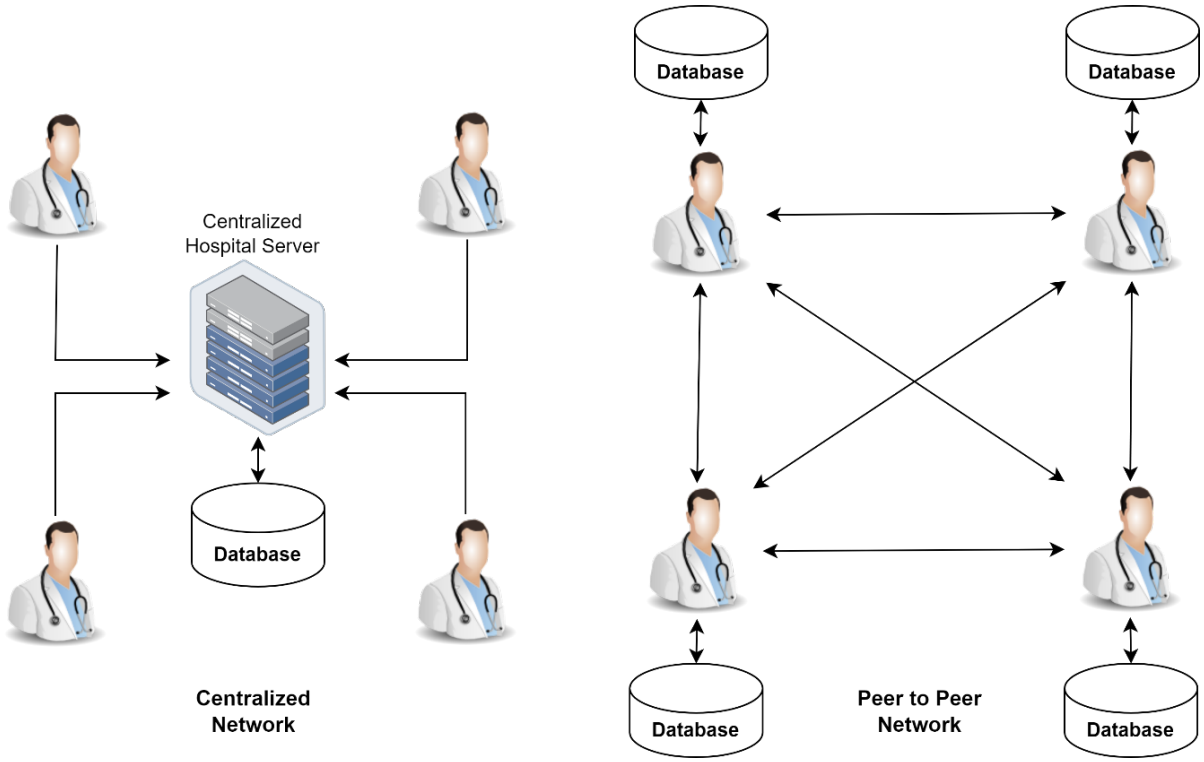


Figure 2.2: Blockchain Architecture

2.4 Consensus Algorithm

To establish trust within the network, blockchain leverages consensus algorithms like Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), Proof of Burn (PoB), Proof of Elapsed Time (PoET) and Proof of Authority(PoA)[43]. Every consensus algorithm has its own pros and cons. Therefore, this section discusses the different consensus algorithms.

- **Proof of Work (PoW)-** Proof of Work (PoW) was first proposed by Dwork and Naor [43] as a way for preventing spam emails. That is, the e-mailer must perform a certain amount of computing work to solve a complex mathematical puzzle before sending the mail, which increases significantly the cost of sending spam, thereby ensuring the security of the system.

  PoW is one of the first consensus algorithms used by blockchains such as Bitcoin, which allows security and agreement in a decentralized way. It requires heavy computation effort from the node of network (miners) which need them to solve a cryptographic problems (a complex mathematical problem), that is a puzzle, where we calculate a nonce value N such that the hash of the block meets a required threshold condition. Once miners find the hash value that meets the specific requirement, the block is broadcasted on the entire network and added to the blockchain after verifying. Then, the miners earn a reward. This way PoW makes sure of the security of the network, as changing a block will need performing all work again, faster than the rest of the network, which is an impractical task. Even though PoW assures security and is well proven, it takes a lot of power and computation at a single end.

- **Proof of Stake (PoS)-** It is a consensus algorithm focused on blockchain networks in a more environmentally friendly way than PoW [44]. Participants (validators)

in PoS are selected to propose and confirm new blocks depending on the amount of tokens they stake and lock up; more stake generally improves the chance of being selected [45, 46]. This greatly reduces energy consumption, adding to its higher scalability than PoW. Also, this means that block creation and transaction processing is faster on PoS. The rewards are sent to the chosen validator after block is validated. PoS is now used in major blockchains like Ethereum, Polkadot and Cardano [44, 45, 47]. While being more sustainable, PoS is argued for favouring wealthier participants.

- **Proof of Space (PoSpace)-** Proof of space (also called proof of capacity) requires participants to demonstrate they have reserved free disk space for block creation [48]. Instead of very intensive computations, miners pre-store large data sets called plots on their hard drives; when a new challenge appears, miners search their plots for the best match, and the best match earns a reward and proposes the next block [49, 50]. This is more energy-efficient because reading pre-stored data consumes far less energy than hashing continuously [48, 49]. This algorithm is more accessible as it needs common disk storage rather than expensive GPUs. However, it promotes hoarding of storage and large-scale data farms. PoSpace is used in blockchains like Chia (and earlier explored in proof-of-capacity systems such as Burstcoin) [49].

- **Practical Byzantine Fault Tolerance (PBFT)-** PBFT is used to handle malicious or faulty nodes in distributed systems, allowing the network to agree on a correct state even if some nodes fail or act incorrectly [51]. It runs multiple rounds (pre-prepare, prepare, commit) to ensure sufficient agreement on the same block or order; decisions are final (instant finality), which suits high-trust environments [51, 52]. One issue is overhead that grows with the number of nodes, making it less scalable than PoW and PoS. Examples include Tendermint BFT and Hyperledger Sawtooth PBFT; note that Hyperledger Fabric today recommends the Raft ordering service rather than PBFT [52–54].

- **Proof of Burn (PoB)-** It is an experimental algorithm where nodes burn their coins to an unrecoverable address to obtain validation rights; the tokens are removed from circulation [55]. This shows commitment to the network (analogous in spirit to staking, but with an irreversible cost). Since burning reduces supply, it can create scarcity; selected validators are rewarded [55]. PoB does not need continuous heavy computation, but it destroys token value, hindering user adoption, and can risk centralization if only a few can afford burns. One of the most well-known implementations of PoB is Counterparty's implementation of an XCP token via a burn that was provably done [56, 57].

- **Proof of Elapsed Time (PoET)-** In this type of algorithm, a random amount of time is assigned to validators as to how long they will have to wait before creating a block. That validator that has the shortest time is the one who gets to create the next block and thus, ensure fair and efficient energy usage [58, 59]. This approach to validation is challenged by proving that the timing is truly random and that it is truly being measured and reported (typically by using trusted hardware such as Intel SGX [60]). Although, this approach to validation results in lower energy consumption and faster finality when compared to PoW/PoS approaches, use of proprietary hardware limits the level of transparency and decentralization of this model. PoET is currently being utilized by Hyperledger Sawtooth [58].

- **Proof of Authority (PoA)-** Proof of Authority is a method of validation that is primarily used in permissioned blockchains. In these models, a select group of validators (based on their reputation/identity), validate transactions and produce new blocks. Unlike other methods of validation, the decision on who produces the next block is not based on computational power or the number of tokens held, but rather on the individual's reputation/identity [61]. As a result, each validator produces blocks in a predetermined order, resulting in high levels of performance, energy efficiency and low latency. This is ideal for scenarios where the need for high performance and rapid transaction times outweigh the need for decentralized validation [61, 62]. Although PoA results in high performance and energy efficiency, it also results in a lack of decentralization and makes it easier for validators to censor transactions (due to the small number of validators). Historically, PoA has been used in several blockchain-based platforms including VeChain [63] and xDai (now referred to as Gnosis Chain) [64] which has since transitioned to PoS model.

2.5 Blockchain Types

There are many ways that a blockchain network can be developed; however, each method is typically applicable to a specific application and its requirements. When creating a cryptocurrency (such as Bitcoin), a public network is used to enable any node to verify any transaction. A completely opposite requirement exists for a financial institution like a bank; a bank will require a private network to enforce strict access control to ensure only those with the proper authorization can access sensitive information [65]. The above described differences have led to the identification of two primary types of blockchain networks: public and private. Public blockchain networks provide access to all members of the network, enabling any member to validate any transaction. Therefore, the amount of power that a node has in validating transactions is distributed equally throughout the network, and there is no one entity responsible for the overall operation of the network. Private blockchain networks restrict access to a predetermined set of nodes, as determined by the network administrator. Only the authorized nodes have the capability to contribute to the blockchain network [66]. Additionally, the types of blockchain can be further divided by the nature of who can create new blocks in the blockchain network, resulting in four total types of blockchain [16, 67]. Table 2.1 provides a comprehensive breakdown of the different types of blockchain networks.

2.5.1 Private Permissionless

The private permissionless category represents blockchain networks where only a limited number of nodes are granted access to the network. All participating nodes in the network are able to both read from the network and write to it. Nodes within the network collaborate to create new blocks in the blockchain. All nodes are considered equal to one another. A notable example of a private permissionless blockchain is Holochain [68].

2.5.2 Private Permissioned

In the private permissioned category, only a predetermined set of authorized nodes are allowed access to the blockchain network. All participating nodes are able to read data from the blockchain network; however, the nodes are restricted to a predetermined subset

of nodes that are authorized to write to the blockchain network and begin the creation of new blocks. Examples of private permissioned networks include Hyperledger Fabric [69] and Hyperledger Besu [31].

2.5.3 Public Permissionless

This type of blockchain network grants an open invitation to any node desiring to join the network to participate in either reading or writing to the blockchain. Building trust between the participating nodes requires the utilization of expensive consensus protocols. The most prominent examples of public permissionless blockchain networks are well-known platforms including Ethereum, Bitcoin, and Witnet [32].

2.5.4 Public Permissioned

Similar to public permissionless blockchain networks, public permissioned blockchain networks grant access to all nodes wishing to participate in reading data and viewing the content of the network. However, the key difference between the two categories lies in that only a select group of nodes or participants are permitted to write to the blockchain network. Due to the lower overhead associated with the consensus protocols required for the validation of writes, the cost of establishing trust among the participants in the network is significantly reduced. Examples of well-known public permissioned blockchain networks include EOSIO [70] and Ripple [71].

Table 2.1: Comprehensive Differentiation of Blockchain Types

Aspect	Private Permissionless	Private Permissioned	Public Permissionless	Public Permissioned
Participants	Limited Authorized	Limited Authorized	Anyone	Anyone
Read Operations	Anyone	Anyone	Anyone	Anyone
Write Operations	Anyone	Limited Authorized	Anyone	Limited Authorized
Platforms	Holochain [68]	Hyperledger Fabric [69], Hyperledger Besu [31]	Ethereum, Witnet [32]	EOSIO [70], Ripple [71]

To summarize, the variety of methods available to develop a blockchain network allows for the development of numerous applications that are suited to specific use cases. Through categorizing blockchain networks into public or private categories and then subdividing them into permissioned and permissionless categories, the blockchain landscape provides a comprehensive structure for developing a wide variety of applications across many different industries.

2.6 Smart Contracts

Blockchain-based patient-centric architectures in the field of healthcare use smart contracts as an increasingly important and transformative instrument. Compared to standard contracts, smart contracts provide many advantages, such as greater automation, better clarity, better security, lower costs, and the removal of third-party intermediaries. Smart contracts are self-enforcing contractual arrangements in that their terms and conditions are pre-defined, embedded within a program of code, and will automatically execute

predetermined actions based upon specific conditions being satisfied [72]. Following deployment of a smart contract, the contract will remain on the blockchain and execute based upon the defined terms and conditions of the contract [73].

Smart contracts can also positively affect the healthcare field. Smart contracts facilitate the automation of specific actions (i.e., there is no need for human involvement) and provide for more efficient completion of healthcare-related transactions. Smart contracts support both healthcare providers and patients by improving the time that healthcare occurs and the burden of administrative tasks associated with healthcare [21]. Due to their immutable nature, smart contracts provide an environment of confidence for all parties involved due to the clarity in stating the terms and conditions of the contract, promote patient-centric care by ensuring transparent processes, and ensure the security of patient data and other sensitive healthcare information through the use of cryptographic algorithms used during the development of smart contracts [74]. Furthermore, smart contracts remove third-party intermediaries from the transactional equation, which reduces the transactional costs associated with healthcare-related transactions, enabling peer-to-peer interactions between patients and providers and potentially saving both patients and providers significant amounts of money.

The implementation and deployment of smart contracts within blockchain-based healthcare architectures require some level of technological competence. Generally, programming languages such as Solidity are used to write smart contracts on platforms like Ethereum. The development process has to be "right" so that the contract can execute well and remain stable. As soon as a contract is deployed on the blockchain, it becomes immutable. To avoid any mistakes or flaws that may have unforeseen consequences, code for a given contract must be meticulously vetted and tested. Seamless integration is necessary when introducing smart contracts into existing healthcare systems. To enable smooth communication between a smart contract and other components of the healthcare infrastructure, interoperability challenges must be addressed [75]. However, despite their numerous advantages, issues concerning legal status still exist among proponents of this concept. These remain to be non-binding universally enforceable ones [72]. Hence, for operating optimally in the existing regulations, it is crucial to integrate components of legal framework which will result in adoption of smart contracts in healthcare sector.

As advancements in technology occur, an emphasis on continued research and development into improving the functionality of smart contracts will provide the most effective means for smart contracts to revolutionize how healthcare is delivered. Smart contracts are essential to creating a strong environment that will maximize the use of smart contracts within healthcare. A collaborative effort from technical professionals, lawyers and all other stakeholders involved with the healthcare system would be necessary to develop this environment. Overall, while smart contracts have many technical and legal implications in developing blockchain based patient-centric healthcare systems; the potential for smart contracts to deliver a more efficient, secure, patient-centered healthcare model is positive and promising.

2.6.1 Smart Contract Development Stack

Smart contracts are blocks of code that are autonomous and self-executing, developed and deployed on various blockchains, and are the building blocks of blockchain-based systems. This section describes the major tools with which to design, implement, and deploy smart contracts in our proposed architectures.

2.6.2 Solidity

Solidity is a static-typed, high-level programming language designed to specifically write smart contracts on Ethereum-compatible blockchain networks. Solidity's syntax is similar to that of JavaScript and supports inheritance, complex user-defined data types, and event-driven programming. Solidity is needed for the specification of contractual logic governing automatically interactions between system parties [76].

2.6.3 Truffle Framework

The main purpose of Truffle is to make the smart contract development process easier through its toolset. It does this by making it simpler to compile, test and deploy smart contracts on the Ethereum blockchain [77]. Truffle allows developers to integrate their Ethereum-based smart contracts into one framework, which has allowed them to work at a consistent and rapid rate.

2.6.4 Hardhat

Hardhat is a full-fledged Ethereum development environment that includes a collection of tools (testing, compiling, deploying) to help developers create, test and deploy smart contracts on the Ethereum blockchain. These tools include a local network, task runner, and a vast library of plugins that can be utilized to automate repetitive tasks, and have extended debugging capabilities [78]. By utilizing Hardhat, developers are able to simplify the management of their Ethereum-based projects, as well as increase the overall reliability of smart contract development.

2.6.5 Ganache

Ganache is a simulation of the Ethereum blockchain that can be utilized during the development of a project [79]. The most notable feature of Ganache is that it creates a simulated blockchain environment that mimics a real-world blockchain network; however, it eliminates the complexity of achieving consensus and eliminates transaction fees. Ganache is a very effective method of prototyping, debugging and verifying the logic of a smart contract in a totally controlled and isolated environment.

2.6.6 MetaMask

MetaMask is a browser extension Ethereum wallet that is being used by many developers as a method to store securely a user's identity, tokens and perform transactions on deployed smart contracts. MetaMask provides users with a secure method to manage their keys, sign transactions and create a simple GUI for interacting with blockchain based applications. In our system, we utilize MetaMask as a method for users to enter the application and ensure that communications between the user and the blockchain are secure [80].
The integration of these tools enables end-to-end and unified development for smart contracts, hence enabling efficient and secure implementation of blockchain solutions.

2.7 Blockchain Deployment and Oracle Integration

The implementation of a blockchain solution in a secure and effective way not only requires the setup of a correct blockchain network but also secure integration of this network with extrinsic, off-chain resources. In this chapter, two key technologies used to achieve these goals will be introduced: Go-Ethereum, for implementing blockchain, and Chainlink Oracles, which provide secure off-chain data integration.

2.7.1 Go-Ethereum (geth)

Go-Ethereum, better known as geth, is the Go language version of an official Ethereum node featuring robust functionality for running Ethereum blockchain networks. Geth provides support for the establishment and maintenance of public and private Ethereum networks, node synchronization, transaction processing, and deployment of smart contracts. In this work, geth is utilized strictly for the deployment and maintenance of permissioned private Ethereum networks. Its support for fine-grained access control, consensus mechanism configuration, and extensive logging makes it priceless for secure, scalable blockchain infrastructure [81].

2.7.2 Chainlink Oracle

Smart contracts usually are not able to directly query data outside of their blockchain, thus oracles are needed to tap into that query. Chainlink is the most widely used oracle network that securely retrieves external data and feeds it to smart contracts. Chainlink fills the gap between use cases for blockchain and off-chain sources like web APIs, databases, and other external data repositories. In our designs, Chainlink oracles securely query patient metadata from off-chain storage in a fashion that preserves data integrity and blockchain immutability. The reputation, data accuracy, and security history of Chainlink also greatly enhances the trust in the blockchain solutions described in the following chapters [82].
Go-Ethereum and Chainlink working together provide an adequate basis to develop scalable blockchain-based solutions, which will allow users to access external data sources efficiently, which is a necessity when accessing healthcare data.

2.8 Data Storage Mechanisms

Secure and efficient data storage is an important component to developing blockchain-based healthcare applications, specifically because of the sensitivity of patient information. Below are descriptions of two key data storage technologies used during the development of this application, the first being MongoDB, the NoSQL database model (off-chain data storage model) utilized in the early stages of this project, and the second being IPFS (decentralized data storage utilizing a P2P network for added security and decentralization).

2.8.1 MongoDB

MongoDB is a well-established NoSQL Database Model that utilizes a flexible schema, scalable architecture, and fast performance to accommodate the large amounts of data

that can be generated from the collection of patient records [83]. Data stored in MongoDB is document-based, making it very useful for collecting and storing various types of patient record data, such as structured and semi-structured records. In the early stages of development of our blockchain-based healthcare application, MongoDB was used to store patient data off-chain so that we could avoid the expense and the decrease in performance that occurs when storing data directly on the blockchain. We took full advantage of MongoDB's strong indexing, querying optimization and scalability options to enable fast and efficient data retrieval and processing.

2.8.2 IPFS

IPFS is a decentralized file system that uses a P2P (peer-to-peer) network to store and distribute data in a decentralized way. Every file stored in IPFS has a unique identifier that is a function of a cryptographic hash of the file's content [84]. The use of IPFS in the enhanced DiabeticChain application allows us to greatly enhance the security, resilience to point failures, and the decentralization of data storage, and therefore makes IPFS a viable replacement for traditional data storage models like MongoDB.
In combination, MongoDB and IPFS demonstrate the need to evolve the practice of data storage from central/off-chain storage to completely decentralized storage methods, while at the same time enhancing the overall efficacy, security, and dependability of blockchain-based healthcare systems.

2.9 Performance Testing and Benchmarking

Performance testing and benchmarking of blockchain-based healthcare systems is necessary so that to verify their reliability, scalability and responsiveness under realistic conditions. This section talks about Hyperledger Caliper, a widely implemented blockchain performance benchmarking tool, used to evaluate our blockchain solutions.

2.9.1 Hyperledger Caliper

Hyperledger Caliper is an open-source blockchain-agnostic benchmarking framework used to measure blockchain platforms' performance. It provides standardized metrics like transaction throughput, latency, resource consumption and network efficiency. Hyperledger Caliper enables intensive and repeatable performance analysis by simulating different blockchain workloads [85]. In our thesis, Caliper is used for evaluating the throughput and latency of transactions and queries in blockchain under various scenarios, so that the solutions pass the required performance standards for real-world healthcare applications. Blockchain architecture scalability and response time validation are efficiently done with the assistance of blockchain performance metric generation capability and versatility.

2.10 Interoperability and Layer-2 Scaling

Interoperability and scalability are essential factors to consider when building operational blockchain-based healthcare platforms because they facilitate the seamless exchange of data between healthcare systems and the effective handling of a high volume of transaction

activity. This area of focus will concentrate on HL7/FHIR interoperability, FHIR APIs for integrating data, and Polygon as a Layer 2 scaling solution.

2.10.1 HL7/FHIR

The Health Level Seven International Fast Healthcare Interoperability Resources (HL7/FHIR) is an extensively used international standard for electronic health records (EHRs) and many other healthcare applications. It offers both a structured, readable method to communicate health data across various systems, and a documented interface to describe how the communicated data should be interpreted [10]. Therefore, by using HL7/FHIR, semantic interoperability may be established between healthcare entities to allow them to exchange data and accurately interpret the data received from each other.

2.10.2 FHIR APIs

Using FHIR APIs enables us to build secure and interoperable pathways of communication between blockchain-based healthcare applications and current healthcare environments. The FHIR API defines the specific interfaces and operations to support the integration of blockchain-based healthcare applications into current EHR systems, labs, and clinical applications [86]. As such, FHIR APIs are a critical component of our solutions to exchange patient information, medical data, and metadata in a structured format to support their seamless integration into large-scale healthcare ecosystems.

2.10.3 Polygon

The Polygon network was originally called Matic Network. It is a layer 2 Ethereum network designed to improve upon some of the scalability issues inherent in traditional blockchains. Offloading computational tasks from the Ethereum mainnet to layer 2 on the Polygon network enables a significant increase in the number of transactions per second, reduces latency and lowers transaction fees. Our use of a next-generation architecture enables the scalable and high-performant management of the increasing volume and complexity of healthcare data transactions, and will remove scalability barriers from the Ethereum mainnet, ensuring that all data requirements of today's healthcare solutions can be processed in real-time [87].

All three — HL7/FHIR standards, FHIR APIs and Polygon — provide a unified framework for developing blockchain-based, scalable, and interoperable healthcare solutions addressing the fundamental issues preventing the adoption of blockchain technology in the healthcare industry.

2.11 Privacy-Preserving Cryptography and Policy-Based Encryption

This section presents two cryptographic primitives employed throughout the thesis: zero-knowledge proofs (with emphasis on zk-SNARKs) and Attribute-Based Encryption (ABE). Both facilitate privacy-preserving consent and allow fine-grained, interoperable access to HL7/FHIR health records.

2.11.1 Zero-Knowledge Proofs

A zero-knowledge proof enables a prover to assure a verifier that a statement holds without disclosing any extra information regarding why it holds. A secure construction provides three properties in reality: honest provers can persuade honest verifiers for true statements; dishonest provers cannot get verifiers to accept false statements except with very low probability; and the verifier learns nothing except that the statement is true. For healthcare applications, these properties enable consent and eligibility checks to be carried out without revealing sensitive patient attributes, maintaining confidentiality but still allowing verifiable access decisions [88].

2.11.2 zk-SNARKs

A zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) is an efficient class of zero-knowledge proofs optimized for environments where proofs are required to be brief and quick to verify, such as on a blockchain. After a setup phase that produces proving and verification keys to a given computation, a prover may produce a concise proof that a secret witness fulfills the computation, and a verifier may verify it at little expense. In this thesis, assertions like adherence to active consent rules can be formulated as arithmetic circuits or constraint systems, enabling verification on-chain without disclosing underlying identifiers or attributes. This keeps private data off the ledger while presenting a clear, auditable decision result [89].

2.11.2.1 Groth16 zk-SNARK

Groth16 is a zk-SNARK construction that produces constant-sized proofs with efficient verification using bilinear pairings. It produces proving and verification keys after a trusted setup specific to the desired circuit; it compiles computations into Rank-1 Constraint Systems (R1CS), allowing succinct proofs without revealing any witness. Groth16 is efficient and tool-friendly enough to make on-chain verification feasible in resource-poor settings, which we utilize later in this thesis [90].

2.11.3 Attribute-Based Encryption for Interoperability

Attribute-Based Encryption offers decryption only if a user's attributes meet an expressed policy. Two variant forms occur in practice: in key-policy ABE, the policy is included in the decryption key and ciphertexts contain attributes; in ciphertext-policy ABE, the ciphertext contains the policy and keys contain attributes. In cross-organization health data sharing, the ciphertext-policy model accords with data-owner control: a record is encrypted according to a policy stating who should be granted access, and only keys that belong to compatible attributes can decrypt. This mechanism inherently implements role-based encryption by handling role, organization, use purpose, consent status, or other such claims as attributes. Together with HL7/FHIR normalization, policy semantics are the same across disjointed EHRs, supporting interoperable sharing while keeping enforcement to be cryptographic, auditable, and independent of any one administrative domain [91–93].

2.11.4 AES-GCM

AES in Galois/Counter Mode (GCM) is an authenticated encryption mode that incorporates counter-mode encryption with polynomial authentication based on a Galois field. It delivers confidentiality in addition to integrity and authenticity of ciphertext and optionally associated data (AEAD), generating an authentication tag to provide tampering detection. GCM is software- and hardware-efficient and parallelizable, which qualifies it for use in high-throughput systems; we use AES-GCM to encrypt sensitive payloads while maintaining implementation simplicity lower than a product of independent encrypt-then-MAC constructions [94].

2.12 Summary

This chapter introduced the essential tools and technologies used in the development and implementation of the blockchain-based healthcare architectures presented in this thesis. By familiarizing the reader with the core components of the smart contract development stack, deployment environments, data storage mechanisms, performance evaluation tools, and interoperability standards, this chapter lays the necessary groundwork for understanding the technical details and design decisions described in the subsequent chapters.

With these foundations established, Chapter 3 presents a systematic review of current blockchain-based patient-centric healthcare architectures, identifying key research gaps in the field.

Chapter 3

Literature Survey

This chapter offers a systematic survey of blockchain architectures for patient-centric healthcare. It describes the review process and structures findings in terms of security/privacy, consent management, interoperability, and compliance, with open gaps that inspire our designs.

3.1 Introduction

In order to develop patient-centric architectures, this literature review explores the applications, advantages and difficulties of implementing blockchain technology in healthcare. It focuses on how blockchain can give patients ownership of their data, eliminate single points of failure, improve cybersecurity, and promote trust among all parties involved in healthcare. With the knowledge and resources necessary to take control of their healthcare journey, patients may at last move from the margins to the center.

3.2 Related Work

This section summarizes the relevant existing literature by highlighting the key findings, contributions, and limitations of each study in a concise and organized manner along with detailed analysis.

3.2.1 Existing Literature Summary

Recent work sharpens particular building blocks for healthcare blockchains. Chenthara et al. suggest HealthChain, a Hyperledger Fabric–IPFS architecture that maintains EHR privacy/integrity without including an explicit HL7/FHIR layer and documents few performance metrics [95]. Li et al. also combine Fabric (on-chain) with IPFS (off-chain) for EMR sharing but offer only high-level discussion of scalability/security [96]. For high-granularity authorization, Tuler De Oliveira et al. propose SmartAccess, an attribute-based access-control scheme appropriate for cross-organization sharing but with increased decentralized overheads [97]. Kenazza et al. outline an Ethereum + IPFS architecture along with recovery measures while mentioning real-world scalability issues [98]. IoT/IoMT security forums cover Chen et al.'s biometric-backed integrated mutual-authentication and key-agreement scheme (validated using the real-or-random (ROR) model with minimized communication/compute expenses [99], and Liu et al.'s permissioned drug-management/authentication framework (PBFT, lifecycle traceability, end-to-end key agreement) exhibiting high attack resilience with minimal resource utilization

Table 3.1: Summary of Existing Literature

Author	Year	Findings/Contributions	Limitations
Azaria et al. [34]	2016	• MedRec: Decentralized EMR management using blockchain. • Ensures data sharing, accountability, authentication, and confidentiality for sensitive medical information. • Working prototype presented with detailed analysis and discussion.	• Scalability challenges persist. • Lack of detailed security analysis. • System relies on provider nodes adhering to external regulations like HIPAA for data copying.
Chen et al. [36]	2018	• Blockchain-based storage for safe sharing of medical data. • Designed block structure, transaction types, and main functions. • Managed personal medical data via blockchain and cloud storage.	• Not yet integrated with existing healthcare systems. • Regulatory and legal challenges not addressed.
Reen et al. [102]	2019	• An Electronic Health Records (EHR) privacy and security solution on permissioned Ethereum blockchain. • Combined symmetric and asymmetric key cryptography for access control and secure data storage.	• The use of biometric authentication for identity verification may not be feasible for all patients. • On-chain existence of a single entity of the hospital.
Dubovitskaya et al. [12]	2020	• ACTION-EHR, a secure patient-centric EHR data sharing framework on a permissioned blockchain, enabling patients to manage records. • Ensures privacy and data security via encryption and digital signatures. • Tested with de-identified patient data using Hyperledger Fabric.	• Single point of failure if the system deployment lacks redundancy (e.g., with a single orderer and single CA).
Kordestani et al. [103]	2020	• HapiChain, a blockchain-based architecture for patient-centered telemedicine that enhances security and reliability. • The system integrates telemonitoring and teleconsultation services with three layers: interface, DApp, and blockchain.	• Does not provide any detailed analysis of performance. • Need to address the attacks and verify the security of HapiChain on those attacks.
Kanagi et al. [27]	2020	• A patient-centric blockchain framework for efficient clinical data sharing. • Demonstrates rapid sharing of health data, addressed interoperability issues and reduced response times in Taiwan's healthcare.	• Costly maintenance and deployment. • Does not promise its generalizability to any healthcare data.
Mani et al. [104]	2021	• Patient-centric healthcare data management (PCHDM) framework based on blockchain, ensuring security and privacy of health records. • It includes a secure Health record chain network architecture, a permissioned network in Hyperledger fabric, and performance testing.	• Requires significant computational resources and energy consumption, leading to scalability issues. • May not provide absolute protection against all security and privacy threats.
Chelladurai et al. [105]	2021	• A patient-centric EHR storage and integrity management system based on blockchain to ensure secure transactions in e-Health systems. • Emphasis on responsible handling of private health data, ensuring integrity and confidentiality.	• Encountered challenges such as regulatory compliance, interoperability, and scalability.

[100]. Identity-focused work like Health-zkIDM (Bai et al.) uses zero-knowledge proofs on Hyperledger Fabric and delivers Caliper metrics, but not HL7/FHIR standardization or decentralized clinical record storage [101].

Azaria et al. [34] introduced MedRec, a blockchain-based EMR management framework using Ethereum. It supported auditability and authentication but relied on private off-chain databases, limiting decentralization.

In separate studies [106] and [33], authors aimed to maintain privacy, data integrity, and patient anonymity by devising a blockchain-based system for medical data access and sharing, facilitated through an immutable distributed ledger. Nonetheless, it is worth noting that scalability concerns were identified within their proposed system. Lodha et al. [40] designed a healthcare system for secure medical data sharing among various stakeholders on blockchain storage system without any trusted third-party. To prevent

repeated extensive check-ups, the proposed system does, however, require the patient to upload their entire medical history. Requirement of necessary hardware and a stable internet connection also serves as a roadblock.

The authors in [107] developed a framework for the earlier diagnosis of diabetes utilizing machine learning algorithms and secure blockchain-based patient EHR management to protect privacy. The data is collected from wearable devices, stored on the IPFS (Interplanetary File System), sent to the EHRs manager and given as an input to ML classification models. After processing, the predictive results are generated and stored on the blockchain after patient's consent. In [36], a system is designed to share and manage personal health data using blockchain and cloud storage, eliminating the dependence on a centralized third-party for sensitive data sharing. However, the proposed system is yet to be integrated with the existing healthcare systems. In [102], a decentralized blockchain-based system for managing patient-centered electronic health records with IPFS storage and a control mechanism for limited access was proposed by Reen et al. The system faces the limitations like a maximum replication factor of 2, privacy infringement and on-chain existence of single entity of hospital.

Dubovitskaya et al. [12] developed a proposed framework called ACTION-EHR that uses a permissioned blockchain to give radiation oncology patients secure access to their medical record sharing. This approach allows medical practitioners and researchers to access patient data more quickly and efficiently while reducing costs. But there is risk of having single point of failure of the system and in case of emergency situations, there is a requirement of "break glass" mechanism which serves as a limitation of this paper. In [103], the authors proposed HapiChain - a blockchain-based patient-centric framework for teleconsultation service in complementary to telemonitoring for effective and efficient communication. The system needs to be tested against security attacks to ensure security. In [27], the authors proposed a framework making clinical data available to patients and enabling clear, traceable, secure, and efficient data exchange but the proposed architecture has a costly maintenance and deployment and does not guarantee that it will be applicable to every healthcare data.

Mani et al. [104] proposed an approach to store health records on an IPFS-based distributed ledger system using Hyperledger Fabric technology and enabled decentralized storing and sharing of health information while guaranteeing scalability, privacy, security, and confidentiality. A large number of resources are required for implementation of multi-blockchain systems and extend framework integrated with non-fungible tokens. Chelladurai et al. [105] proposed an approach for EHR management when health records are fragmented among healthcare entities.

Jabarulla et al. [108] devised a decentralized approach for managing patient-centric images using blockchain technology. Their system aimed to enhance data security and privacy while eliminating the need for a centralized authority. However, this system presented a potential drawback in terms of private key management due to the manual encryption and decryption of medical images. In Fatokun et al [28], an EHR system that is based on the Ethereum blockchain and smart contracts was introduced. The authors' approach aimed at preventing any reliance on third-party systems as a way of ensuring patient-centricity, interoperability, and data security. Furthermore, they noticed that the said system had some issues connected with scalability of blockchain and communication overheads related to it such as bandwidth requirements for block mining and overall network communication. An innovative framework called PCH was introduced by the authors in [13]. Within this architecture, blockchain, cloud computing and IoT

are combined together to enhance semantic interoperability among healthcare systems. It should be realized though that implementation and maintenance costs would be enormous. This approach also does not entirely overcome legal complexities associated with sharing patients' data across different healthcare platforms.

SynCare is a novel blockchain and cryptographic remote patient monitoring platform introduced by Pighini et al. in their work [109]. The innovation helps to make possible storage of patient focused data, sharing and retrieval of it, while strictly protecting privacy for sensitive details. Nevertheless, one disadvantage is that the process encrypts and decrypts information when shifting between functions thereby leading to long loading time. Taylor et al. [110] proposed VigilRx—a drug dispensing system which is designed with a focus on patients and interoperability. This therefore gives patients more control over their data as well as reduces bottlenecks of information through better transfer efficiency of data. It also seeks to promote compatibility, curb fraud, and improve precision and availability of data at all times. It mainly relies on blockchain technology so that it can achieve scalability but not security in network terms. Due to this heavy dependence on it, there are challenges such as blockchain congestion or susceptibility to security breaches.

George et al. [14] presented the MediTrans system for safe interoperable data access management to address the concerns of patient-centricity, transfer of large-scale data sets, and lack of trust. The system is dependent on willingness of stakeholders to join the network. Mudaliar et al. [111] presented a blockchain-based and machine learning-based electronic health record management system to track and predict the risk of arrhythmia and heart illnesses, but, respectfully, the technology carries a potential data theft risk.

In [112], the authors proposed a blockchain-based health record access framework using Hyperledger Fabric to enhance patient control, protect privacy and data integrity. Patients can impose necessary conditions before granting access to their medical data. But they proposed a conceptual framework that has not been implemented in a real-world setting and integrated with the existing healthcare models. Rai in [113], proposed a patient-controlled blockchain-based electronic health records management system to ensure patient's ownership of their medical data, elimination of security risks, data integrity and confidentiality. But the smart contracts of the system are based on time and notifications.

To conclude, the analysis of blockchain-based healthcare architectures demonstrates advancements and challenges in patient-centricity, data security, privacy, and interoperability. It is true that many existing healthcare systems enable patient features, but with data scalability and security trade-offs. These studies show blockchain's potential in healthcare improvement but also the need for robust solutions for the same.

3.2.2 Analysis of Existing Literature

An examination of various blockchain healthcare architectures showed both similarities and differences between methods and trade-offs. Data protection and privacy to the patient were two main issues across all studies. Most preferred blockchains included Ethereum, Polygon, and Hyperledger Fabric for data protection and privacy. Many studies found that patients have full control of their data and can track it. However, many studies also found limitations in how well they addressed the above two items. The interoperability [114] and scalability [115] considerations varied greatly among the studies as both used off-chain and on-chain data storage solutions. Lastly, most of the consensus mechanism studies used either Proof-of-Work or Proof-of-Authority. A majority of

studies lack detailed security analysis and performance evaluation. Thus, while these architectures demonstrate potential to improve healthcare systems, the existing literature suggests the need for more comprehensive solutions that balance security, privacy, user control over data, interoperability, and scalability while addressing specific healthcare domain requirements. Table 3.2 presents an analysis of the existing literature. In general, the literature optimizes one pillar at a time—(i) HL7/FHIR-based semantic exchange, (ii) IPFS-backed integrity with on-chain anchors, (iii) patient-centric IoT/cloud/blockchain workflows, (iv) attribute-based, policy-driven authorization, or (v) IoT/IoMT authentication/provenance—instead of providing an end-to-end, privacy-preserving pipeline [95–97, 99, 100]. Gaps keep reappearing around consent verification without disclosure (e.g., zk proofs), standards-aligned data normalization, and systematic multi-round benchmarking; various works call for stronger evaluation explicitly or report sparse metrics [95, 96, 116]. Identity-centric designs like Health-zkIDM promote private authentication without incorporating HL7/FHIR standardization, encrypted record storage on IPFS, or high-granularity ABE policies, making clinical data flows incomplete [101].

Table 3.2: Analysis of Existing Literature

Study	Existing Challenges						Use case	Type of Blockchain	Technology	Data storage	Consensus Algorithm	Security Analysis	Performance Evaluation
	a	b	c	d	e	f							
Azaria et al. [34]	✓	✓	✓	✓	✓	X	Electronic Medical Records	Permissioned	Ethereum	Off-chain	Proof of Work	X	X
Chen et al. [36]	✓	✓	✓	✓	X	X	Healthcare Data Sharing	Permissioned	-	-	-	X	X
Dubovitskaya et al. [12]	✓	✓	✓	✓	✓	✓	Radiation Oncology Data	Permissioned	Hyperledger Fabric	Mixed	-	✓	X
Kordestani et al. [103]	✓	✓	✓	✓	X	✓	Telemedicine and Telehealth	-	Ethereum	Off-Chain (IPFS)	Proof of Work	X	X
Kanagi et al. [27]	✓	✓	✓	✓	✓	X	Clinical Data Sharing	Permissioned	Ethereum	Mixed	Proof of Work	✓	X
Chelladurai et al. [105]	✓	✓	✓	X	✓	X	Electronic Health Record Management	-	-	On-Chain	Proof of Work	✓	✓
Jabarulla et al. [108]	✓	✓	✓	✓	X	X	Medical Image Management	Permissioned	Ethereum	On-Chain	Proof of Work	✓	✓
Mani et al. [104]	✓	✓	✓	✓	✓	✓	Healthcare Data Sharing	Permissioned	Hyperledger Fabric	On-chain	Byzantine Fault Tolerance	✓	✓
Fatokun et al. [28]	✓	✓	✓	X	✓	X	Electronic Health Record Management	Consortium	Hyperledger Fabric	On-Chain	-	✓	✓
Gohar et al. [13]	✓	✓	✓	✓	✓	X	Healthcare Data Sharing	Permissioned	Hyperledger Fabric	Mixed	-	✓	X
Pighini et al. [109]	✓	✓	✓	X	X	X	Remote Patient Monitoring	Permissionless	Ethereum	Off-Chain (Cloud)	Proof of Work	✓	X
Taylor et al. [110]	X	✓	✓	✓	✓	✓	Prescription Management System	Permissionless	Ethereum	On-chain	Proof of Work	✓	✓
George et al. [14]	✓	✓	✓	✓	✓	X	Data Access Management	Permissioned	Ethereum	Off-Chain (Cloud)	Proof of Authority	X	✓
Abutaleb et al. [112]	✓	✓	✓	X	✓	X	Healthcare Data Sharing	Permissioned	Hyperledger Fabric	-	-	✓	✓
Rai [113]	✓	✓	✓	✓	✓	X	Electronic Health Records	Permissionless	Ethereum	On-Chain	Proof of Work	X	✓

a: Data Security, b: Patient Privacy, c: User Control over Data, d: Traceability, e: Interoperability, f: Scalability

3.2.3 Objectives and Contributions

To enhance the security and privacy of healthcare data in blockchain-based architectures, this research paper analyses patient-centric blockchain-based healthcare systems. The following are the unique objectives and contributions of this paper:

- Provides a systematic literature review to identify, classify and analyze the patient-centric blockchain-based healthcare architectures to discover the main research gaps, trends, and challenges and proposes areas for future research with potential solutions.
- Compiles a selection of 41 representative papers from a collection of 217 papers to ensure a comprehensive analysis and identifies the trend over the publication years,

publication types, their geographical distribution, publication channels, source of publication, main challenges addressed and their contributions, catered use cases, most prevalent blockchain platforms, types of blockchains and smart contract programming languages used in the implementation of existing architectures.

- Critically discusses the four research questions centered on enhancing security and privacy, streamlining patient consent management, addressing interoperability, and aligning architectures with healthcare data regulations while proposing future research directions for patient-centric blockchain-based healthcare architectures.

The contributions of this thesis can have a significant impact on shaping the landscape of patient-centric blockchain-based healthcare architectures. This thesis is intended to serve as a resource for both practitioner and researcher by completing an exhaustive review of the relevant literature; therefore, it provides a basis for informed decision making. The future research direction proposed here will establish a pathway toward advanced security mechanisms, dynamic mechanisms of consent, and legal frameworks that could potentially create a "ripple" of innovation throughout the creation of healthcare systems which are secure, privacy conscious, and compliant with regulations.

3.3 Research Methodology

A systematic review was chosen as the research methodology for this investigation since it will provide a comprehensive review of the current literature regarding patient-centric systems utilizing blockchain technology and present a quantitative assessment of existing research-based evidence in the area of interest [117].

3.3.1 Article Selection

Selection of the right research papers based on the topic is the first step in our systematic literature review. This particular section gives an explanation of the selection procedure of the articles; covering the search strategy, inclusion and exclusion criteria, their screening, and the outcomes. The entire process of selecting relevant articles is depicted in Figure 3.1.

3.3.1.1 Search Strategy

To conduct a systematic literature search, a well-defined search protocol was created. To ensure the retrieval of all publications pertaining to the convergence of blockchain technology and patient-centric healthcare, the search phrase was carefully developed. Pilot searches were performed to identify suitable keywords and terms. Ultimately, the search query was designed to capture papers with abstracts mentioning "patient centric", "patient-centric", "patient-controlled", "blockchain", "smart contracts", "distributed ledger technologies (DLT)" or "DLT". The search period was limited from January 2018 to December 2023 to focus on recent and relevant publications. Table 3.3 illustrates the search query.

Clear inclusion and exclusion criteria were established to ensure the selection of appropriate papers. To be included, articles needed to be original, peer-reviewed publications discussing blockchain-based patient-centric systems. The publications had to describe

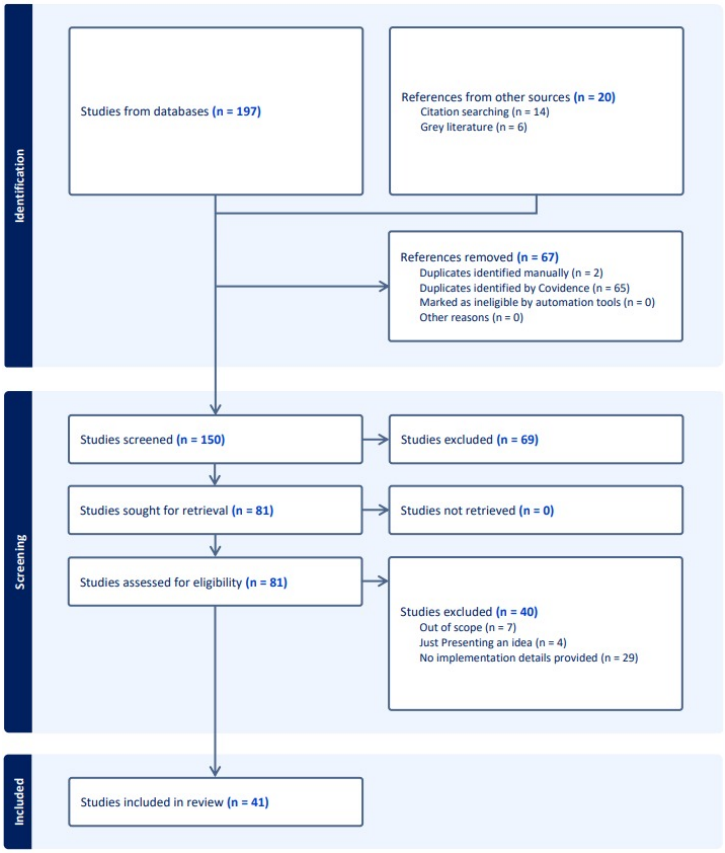


Figure 3.1: PRISMA Search Methodology

conceptual models, frameworks, architectures, prototypes, proof-of-concepts, implementations, or pilots relevant to patient-centric applications. Moreover, included articles should provide sufficient explanation of their research findings. Exclusion criteria encompassed non-relevant publications, non-English articles and review articles. Table 3.4 and 3.5 depict the inclusion and exclusion criteria respectively.

Table 3.3: Search Query to Find Relevant Papers

SEARCH QUERY:
[[Abstract: ”patient centric”] OR [Abstract: ”patient-centric”] OR
[Abstract: ”patient-controlled”] OR [Abstract: ”patient controlled”]] AND
[[Abstract: blockchain] OR [Abstract: ”smart contracts”] OR
[Abstract: ”distributed ledger technologies”] OR [Abstract: dlt]]
AND [E-Publication Date: (01/01/2018 TO 31/12/2023)]

3.3.1.2 Identification of Relevant Studies

The search yielded a total of 197 studies from various scientific databases. The primary papers were searched using ScienceDirect, Elsevier, Plos One, Google Scholar, MDPI, Springer, SpringerLink, JMIR (Journal of Medical Internet Research) Publications, Scopus and IEEE Xplore. In addition to the database searches, 20 potentially relevant papers through citation searching and references from other sources, as well as exploring grey

Table 3.4: Inclusion Criteria

INCLUSION CRITERIA:
1. Original peer-reviewed publication
2. Publication on the topic of blockchain-based patient-centric systems
3. Publication must describe a blockchain-based conceptual model, framework, architecture, prototype, proof-of-concept, implementations, or pilot relevant to patient-centric applications.
4. Publications including sufficient explanation of the research findings

Table 3.5: Exclusion Criteria

EXCLUSION CRITERIA:
1. Publications not in English
2. Review articles

literature, were identified resulting in a combined pool of 217 potential studies. The generated articles were transferred to Covidence, an online application tool utilized for systematic review management. After removing 67 duplicate references, the remaining 150 studies proceeded to the screening phase.

3.3.1.3 Screening Process

During the screening phase, the articles were carefully assessed based on their titles, abstracts, and full-text availability. The screening phase involved assessing titles and abstracts to exclude studies that were clearly unrelated to blockchain-based patient-centric healthcare architectures. Studies that mentioned "blockchain" in contexts not aligned with technical aspects of computer science were also excluded. For those articles that lacked sufficient detail from their titles and/or abstracts, these were then sent to the next screening phase for an in-depth review. Articles lacking complete text, articles written in a language other than English, duplicate articles, and articles covering unrelated subjects were also eliminated at this point.

Following the application of inclusion/exclusion criteria, 69 studies were eliminated as they failed to meet the research topic or were outside of the scope of the research. 81 studies remained which were eligible for retrieval and all 81 of them were successfully obtained for additional analysis. All 81 studies were reviewed for their suitability as part of a rigorous eligibility assessment using the inclusion/exclusion criteria previously established. A total of 40 studies were removed due to lack of relevance (out of scope), lack of specific details regarding how the ideas presented in the studies could be implemented, and/or because the objectives of the studies were inconsistent with the objectives of the research.

The remaining 41 articles provide a basis for the in-depth examination to identify the best practices within the industry and illustrate the potential of blockchain technology to alter patient-centric healthcare architecture.

3.3.2 Data Extraction and Formulated Research Questions

Extracting data for systematic literature review on blockchain-based patient-centric healthcare architectures; involves gathering key information from selected papers to effectively answer research questions. To accomplish this, a very detailed data extraction form was created to capture all necessary data elements to allow for an overall comprehensive review of the papers selected. As shown in Table 3.6 the data extraction form contains several data elements. Formulated research questions on blockchain-based patient-centric healthcare systems:

Table 3.6: Data Extraction Items

#	Data Item	Description
1	Study Identifier	Unique identifier for each research paper (e.g., #1, #2, etc.)
2	Title	Title of the article
3	Authors	Name of the authors
4	Country	Country of origin of the author(s)
5	Year	Publication year of the research paper
6	Publication channel	Name of publication place
7	Publication type	Type of Publication (Conference/Journal)
8	Publication source	Source of Publication (Academia/Industry)
9	Use Case	The specific use case addressed in the paper
10	Main Challenge Addressed	The primary challenge or problem discussed in the paper
11	Contribution	The contribution of the paper to the research domain
12	Limitations/ Disadvantages	Limitations or disadvantages mentioned in the paper
13	Contribution Type	Type of contribution (e.g. architecture, framework, system)
14	Blockchain Type	Type of blockchain (e.g., permissioned, permissionless, consortium)
15	Platform/Framework	The blockchain platform or framework utilized in the study
16	Smart Contracts Language	The programming language used for smart contracts in the study

- RQ1: What methods can be used to enhance the security and privacy of healthcare data using blockchain storage and sharing mechanisms?

  Current blockchain-based healthcare data sharing methods may not adequately protect patients' personal information regarding security and privacy. Due to the sensitive and private nature of healthcare data, it is essential to protect these types of data from unauthorized access and data breaches. Therefore, researchers should concentrate on developing improved encryption techniques and more stringent access control mechanisms to further enhance the security and privacy of patient data within the blockchain-based healthcare system.

- RQ2: What advanced methods need to be developed to accurately identify and manage patient consent for data sharing in blockchain-based healthcare architectures?

  Due to the decentralized nature of blockchain-based healthcare architectures, there is a significant need for more sophisticated methods to develop and implement accurate and reliable methods to identify and manage patient consent for data sharing. Most current consent methods used for managing patient consent do not take into consideration the complexity and fluidity of patient data in a decentralized environment. Therefore, researchers need to develop methods to reliably identify and

manage patient consent for data sharing, which will ultimately give patients more control over their data and increase confidence in the healthcare system.

- RQ3: What strategies should be implemented to achieve interoperability among varying stakeholders within the healthcare system to ensure seamless communication and data exchange?

  The current lack of patient-centric interoperability within the healthcare system hinders the ability of various stakeholders such as healthcare providers, insurance companies, and patients to communicate seamlessly and share patient data. Implementing successful strategies to achieve interoperability will enable the integration of multiple entities in the healthcare system, and support the delivery of cooperative patient-centric care. If the issue of interoperability is not addressed, then patient data may continue to exist in silos and hinder the establishment of a true patient-centric healthcare ecosystem.

- RQ4: What design options are available to use to create blockchain-based patient-centric healthcare architectures that support compliance with healthcare regulations and provide patients with clear ownership of their healthcare data?

  There currently does not exist a blockchain-based patient-centric healthcare architecture that supports compliance with healthcare regulations and provides patients with a clear ownership of their healthcare data. Many solutions available today either do not have the appropriate level of compliance or data control mechanisms in place, which could limit the widespread acceptance of these solutions in the healthcare industry. The answers to this research question would help create a robust and compliant solution for use in the healthcare industry.

To answer these questions, the recent research articles are analyzed in Results section 3.4 and thereafter answered in Discussion section 3.5.

3.4 Results

This section discusses the results of analyzing patient-centric blockchain-based healthcare architectures. It covers the analysis of publication trends over recent years, geographic distribution, different publication channels, source distribution, contribution types, use cases of the research conducted, main challenges addressed, blockchain platforms used, and smart contract implementations.

3.4.1 Publication Years

The analysis of publication years reveals that the research papers selected for the systematic literature review predominantly belong to recent years, indicating the emerging nature of blockchain in patient-centric healthcare architectures as a research area as demonstrated in Figure 3.2. The earliest publication identified was in 2018. However, the majority of the selected papers were published in subsequent years. Specifically, in the year 2019, four papers (9.75%) were published, while in 2020, six papers (14.63%) were identified. The number of publications increased further in 2021, with nine papers (21.95%) published. In 2022, ten studies (24.39%) were completed. Finally, in 2023, eleven studies (26.83%) were

published. Therefore, although the total number of studies has been increasing steadily since 2019, it is apparent that the research area of patient-centric blockchain-based healthcare architectures is receiving increasing attention and interest by researchers.

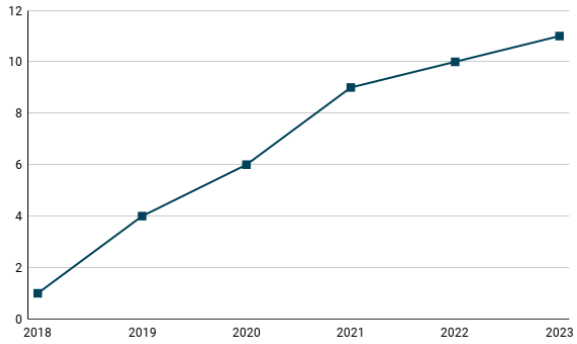


Figure 3.2: Timeline of Extracted Papers by Publication Year

3.4.2 Publication Type

According to the type of publication, 24.39% of the studies analyzed were conference papers and 75.61% were journal articles as shown in Figure 3.3. As a result, it appears that most researchers prefer to disseminate their findings through journals rather than other avenues. The large percent of journal articles also indicates that there is a strong need for rigorous and peer-reviewed research in this developing field.

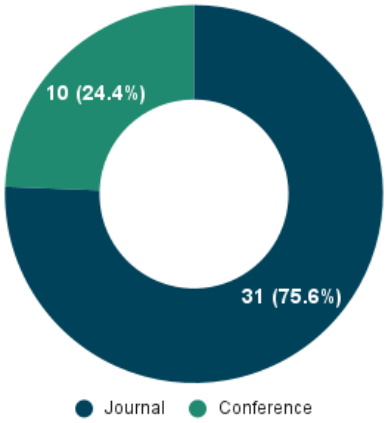


Figure 3.3: Publication Type of the Selected Research Papers

3.4.3 Geographical Distribution

The fact that authors of the studies included in this review are from many different countries provides evidence of a global interest in this subject. The largest percent of studies (39.02%) were authored by researchers from India, which demonstrates the significant contributions being made by academics and industry professionals from the Indian subcontinent. Researchers from the USA and China also made significant contributions with 9.75% and 12.19%, respectively, of the studies analyzed. The remaining 39.04% of the

studies that were selected were authored by researchers from Saudi Arabia, Ireland, Italy, Taiwan, Sweden, Australia, Bangladesh, Egypt, France, Colombia, Switzerland, Malaysia, and the United Arab Emirates. A graphical representation of the diversity of geographic locations represented by the studies is provided in Figure 3.4.

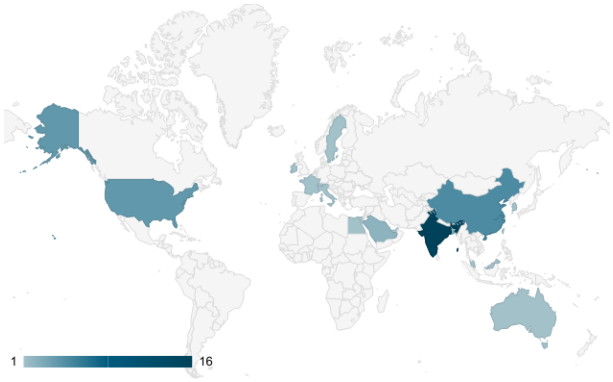


Figure 3.4: Geographical Distribution of the Selected Research Papers

3.4.4 Publication Channel

A number of sources were used in the papers selected for this systematic literature review and were represented in all types of well-established and recognized sources. IEEE had the largest representation among those selected, representing approximately 41.45% of the sources selected for this literature review. Second, the next source selected was MDPI which included 21.95% of the sources selected for this literature review. ELSEVIER and SPRINGER were third, and both of these sources had the same percentage representation at 7.30% of the sources selected for this literature review. All other sources that were identified included, but were limited to, WILEY, JMIR, GEORG THIEME VERLAG KG, OXFORD UNIV PRESS, ICMJE, NATURE PORTFOLIO, IEEE Access, Frontiers, and ACM, and together they comprised the remaining 22% of the sources selected for this literature review. A representation of the various sources is shown in Figure 3.5.

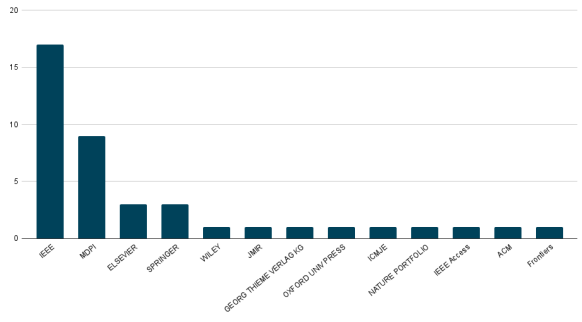


Figure 3.5: Publication Channel of the Selected Research Papers

3.4.5 Publication Source

A representation of the publication sources used for the selected research papers is illustrated in Figure 3.6, where it can be seen that out of the total 41 papers selected for this

literature review, there was a very large majority (i.e., 40 papers), which were published by academia. This indicates that academics have a strong interest in and are actively involved in this area of research. In comparison, only 1 paper was identified as having originated from the industry. Therefore, as shown in the figure, this large disparity between the number of papers from academic institutions and the number of papers from industry organizations emphasizes the predominant contributions made by academia to better understand and develop patient-centric blockchain solutions in the healthcare sector. Nevertheless, it should be acknowledged that industry-led perspectives and practices could have a significant influence on the actual implementation of blockchain technology within the healthcare sector.

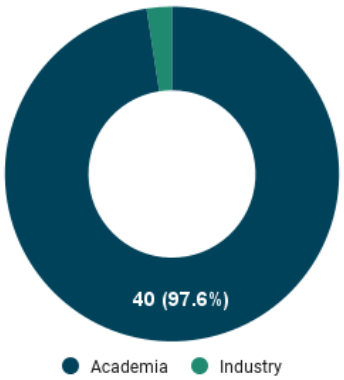


Figure 3.6: Publication Source of the Selected Research Papers

3.4.6 Contributions

As shown in Figure 3.7, the systematic literature review found three primary areas of contribution, namely, patient-centric data management and control, security and privacy enhancements, and interoperability and data sharing efficiency. As such, the first area of contribution (patient-centric data management and control) was evident in 41.5% of the selected research papers, the second area of contribution (security and privacy enhancements) was evident in 34.1% of the selected research papers, and the third area of contribution (interoperability and efficient data sharing) was evident in 24.4% of the selected research papers. These results indicate that researchers are actively working toward improving the way patient data is managed and secured, while increasing interoperability and the sharing of data in the context of blockchain-based healthcare solutions.

3.4.7 Contribution Type

With respect to the different types of contributions, the selected research papers primarily proposed frameworks (approximately 39% of the selected research papers) as shown in Figure 3.8. Systems designs were the next most commonly identified type of contribution (26.8%). Models and architectures were the least commonly identified type of contribution, although still significant, representing 14.6% and 19.5% of the selected research papers, respectively. Overall, the variety of types of contributions presented in the selected research papers illustrate the diverse and interdisciplinary nature of research efforts in patient-centric blockchain-based healthcare architectures.

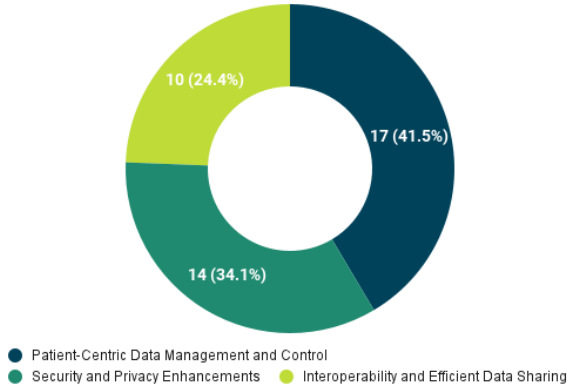


Figure 3.7: Contribution of the Selected Research Papers

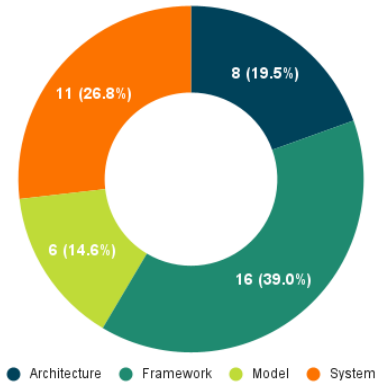


Figure 3.8: Contribution Type of the Selected Research Papers

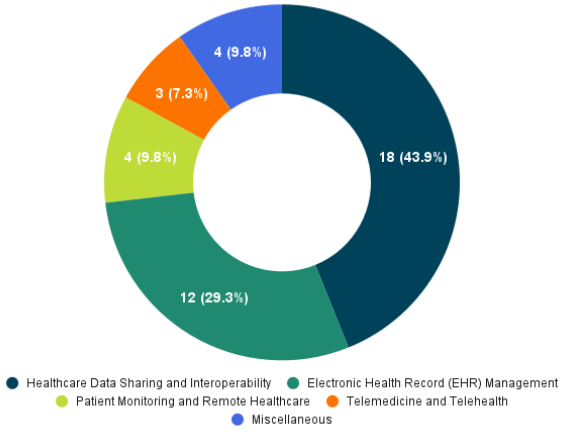


Figure 3.9: Use Cases Identified in the Selected Research Papers

3.4.8 Use Cases

Various use cases where blockchain was applied in healthcare are seen in Figure 3.9 from the selected papers in the systematic literature review. The most common use case, seen in 43.9% of the papers, was healthcare data sharing and interoperability. The second most common use case was Electronic Health Record (EHR) management, seen in 29.3% of the papers. Patient monitoring and remote healthcare, as well as some other odd and end

use cases, appeared in 9.8% of the papers. Finally, the use case with the least presence in the papers was for telemedicine and telehealth, seen in 7.3% of the papers.

3.4.9 Main Challenges Addressed

The selected research papers addressed copious challenges, and the results of this are seen in Figure 3.10. The most addressed challenge was that of data interoperability and management, present in 43.9% of the papers. Data privacy and protection were also major concerns, seen in 34.1% of the papers. Security and trust are the third most addressed challenge, seen in 22% of the papers. This shows that researchers are aware of the urgent need to overcome interoperability, security, trust, and also protect patient data privacy.

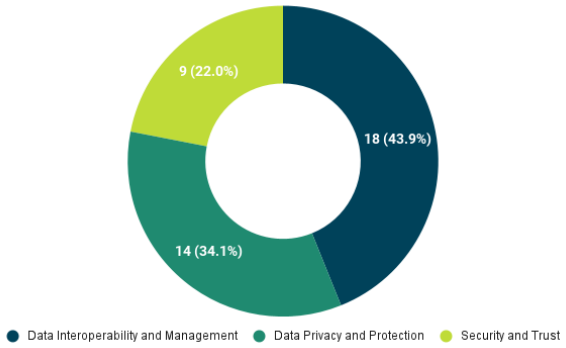


Figure 3.10: Main Challenges Addressed in the Selected Research Papers

3.4.10 Blockchain Platforms

Ethereum/Ganache/Remix and Hyperledger Fabric were the most widely used options, as seen during analysis of the blockchain platforms used in the selected research papers, shown in figure 3.11. In comparison, 48.8% of the papers used Ethereum/Ganache/Remix, whereas 31.7% of the papers used Hyperledger Fabric. Other blockchain platforms such as Hyperledger Indy appeared in a smaller number of papers (7.3% of the papers). This reveals a preference for implementing patient-centric blockchain-based healthcare architectures on mature and well-known blockchain platforms. The remaining 7.3% of the selected papers utilized other blockchain platforms for their implementation.

3.4.11 Blockchain Type

The majority of the selected papers (68.3%) implemented architectures using permissioned/private blockchains. This particular blockchain type offers enhanced privacy and control, making it appropriate for use in healthcare applications. The less desirable choice, permissionless/public blockchains, was mentioned in only 7.3% of the articles. Consortium blockchains, offering a balance between centralization and decentralization, were employed in 7.3% of the papers. The rest papers (17.1%) did not specify the type of blockchain used. The type of blockchain used for implementation is demonstrated in Figure 3.12.

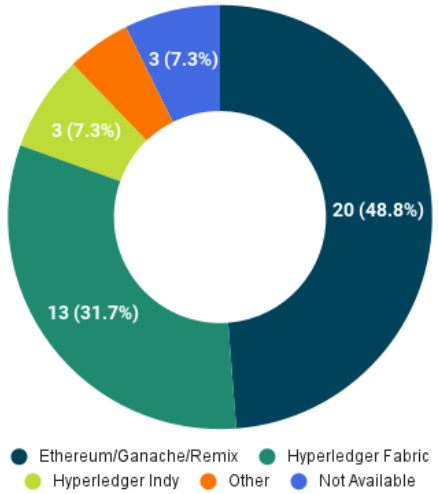


Figure 3.11: Blockchain Platforms Used in the Selected Research Papers

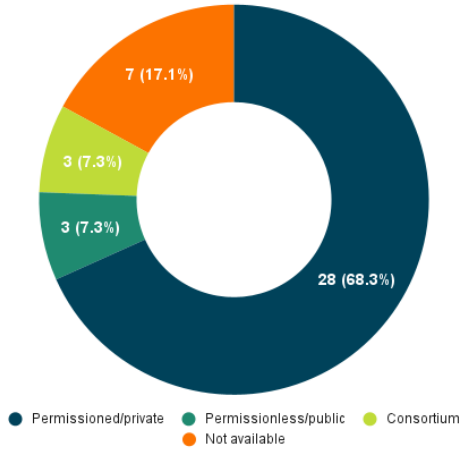


Figure 3.12: Types of Blockchain Used in the Selected Research Papers

3.4.12 Smart Contracts

Smart contracts, an essential feature of blockchain technology, were addressed in the selected papers with varying frequency. The majority of the papers (43.9%) did not provide details regarding the smart contract platforms used. Solidity, a popular smart contract language associated with Ethereum, was employed in 31.7% of the papers that explicitly mentioned the smart contract platform. Chaincode, typically used in Hyperledger Fabric, was featured in 19.5% of the papers. Python and JavaScript were the least common choices for implementing smart contracts, appearing in only 2.4% of the papers each. Figure 3.13 shows the distribution of smart contracts employed.

These results of the systematic literature review showcase the emerging nature of patient-centric blockchain-based healthcare architectures as a research area, with a surge in publications in recent years. The journals have been the most trusted means of distribution, highlighting the research peer-reviewed need and thus its desire. Geographically distributed authors, suggests a far and wide spread interest confronted by the bulk of the articles. The dominant academic contributors signifies the need creative and inge-

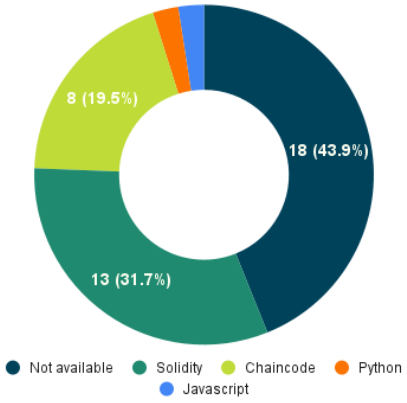


Figure 3.13: Smart Contracts Employed in the Selected Research Papers

nious fixes to be sought after rather than happening on there own. The contributions we identified patient-centric data management, security improvements, interoperability and data sharing. The implementations suggest established blockchains and permissioned blockchains to hold patient data for privacy. The different use cases, the problems, to which they are applied to, and the smart contracts all hint to the multi-pronged nature of deployment of patient-centric blockchain applications.

3.5 Discussion

This section importantly considers the results and their implications, addressing the four major research questions, interpreting the findings in light of the previous literature and highlighting their significance. The gaps in the research are recognised, paving the way for future studies. Figure 3.14 provides a consolidated overview of the key challenges identified in the systematic review and the corresponding future research directions. The mind-map groups the challenges into recurring themes (e.g., privacy/security, interoperability, consent management, scalability, and governance) and links them to the mitigation approaches discussed across the selected studies. This summary is used as a reference point in the remainder of the thesis to motivate where the proposed solutions contribute and where open gaps still remain.

3.5.1 RQ1

The study into enhancing the security and privacy of healthcare data in blockchain-based storage and sharing approach explores nicely. The synthesis of the papers selected reveals a strong focus upon the substantial importance of security and privacy in healthcare data in storage and sharing where it is recognized readily that patient information is vulnerable to external and internal attacks, leading to the concern with data breaches overall [12, 25, 27, 29, 30, 35, 75, 103, 105, 108–111, 113, 118–126]. Several proposed solutions emphasize the need for patient-centric models [13, 14, 25, 29, 30, 35, 108–112, 118, 120, 121, 125–127] ensuring that patients have complete control over their health data [13, 14, 30, 75, 103, 108, 109, 111, 112, 121]. These frameworks usually rely on blockchain technology to make sure of decentralization, transparency, and immutability of patient records. Encryption techniques, such as AES and asymmetric cryptography [128],

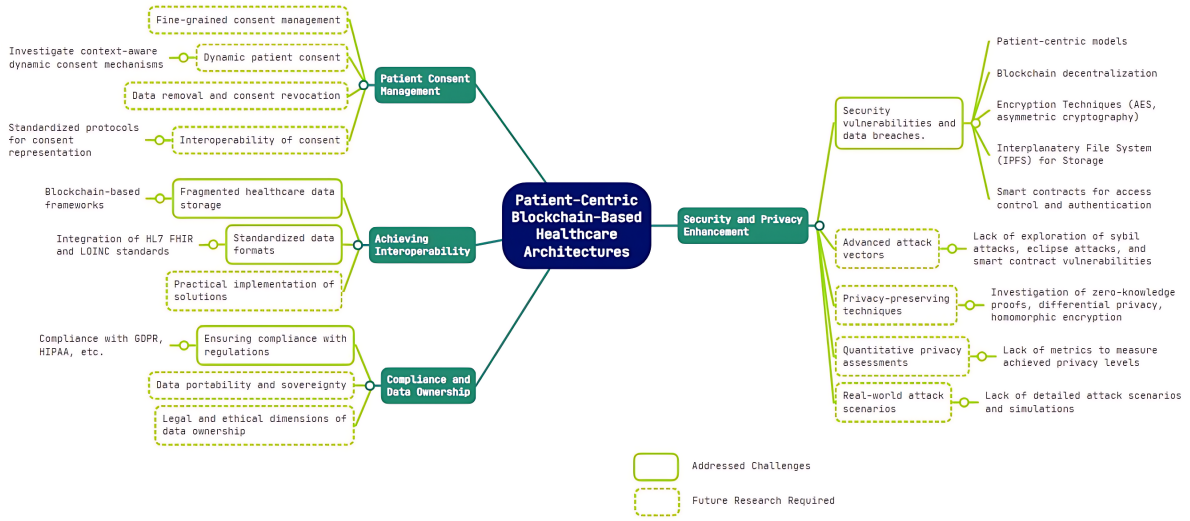


Figure 3.14: Summary of Addressed Challenges and Future Research Directions

are widely adopted to secure sensitive medical data [13, 14, 14, 75, 102, 103, 105, 109, 118–121, 125–127, 129, 130]. Off-chain solutions, such as Inter-Planetary File System (IPFS), are integrated for decentralized and secure data storage [13, 14, 25, 27, 29, 30, 75, 102, 103, 105, 110, 111, 113, 118–120, 126, 129, 130]. Smart contracts are employed for access control, authentication, and data sharing [12–14, 25, 29, 30, 35, 75, 102–105, 108, 110, 111, 113, 120–122, 125–127]. Furthermore, research addresses the use of trust-building mechanisms, audit trails, and easy-to-use interfaces for accountability and user-friendliness. While most papers aim at secure access, data sharing and storage, a few approaches explore other aspects, such as patient identity management, telemedicine, interoperability, and remote patients monitoring, and data privacy. In essence, through the synthesis of all research papers, they collectively conclude that patient-centric blockchain healthcare data management constitutes a promising solution to enhance not only its security but also its privacy. Relying on encryption, decentralization of storage, access control, and trust-building mechanisms, these studies offer a holistic approach that empowers patients, yet shields them from threats to their sensitive data.

Finally, the review of the research landscape with regards to blockchain-based sharing and storage of healthcare data reveals several important dimensions that would serve as gaps that also need to be addressed to enhance the field's existing security and privacy. While we have touched on a few, some not very obvious ones, there nevertheless remains some important areas that are less addressed, and which researchers should take a close look at [131].

- While most of the attacks considered in the literature have been enumerated as being common threats to blockchain systems, particularly healthcare-based ones, such as the 51% attack, selfish mine, malleability of transaction data, and deanonymization [109]. There exist many others beyond those discussed here in various papers and texts, some being sybil attacks, eclipse attacks, and advanced smart contract vulnerabilities.

- While many of the works that were able to be read have listed encryption as a mean to secure data, few if any, have looked at privacy-preserving techniques such as zero-knowledge proofs, differential privacy, homomorphic encryption, and others.

Investigating these techniques and their applicability to healthcare data sharing would strengthen the overall privacy framework.

- While some works have recognized the need for privacy, very few have quantitatively assessed the level of privacy that could be offered by the approach they are suggesting. One possible approach of strengthening the assessment of proposed blockchain-based healthcare systems is incorporating robust metrics that quantify the degree of privacy protection that different systems provide. This could include metrics based on information theory, complexity theory, and so on.
- For the most part, studies have not explicitly looked at potential attacks that could be made against the systems they propose which would compromise patient data security and privacy, including detailed attack scenarios and simulations that can be instrumental in assessing the resilience of proposed systems and help in devising countermeasures.

Thus, by knowing these gaps and taking lessons from such attack vectors, novel privacy solutions, and real-world use cases, the privacy and security of healthcare data in blockchain-based systems can be significantly improved. This approach will help to ensure a more robust framework for securing patient information.

3.5.2 RQ2

An analysis of these works reveals a variety of approaches and techniques to properly consent patients regarding data sharing in blockchain-based architectures where patients play a crucial role in both consenting and controlling. Even though patient consent is not specifically talked in all papers, it can be drawn that patient-centric control and privacy are main themes in various proposed solutions.

The surveyed research papers propose a variety of methods to address this; many papers describe systems where patients retain full control of their health data. Blockchain will allow patients to grant/revoke access to healthcare providers with the result that sharing can only take place through releasable consent from the patient [12, 14, 28–30, 75, 102–104, 110, 111, 119, 121, 124–127, 130]. Several papers make point of implementing smart contracts for enforcing patient consent and access control. Smart contracts assure of permissioned data access as such the provider can only attempt to read patients' data through their prior access with the patient's consent [13, 29, 35, 75, 111, 113, 118, 120, 121, 125, 130]. Some papers describe how patients should be represented and identities managed through blockchain architecture. We suspect this without explicitly mentioning that of course control over their access is also control over release of information about them, implying that proper management of their data is synonymous with granting consent [35, 122, 125, 127]. Several papers describe electronic consent granting access to their data and assure functionality that permissions will only take place where there is proper granting of consent that's also securely written to the blockchain [27, 35, 127, 130].

Currently the research in most of blockchain-based architectures for healthcare is inarguably centered around the two premises of patient-centric control and privacy. Patient consent is built into the systems, given that they may manage and grant access to their health data via various means, like smart contracts, encryption, electronic consent, etc. and this ultimately leads to more focus on enhancing patient autonomy and safeguarding of data privacy from when they were devised as stores of delicate information. Patient-controlled consent mechanisms present in existing research on blockchain-based healthcare

architectures, are namely traceable consent history where upon through transparency we know that consent has been feasibly rendered by the queried party, electronic consent records, which are securely recorded on the blockchain, and as well as how patients manage access, or rather have the management of access under their control, which empower them to be the primary source of consent. Some areas currently unexplored in research:

- While several papers speak for the importance of patients to have control over their data sharing, there is currently a lack of work aimed at more fine-grained control of consent management via models that permit patients to define what health systems get access to what kind of data. Future work will be focused on these kinds of consent models that exhibit larger degrees of privacy over the patients' information via customization [132].
- Dynamic patient consent is an emerging area of research, and most research is focused on more static or initial consent processes. Future work can be aimed at context-aware dynamic consent methods/models. Patients will be able to grant or revoke consent to access, based on changing situations such as purpose of use, timeframe, etc. and specific healthcare situations.
- Most research so far has dealt mainly with data access control (i.e. ability to restrict access), less focus has been given to papers discussing data removal, or revocation of consent. Future studies can focus on how patients can call for access to their data to be removed from the blockchain system and abide to fulfill data integrity and opacity requirements.
- Interoperability of consent over various different platforms is a related but lesser studied area. Standardized protocols can be looked into around consent representation and "sharing" to provide seamless interoperability amongst systems.

3.5.3 RQ3

The insights gained from the examination of strategies such as standardization of data formats and protocols, and integration of standards across different domains for enhancing patient-centric interoperability in healthcare systems are presented in this section. The analysis of the research papers collectively shed light on the criticality of interoperability as a key challenge in modern healthcare systems. The interoperability issues are often caused by the cluttered separation of healthcare data from storage systems, the absence of standardized data formats, and the necessity of secure communication for smooth information transfer between stakeholders like patients, healthcare providers, and institutions.

An idea that comes up repeatedly in the selected papers is the use of blockchain technology to create patient-centric interoperability. Blockchain allows secure decentralized storage of health records allowing the patient to control their data. MediTrans [14], MedHypChain [121], and VigilRx [110] are some examples of systems that suggest use of a blockchain-based framework for interoperability and ability to share patient data with healthcare. Also, more sophisticated patient-centric architectures [28, 105], allow the unified co-location of patient data from different sources, fostering interoperability and enabling faster data exchange. The necessity for data standardization is underscored by several studies. The data of patients often reside in a heterogeneous system using

different formats. Here are common methods to stimulate interoperability in the papers we examined:

- Interoperability can be bolstered by the use of established standards and frameworks such as Health Level Seven(HL7), Fast Healthcare Interoperability Resources (FHIR) and Logical Observation Identifiers Names and Codes (LOINC) [12, 27, 75, 122, 126, 129]. The implementation of such standards helps close the chasm between different healthcare providers and systems, ensuring data is exchanged and interpreted consistently [12, 126].

- Many accommodate the availability and accessibility of data by utilizing decentralization: Interplanetary File Storage (IPFS) is used in various papers [104], [108], in order to facilitate this distributed data storage, providing tamper-resistant data.

- In addressing the difficult problem of interoperability, one mechanism typically relies on blockchain relays i.e. smart contracts running on one blockchain that can verify events on another blockchain, forming a type of protocol. The other involves trusted gateways that relay to participants the details of a cross-chain transaction, relevant mostly for private blockchains. Beyond this, the notion of a "blockchain of blockchains" has draw interest to achieve interoperability of disparate blockchain [133].

Even if blockchain based solutions seem to sparkle for better interoperability, we need more real-life implementations and evaluations of their effectiveness. Many of the reviewed papers focus on certain aspects of interoperability, needing holistic studies of the whole healthcare ecosystem. Besides, standards like FHIR improves interoperability, but assessment on different healthcare settings demands an analysis. The healthcare systems in different domains may be better off being interoperable through several standards and protocols [12], [126].

To encapsulate, patient-centric interoperability in healthcare requires the coming together of blockchain, standardized frameworks, and patients managing access to health data. The current body of literature provides valuable insights, there is a need for further such explorative studies.

3.5.4 RQ4

Designing a patient-centric architecture based on blockchain technology for health is also very essential to assuring adherence to healthcare data laws and giving clear data ownership to patients. Most of the presented systems try to emphasize the privacy of patients' records and ownership facilitated with the help of blockchain-based solutions [28, 102, 104, 108, 109, 113, 118]. In these systems, patient can control who has access to their health data and how they share their health data [112, 119, 126]. Patients are empowered to grant or revoke access, guaranteeing that only authorized parties can access their medical records [105, 120, 121]. The following are the observations made by analyzing the current research:

- Numerous systems are geared towards compliance of regulation appropriate to handling healthcare data such as General Data Protection Regulation (GDPR) , and, Health Insurance Portability and Accountability Act (HIPAA) [12, 14, 27, 29, 35, 104, 109, 118]. Systems which leverage on the tamper-proof nature of blockchain and

access control help for reaching compliance [30, 111, 129]. Permissioned blockchain networks, help in maintaining data integrity as well as privacy, and combined with a regulatory aspect [25], [125], [127].

- Future research could investigate how well blockchain-based healthcare systems line up to develop regulations such as GDPR and HIPAA and specific region laws. Finding a strong and resilient regulatory framework inside blockchain solutions for overcoming likely regulatory potential challenge would be essential.

- Even with the great benefits of blockchain in enabling patient data ownership, ensuring the portability and sovereignty of that data across different healthcare providers and platforms remains a complex issue. The mechanisms behind these are needed to provide smooth data transfer between the various healthcare systems and also have need for further investigation.

- The ethical and legal issues that stem from patient-controlled data ownership also require exploration [134], including how would the legal framework deal with issues of dispute, liability and situations of data breaches in decentralized healthcare systems.

3.6 Summary

Concluding, this work presents the maturity of existing research through a detailed review on patient-centric blockchain-based architectures. The review uncovers the high interest and ongoing research in this very dynamic and growing field. The analysis presented shows that most selected research papers deal with the enhancement of the security and privacy of a patient's data in a blockchain system, revolving around patient-centric models and encryption techniques. Patient consent management is the main theme that various authors touch about, the various way of using a blockchain or a smart contract to assure to the patient explicit control on what information is shared, when and for which purposes. Interoperability is indeed a key challenge in modern healthcare, accomplishing patient centricity on data exchange; here blockchain is presented as one of the ways it can be resolved. In the healthcare compliance and regulation issues, blockchain's tamper-proof nature and access deal with these challenges.

Although the analyzed papers are very enriching, other topics remain to be investigated, and opportunities for continued research could be discussed, such as a more profound studying of advanced security threats, combining the privacy-preserving techniques with quantitative assessment regarding what level of privacy is achieved therefore, the real scenarios and kinds of attacks, all need to be investigated. The research can further jump into subtle consent model, standardization protocols, and data revocation and removal mechanisms. Achieving seamless and uninterrupted data exchange and communication among all stakeholders in the network require comprehensive solutions that help to address the whole healthcare ecosystem. Furthermore, the alignment of blockchain-based healthcare systems with evolving and harsh healthcare data regulations can further improve the scrutiny. Taken together, the findings across RQ1–RQ4 indicate that patient-centric blockchain systems must be designed by co-optimizing security/privacy, smart-contract based consent enforcement, standards-driven interoperability, and governance for regulatory compliance, rather than treating these dimensions in isolation (see Figure 3.14). At the same time, the literature highlights recurring trade-offs between transparency

and confidentiality, immutability and consent revocation, and strong access control and scalability/performance in real deployments. Consequently, this review synthesizes the following design requirements for the proposed architecture: a permissioned setting with encrypted off-chain storage, fine-grained dynamic consent (including revocation) enforced via smart contracts, and standards-aligned data exchange (e.g., HL7/FHIR) with auditable policy enforcement to support compliance.

By addressing these gaps and these challenges, the field of patient-centric blockchain-based healthcare architectures can achieve more robust, secure and patient-centric solutions that authorized individuals while safeguarding their sensitive health data. The gaps identified in this review, particularly around interoperability, fine-grained consent management, and scalability of blockchain-based healthcare systems, motivate the design of a new patient-centric architecture. Chapter 4, therefore introduces DiabeticChain, a permissioned blockchain framework that directly addresses these challenges.

Chapter 4

DiabeticChain: A Novel Blockchain Approach for Patient-Centric Diabetic Data Management

This chapter proposes a permissioned blockchain-based patient-centric healthcare architecture that enables patients to control access to their diabetic data named DiabeticChain. To ensure secure sharing while preserving privacy, DiabeticChain allows Data Consumers to access patients' data with their specified consent policy, assembled using smart contracts which is developed, verified, and deployed using the Truffle framework. This study includes a proof-of-concept using Go-Ethereum to establish a permissioned blockchain and Chainlink for secure smart contract connectivity to off-chain resources. A hybrid model is used for data storage to store confidential diabetic data in an off-chain secure database while keeping only metadata on the blockchain. The performance was evaluated using 4 Azure hosted Virtual machines (VMs) and Hyperledger Caliper, showing throughput and latency as key performance metrics to calculate effectiveness.

4.1 Introduction

Diabetes is a chronic disease that brings with it an enormous burden of self-care. According to the International Diabetes Federation, there are 537 million adults aged 20-79 living with diabetes, with projections of 643 million by 2030 and 783 million by 2045 [135]. This frightening rise shows a worldwide public health crisis that is showing no signs of slowing. The areas with the fastest-growing diabetes prevalence are currently in Southeast Asia and Africa [74], impacting immediately the already stressed healthcare systems.

Even if the life expectancy of people with diabetes is lower than the ones without the disease, the life-time medical cost of diabetes is still significant [136]. Diabetes patients are dependent on daily care, from choices of diet, to exercises, regular medications, and continuous monitoring of sugar levels. This data is heterogeneous and scattered, and a patient does not have control over the use of data. As shown in SmartCHANGE Project [41], InteropEHRate Project [1] and beHEALTHIER Project [134]; data sharing is complicated. CrowdHEALTH [2] further demonstrates the importance of health records in one place, proving the need of an integrated solution like DiabeticChain.

4.1.1 Motivation

Healthcare data is captured and generated from multiple sources, ranging from the patients themselves to wearable devices like smartwatches, smart bands, smartphones, etc. With increasing technological advancements, IoT devices like wireless sensors are being

used to monitor continuous patient health, generating huge amounts of personal, valuable, and complex data. Providing controlled access to complete data can significantly strengthen the entire healthcare system. Since patient data is possessed by trustless third-party authorities like hospitals, laboratories, and health insurance companies, secure and efficient data sharing is not possible. There exists no method for verifying the medical data sharing by hospitals to an untrusted third party [40]. This creates vulnerabilities in data security, increasing the possibility of unauthorized access.

There also exists a possible risk of a data breach due to the existence of a single point of failure. According to existing reports, 89% of users using IoT in healthcare have suffered security breaches, and 80% of them are still willing to use this technology [42]. Unauthorized access through malicious attacks on diabetic healthcare data can have disastrous results like medical identity theft, insurance fraud, and tampering with ongoing treatment or medical history. A secure and efficient digital EHR data sharing system is thus required. Therefore, this study discusses a blockchain-based method for patient security and medical data ownership that aims to overcome the limitations of conventional data sharing models by minimizing cyberattacks, fostering confidence, and managing diabetic data effectively.

There are some unique challenges as well associated with diabetic data, like diabetes is a disease that requires self-care by substantially burdening the patients with continuous glucose monitoring and carbohydrate counting [137]. As per the research in [43], an average adult with type 2 diabetes is recommended to spend 4 hours per day as the total estimated time for self-care. Being a chronic disease, diabetes also burdens the healthcare providers to analyze and access the huge amount of data associated with it [137]. While numerous researchers have worked in medical data sharing, there exists a need to specifically address these unique challenges associated with the dissemination of diabetic data among stakeholders. This gap in the existing literature serves as the primary motivation of this research.

4.1.2 Novel Contributions

This section mentions all the novel contributions of the proposed scheme as follows: Blockchain is a decentralized distributed ledger technology that has the potential to revolutionize the way confidential data is shared and accessed today by providing benefits like immutability, transparency, traceability, and enhanced security [138]. In this thesis, we propose a system—DiabeticChain—a patient-centric healthcare architecture to ensure secure and efficient data sharing of diabetic data, provide an implementation as a proof-of-concept study, and then verify its performance using various assessment metrics. Patient data can be accessed with their consent using a one-time access URL, making our approach novel and viable as a solution.

The author aims to address the present limitations of conventional data sharing models by switching to a decentralized technology on blockchain, for the diabetic data sharing. DiabeticChain aims to address the requirements of patient privacy, data security, user control over data, traceability, data integrity, interoperability, and standardization. The following outline highlights the key contributions of this work:

- A novel blockchain-based patient-centric healthcare architecture is proposed for efficient and secure diabetic data sharing among stakeholders like healthcare providers, lab technicians, health insurance organizations, and researchers.

- The proposed architecture has been implemented as proof of concept using a private permissioned Ethereum blockchain that overcomes the current challenges such as privacy, data ownership by patients, fragmentation of data, and data integrity in current healthcare models.
- The proposed architecture performance is evaluated based on throughput and latency assessment metrics to ensure its feasibility and reliability in real-world health applications.

4.1.3 Dynamic Consent and Blockchain

Dynamic Consent is the approach that involves collecting, storing and managing of data obtained from people for the purpose of research and healthcare. This approach is very different from the traditional consent in which the data was collected from the people only at the beginning of a study that is taking place but on the other hand dynamic consent approach involves dynamic or continuous flow or control of the data for various research activities or purposes. Despite such flow, dynamic consent has limited dimensionality in the practical field due to the security and legality of the personal data belonging to people. This poses major trust issues for the people including the fear of maintaining the confidentiality [14, 110] of the data that belongs to their personal belongings. Now, Blockchain deals with the rapidly multiplying part of data in the form of a decentralized database. [111] Blockchain has various functionalities which could support dynamic consent in different ways like high accountability, a greater sense of data security and privacy, immutability, and an extremely efficient management system [104].

Diabetic data sharing takes place in a continuously changing environment and therefore dynamic consent is necessary for personal protection of sensitive data, preventing the unforeseen risks of privacy and driving patient-centric research [112]. Blockchain can enable data sharing in a trusted environment according to the modifiable consent preferences of the patient where they can dynamically update their consent options anytime.

4.2 Design Requirements

In order to implement our proposed architecture, five design requirements have been identified in accordance with our blockchain use case. The design requirements for DiabeticChain were derived through a combination of stakeholder analysis and literature review. These requirements prioritize data security, patient control, and system scalability, reflecting the needs of both patients and healthcare providers.

4.2.1 Requirement 1: Data Security

Data security is a crucial requirement in sharing sensitive diabetic data which is susceptible to cyberattacks. Both active (taking control of medical devices) and passive (overhearing the communication through wireless devices) attacks can occur on the collected diabetic data [139]. Patients' data should be stored in a secure database to eliminate the possibility of malicious attacks and prevent the misuse of data by attackers. Data encryption techniques like public key cryptography and access control mechanisms have to be employed to eliminate the unauthorized access and hence ensure data security [16, 39, 118, 119, 140].

4.2.2 Requirement 2: Patient Privacy

Patients' diabetic data should be shared with healthcare professionals and researchers with proper authentication while minimizing the disclosure of their identity and without compromising their privacy. Only the authorized users should have access to the confidential diabetic data of patients with their consent [31, 120, 121, 140].

4.2.3 Requirement 3: User Control Over Data

Healthcare has moved from document-based storage to EHRs but the ownership and control of confidential data is still in the possession of unknown, untrusted, centralized authorities like hospitals with a possible risk of data breaches [37]. Diabetic data of the patients should be owned, controlled and managed by the patients themselves resulting in patient-centricity instead of being owned by trustless centralized servers like hospital databases. When patients are the owner of their data, they will grant permission to anyone who wants to access that data and how their data will be used [29–31, 69].

4.2.4 Requirement 4: Traceability and Data Integrity

Data traceability and data integrity are critical concerns in data sharing among stakeholders. The Data Owner i.e., the patient should be able to track how their healthcare information is being shared and give their dynamic consent as per the purpose of data utilization to achieve transparency. During the process if any malicious node tries to manipulate the health record, then it should be detected and prevented from any change ensuring data integrity [32, 122–124].

4.2.5 Requirement 5: Interoperability and Standardization

Along with the implementation of the proposed model, it should be feasible to integrate it with the existing healthcare systems consisting of Electronic Health Records (EHRs) to achieve the requirement of interoperability which allows the different decentralized servers operating in different locations to interact and share information over a blockchain network. In order to achieve interoperability among various entities, there is a requirement of a standard format to capture the diabetic data and thereby ensure efficient data sharing [16, 35, 125–127].

These design requirements are addressed accordingly in subsection 4.5.1.

4.3 System Architecture

This section discusses the detailed system architecture for blockchain-based patient-centric diabetic data sharing and its integration with existing EHR systems for practical implementation in healthcare environments.

4.3.1 System Components

The constituting components of the proposed architecture are depicted in Figure 4.1. The diagram clarifies the roles of the patient (data owner), healthcare organization (data controller), and authorized requester (data consumer), and shows how each interacts with

the system during consent-driven data sharing. It also separates on-chain components responsible for consent/policy enforcement, metadata pointers, and audit logging from the off-chain hospital/EHR infrastructure where sensitive clinical records are stored to preserve privacy and avoid unnecessary on-chain storage.

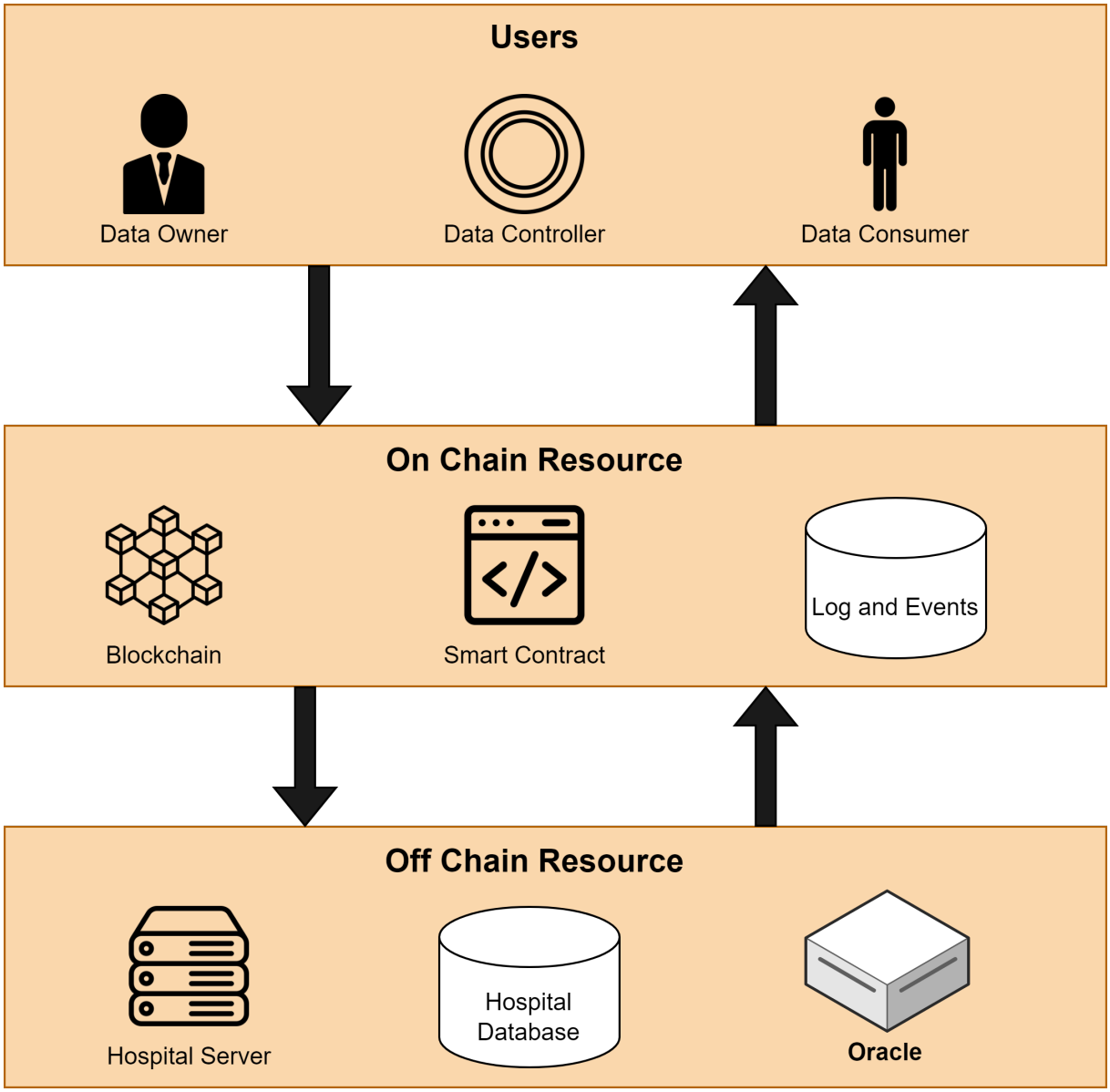


Figure 4.1: System Components

- **Users:** In our architecture, we have three types of users who can interact with the system.
  - **Data Owner:** A patient, who is the owner of the data, has access rights and permission to remove the previously granted access.
  - **Data Controller:** An organizational entity, such as a hospital or lab, captures and stores patient data in their secure off-chain databases maintained within an existing EHR system.

- **Data Consumer:** Individual entity, such as healthcare organizations, academicians, or medical policy agencies, requires patient data for their specific needs and seeks access to it.

- **On-chain Components**
  - **Blockchain:** The distributed and decentralized database ensures the privacy, immutable, and transparent repository and sharing of diabetic data.
  - **Smart Contracts:** It offers the system features like user registration, EHR metadata storage, and patient access policy management, among others.
  - **Logs and Events:** Smart contracts generate events and logs in order to maintain and track information for system data auditing.

- **Off-chain Components**
  - **Hospital Server:** The server interfaces with the hospital's existing EHR system to ensure seamless integration of the blockchain with the existing patient data storage architecture.
  - **Hospital Database:** An off-chain database is maintained for patient's record privacy, which is controlled and managed by the data controller. This data remains within the EHR system, and only metadata is shared with the blockchain, ensuring compliance with privacy regulations.
  - **Oracle:** Since Smart Contracts and Blockchain cannot access the off-chain data, an oracle is used as a trusted data service to read the data to the blockchain. Oracle enables secure communication between smart contracts and the database.

4.3.1.1 Integration with EHR Systems

The proposed system architecture supports integration with existing Electronic Health Records (EHR) systems through a middleware layer that interfaces between the on-chain and off-chain components. The middleware layer uses blockchain to enhance security and interoperability and it can be used by hospitals and healthcare services while maintaining their existing EHRs. The APIs ensures data consistency while smart contracts handle data access when the interaction between blockchain and EHR system happens. This allows for a smooth data exchange without creating and issues in the current EHR workflow.

4.3.2 Proof-of-Concept: DiabeticChain

This section, we propose a proof-of-concept based architecture, DiabeticChain, which enables the patient to grant or revoke their consent to the data consumers and also to the data controllers to collect and store their diabetic data. Oracle service is integrated to facilitate secure communication between off-chain data sources and on-chain smart contracts. Oracle acts as trusted third-party data providers to fetch external data, which smart contracts on the blockchain cannot directly access. This is critical for retrieving patient records stored in hospital databases during the data access process in a tamper-proof manner. Figure 4.2 summarizes the end-to-end workflow executed when a data consumer requests access to a patient record. The sequence highlights where consent and access-policy checks occur on-chain, how the smart contract coordinates with the

oracle to retrieve the corresponding off-chain record, and how the response is returned as a controlled (encrypted, one-time) access artifact. This overview helps interpret the step-by-step workflow described below and makes explicit which operations are handled by smart contracts versus external services.

Trust and threat assumptions: DiabeticChain assumes a permissioned blockchain setting where validators are known entities and a majority behave honestly (no sustained majority collusion). The hospital database/IPFS layer is treated as untrusted for confidentiality and integrity (data may be read, copied, or tampered with), hence records are encrypted and only verifiable pointers/hashes are anchored on-chain. The oracle is semi-trusted, it is trusted for correct execution of retrieval and delivery of encrypted artifacts, but not trusted to learn plaintext; if compromised, it may attempt to misdeliver, delay, or deny service, which is mitigated by on-chain policy enforcement, encryption, and audit logs.

The following steps demonstrate the workflow of DiabeticChain:

1. **Patient Registration:** All stakeholders register on the system through the User Smart Contract (USC) and acquire a unique identifier number.

2. **Patient Consent Setup:** The patient defines the access policy using the Access Policy Smart Contract (APSC), which allows them to specify who (Data Consumers) can access their data and under what conditions and for how long.

3. **Data Collection by Data Controller:** The Data Controller (such as a hospital)collects the patient's diabetic data. This data is stored in an off-chain secure database managed by the hospital.

4. **Metadata:** The metadata serves as a pointer to the off-chain data, which is stored in the Metadata Smart Contract (MDSC) ensuring that the patient's sensitive information is not stored directly on the blockchain.

5. **Search by Data Consumer:** The Data Consumer (such as a healthcare provider or researcher) searches for relevant patient data by querying the blockchain. The relevant patient metadata stored in MDSC is identified.

6. **Access Request by Data Consumer:** Once the relevant metadata is identified, the Data Consumer requests access to the data using the Access Management Smart Contract (AMSC).

7. **Policy Validation:** AMSC checks the patient's consent policy from APSC on policy request for access. If the data consumer request is equal to the patient's access policy, the access request is approved. If not, the access request is denied.

8. **Oracle Service Invocation:** After approval, the AMSC sends the transaction to the oracle service to obtain the necessary off-chain patient data from hospital's secure database.

9. **Data Retrieval by Oracle:** The oracle obtains the patient data from the database at the hospital, which are encrypted. It also obtains the public key of the data consumer from the USC.

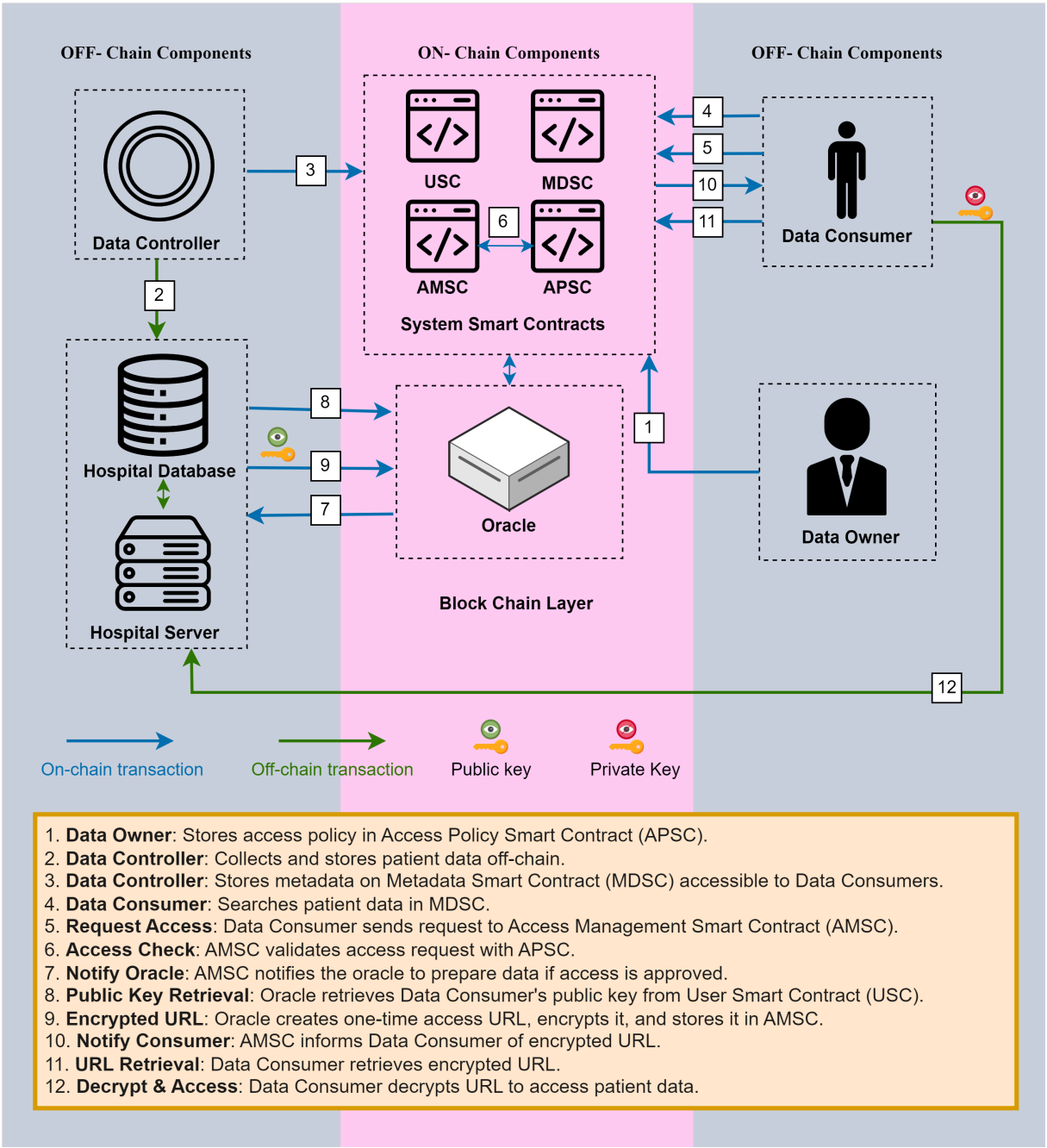


Figure 4.2: Detailed Workflow of the Proposed Architecture

10. **One-Time Access URL Generation:** Based on the data requested by the data consumer, the oracle generates a one-time access URL and sends it to the data consumer after encryption using the public key of the data consumer.

11. **Access Notification:** The encrypted one-time access URL is stored in the AMSC and it notifies the data consumer that the data is ready for them to retrieve.

12. **Data Access by Consumer:** The data consumer now retrieves the one-time access URL and decrypts it using his private key and now can securely access the diabetic data of the patient. The access is defined according to the scope in patient's consent policy.

This 12-step workflow builds secure and efficient data sharing, while maintaining patient privacy and control over their sensitive diabetic data.

4.4 Implementation

The proof-of-concept of the proposed blockchain-based architecture was implemented on a privately permissioned blockchain to showcase its viability. At the infrastructure level, Go-Ethereum [130], an open-source Ethereum client enabling permissioned private blockchain networks, was utilized to establish a private blockchain. The system's smart contracts are written using Solidity, and the Truffle framework was employed for developing, compiling, testing and deploying these smart contracts. The patient's smart contract code is partially illustrated in Figure 4.3. Chainlink was used as an oracle service to connect smart contracts to off-chain resources, and MongoDB was employed to develop the off-chain database.

There are four smart contracts to handle the blockchain transactions: User Smart Contract (USC), Access Policy Smart Contract (APSC), Metadata Smart Contract (MDSC), and Access Management Smart Contract (AMSC). These smart contracts offer 6 primary system functions: createUser, setPatientConsent, cancelPatientConsent, checkPatient-Consent, createMetadata and requestAccess. Only authorized users are limited to calling these functions which are implemented using smart contract modifiers. If a function call is unauthorized then it will result in the termination of the function execution and the reversion of all modifications to their initial state. The subsequent section elaborates on the use of smart contract functions to implement the main system functionalities.

4.4.1 User Smart Contract (USC)

This smart contract is responsible for user registration into the system. Every system participant engages with the system using their own wallet, which contains all necessary information for system interaction. As a result, participants must be registered within the system, where a smart contract is established. The role of the system admin is to verify the users' identities and professional registrations before initiating the system registration process. Algorithm 1 describes the user registration process.

Algorithm 1 Pseudo-code for registering a new user (RegisterUser)

1: **Input:** initiator, userWalletAddress, role
2: **Output:** userIdentifier
3: **if** $initiator = owner$ **and** $userWalletAddress \neq null$ **then**
4: Require ← newUser
5: Display "User already approved!"
6: Create newUserProfile
7: **return** $userIdentifier$
8: **else**
9: Display an error message and undo smart contract state

```
pragma solidity ^0.8.0;

import "./MDSC.sol";
import "./USC.sol";
contract APSC{
    mapping(uint256 => AccessTicket) public accessTicket;

    mapping(bytes32 => bool) public accessTicketSigns;

    uint256[] public accessTicketIds;

    uint256 public lastAccessTicketId =200;

    mapping(bytes32 => Consent) private consent;
    bytes32[] public consentSigns;

    struct Consent{
        bytes32 consentSign;
        bool status;
        bytes32 MetaData;
        bytes32 role;
        bytes32 aim;
        uint256 timestamp;
        uint256[] issuedAccessTicket;
    }

    event consentlog(
        address indexed patientConsent,
        bytes32 _consentSign,
        uint256 _timestamp,
        bool indexed _status
    );
    struct AccessTicket{
```

Figure 4.3: Partial Code of the Access Policy Smart Contract (APSC)

4.4.2 Access Policy Smart Contract (APSC)

This smart contract keeps patient consent as access policy tree. A consent signature is produced by hashing the access policy tree and stored in APSC. Hashing and storing patient consent in APSC allows for efficient consent status retrieval and validation. Algorithm 2 describes the process of setting the consent of the patient.

Algorithm 2 Pseudo-code for saving consent of the patient. (SavePatientConsent)

```
1: Input: initiator, aim, role
2: Output: status
3: PatientConsent, consentSign ← mapping
4: if initiator = owner AND aim ≠ null AND role ≠ null then
5:     h ← hash(role, aim)
6:     if PatientConsent.contains(h) = true then
7:         Display an error message and undo smart contract state
8:     else
9:         PatientConsent.add(h, true)
10:        return true
11: else
12:    Display an error message and undo smart contract state
```

4.4.3 Metadata Smart Contract (MDSC)

This smart contract contains patient metadata. Once the patient data has been collected and securely stored in an off-chain secure database such as a diabetic laboratory database, the data controller then submits the patient's metadata to the system. This metadata is a representation of the patient data that contains a description without disclosing any confidential or identifiable patient information, like the data type, hash of the stored data and conditions. The patient metadata is then kept in a separate data structure, where the hash of the stored data serves as the key and the remaining patient data as the value. Algorithm 3 shows the process of creating patient metadata.

Algorithm 3 Pseudo-code for generating patient metadata. (GenerateMetadata)

```
1: Input: initiator, patientWalletAddress, dataHash
2: Output: id
3: Metadata ← mapping
4: i ← counter
5: if initiator = dataControllerWalletAddress AND patientWalletAddress ≠ null AND
   dataHash ≠ null then
6:     i ← i + 1
7:     Metadata.add(i, [patientWalletAddress, dataHash, dataControllerWalletAddress])
8:     return i
9: else
10:    Display an error message and undo smart contract state
```

4.4.4 Access Management Smart Contract (AMSC)

This smart contract is responsible for managing access to patient data. The Data Consumer should obtain an Access Ticket (AT) to be able to access the patient information. Using the AT, the access to the requested data is limited to the lowest possible level. Once the data consumer identifies relevant patient metadata, they have to request an AT from the system by providing the hash of the data they requested, the reason for requesting access to patient data and their role. This request is then verified by APSC, which stored the consent policy of the patient. If the consent is valid and aligns with the

data consumer's request, an AT is automatically generated for data consumer. Algorithm 4. shows the process of requesting access to off-chain relevant patient data.

Algorithm 4 Pseudo-code for requesting permission for accessing off-chain patient information. GenerateMetadata (RequestAccess)

```
1: Input: dataConsumer, Metadata, aim, role
2: Output: ticketId
3: METADATA ← mapping
4: if dataConsumer = owner AND Metadata ≠ null AND aim ≠ null AND role ≠ null then
5:     d ← data consumer.get(MetadataID)
6:     patient ← d.patientWalletAddress
7:     h ← hash(role, aim)
8:     if PatientConsent.get(h) == TRUE then
9:         ticket ← patient.CreateAccessTicket(dataConsumer, MetadataID)
10:        ticket.status ← TRUE
11:        return ticket.id
12:    else
13:        Display an error message and undo smart contract state
14: else
15:    Display an error message and undo smart contract state
```

4.5 Results and Evaluation

In this section, we discuss the important findings and outcomes received from the implementation and evaluation of the DiabeticChain proof-of-concept. The results are analyzed by addressing the design requirements, performing the security analysis and the performance evaluation.

4.5.1 Addressing Design Requirements

In this section, we discuss how our system fulfills each design requirement and addresses them by implementing various features using the Ethereum blockchain and smart contracts.

4.5.1.1 Data Security

We have proposed that the sensitive diabetic health data should be stored in the secure off-chain database, and only the metadata with a reference pointer to the source data, encrypted by public key cryptography, should be stored on the blockchain. Data consumers can request access to the patient's data, and the patient can then grant or revoke access as per their consent policy. Cryptography methods and access control mechanisms ensure the security of data in DiabeticChain and thereby prevent the risk of cyberattacks.

4.5.1.2 Patient Data Privacy

Our system, DiabeticChain, stores the sensitive record of the patient in a secure off-chain private database to preserve privacy, while its metadata is stored on the blockchain, which is encrypted using public key cryptography. Every user, i.e., Data Owner, data consumer, and data controller, is required to register in the User Smart Contract on the blockchain to authorize and validate their participation in the network. An access control mechanism has been used to safeguard the data from leakage. Only the data controller can access the data with their respective authorized private key to protect patient privacy, the anonymity of data is maintained while it is shared with hospitals, researchers etc. The patient data is anonymized by randomly generating unique accounts connected to the patient using their public key, so that their real identity is not disclosed. This way, data encryption, anonymization and access control mechanisms are helpful in privacy preservation for the patients.

4.5.1.3 User Control over Data

Patient-Centric Approach: The proposed system, DiabeticChain, helps patients take complete control of their diabetic data. With DiabeticChain, patients can grant access to their diabetic data whomsoever they authorize, in a secure manner. In a decentralized peer-to-peer network, no central authorities will have powers to take control over patient data. It will solely; owned, managed and controlled by the patient and would be shared with the stakeholders on their consent. In this way, the patient is always in control of their sensitive health data. The patients can update their consent policy dynamically as per their changing environments. The system will give them full control over their diabetic data, from real-time access modifications to granting revocation at will of the patient.

4.5.1.4 Traceability and Data Integrity

The data owner can track their data and its usage, as each of the transactions is recorded on the blockchain, ensuring the integrity and traceability of the data.

4.5.1.5 Interoperability and Standardization

While using blockchain features, DiabeticChain models a feasible design that can be integrated with the existing healthcare systems and thus ensures interoperability among systems. Since the credibility of the new records added to the blockchain depends on the credibility of its participating nodes, a form of data standardization is followed so that all the nodes can understand the data [141]. One such standard is HL7 which is a human-readable standard to address the requirement of interoperability and ensure seamless integration and data sharing among different healthcare entities [19]. A potential limitation of using such standards can be the mismatch of versions used among different healthcare centers or possible technological issues which need to be taken care of.

4.5.2 Security Analysis

The security of DiabeticChain is analyzed in this section based on data storage in the system, sharing of data in the system and tamper-proofing.

4.5.2.1 Data Storage

An off-chain secure and private database stores the patient's confidential diabetic data, while the blockchain stores only the metadata, hash, and a reference pointer. During every appointment, hospitals and labs record the patients' diabetic information, including current medication, blood sugar levels, insulin dosage, blood pressure, and other sensitive data. In the context of this study, we have assumed that the off-chain hospital databases securely store and manage this data. Such a large dataset cannot be stored on the blockchain itself since it is not computationally and memory-efficient to manage it [24]. Therefore, blockchain stores only the metadata within MDSC, along with a reference link to underlying data in the secure off-chain database. This division of data thus helps in optimizing the system's performance and security. The proposed hybrid storage model thus addresses the scalability and efficiency concerns associated with blockchain-based systems, such as fulfilling the storage requirements and reduction in transaction processing time.

The storage of metadata on the blockchain gives a comprehensive view of the entire diabetic data to the Data Consumer. Since every node in the blockchain network has a local copy of the data, Data Consumers can easily look for the patient data of interest in the Metadata (stored on-chain) without searching exhaustively in the large off-chain datasets. Even in the case of a server failure or any other technical issue, the hybrid storage model prevents data loss by eliminating single points of failure, handles data backup as sensitive diabetic data is stored across numerous laboratories without storing any redundant data, and thus ensures data availability. Using blockchain for data storage is beneficial over traditional relational databases (RDBs) in building trust, security and privacy aspects. But compared to blockchain, RDBs offer a higher throughput in writing data [128]. Hence, a hybrid model is the most suitable option for data storage in DiabeticChain healthcare system that leverages the benefits of both blockchain and relational databases.

4.5.2.2 Data Sharing

Patient data is stored on the off-chain database and is effectively shared among healthcare entities like researchers, doctors, lab technicians, etc using DiabeticChain. Only authenticated users can request to access the sensitive diabetic data and can only access that data using a One Time Access URL within a given timeframe according to the self-executing conditions coded in the smart contracts. DiabeticChain system also allows the patients to dynamically update their consent allowing the patient to grant or revoke real-time access to their data at any point. The data access request of the data consumer is verified only as per the consent policy of the patient stored in Access Policy Smart Contract (APSC).

When a hospital receives a validated request from data consumer, the hospital generates a JSON format file that consists of the requested patient data and this file is then stored on the off-chain database at a temporary location. It can be accessed using HTTPS but only once within a given short timeframe. Now, the hospital generates a one-time access URL and encrypts it using the data controller's public key obtained from the User Smart Contract (USC). This URL is then stored on the blockchain and upon decryption can direct the user to the JSON file containing the required data. The URL can be only decrypted using the corresponding private key and used to access the data once. So, the Data consumers obtain the URL from the blockchain, decrypts it using its private key and accesses the data. Once the URL is used to access the data, it becomes useless as the JSON file is removed from the temporary storage and now that URL will lead to

nothing. Even if someone retrieves the private key of data consumer to decrypt the URL, the patient data will still be safe. If the one-time access URL is not used within the given timeframe, then also the JSON file will be removed from the temporary storage and the URL expires to prevent unauthorized access attempts. If a need to access the same data arises in the future, data consumer must again submit the request to access the data and a new URL will be generated. This process allows the secure sharing of diabetic data with the patient's consent.

The DiabeticChain system maintains a record of data sharing transactions like who accessed the diabetic data, time of accessing the data, and its purpose and this traceability of the stored diabetic data contributes to the overall security and transparency of the data sharing process by ensuring patient's trust in healthcare providers. In case of multiple Data Consumers leading to large number of data sharing requests, high volumes of data are shared and the system may face an issue of scalability but DiabeticChain provides a reasonable performance, efficiency and scalability with the current version of Ethereum blockchain implemented and hybrid model of data storage used.

4.5.2.3 Tamper-Proofing

Mechanisms and features that make the DiabeticChain system tamper-proof are the immutability of blockchain, the use of consensus algorithms to add new records and cryptographic hashing. Blockchain is an immutable ledger that cannot be changed and generation of a new block requires consensus among all the participating nodes to prevent tampering with data. It is difficult for any malicious node to change the patient records owing to the cryptographic hash used with every block and it forbids any manipulation or modification of the data. But, cyberattacks like Sybil attacks, man-in-the-middle attacks, denial of service attacks and 51% attacks may pose a potential risk to the system.

Sybil Attack is where identity theft occurs in a peer-to-peer network and some node is completely cut-off from their neighbouring nodes, thereby establishing connections with the attacker and this node is then fed malicious information [142]. By ensuring at least one honest surrounding node, we can make our blockchain system resistant to attacks. Increasing the number of validations before verifying a transaction, having multiple paths available on the network and selecting many nodes to check the transaction status can act as countermeasures [143]. Another threat is a denial-of-service attack, which prevents authorized requests by sending more requests than the victim server can handle. As a countermeasure, clients are required to solve puzzles before getting the access to a resource or server [114]. In the 51% attack, the attacker gains access to 51% of the network's computational power while using proof-of-work consensus protocol or it gains control over 51% of the participating nodes when proof-of-authority is used. To prevent these attacks, a proof-of-authority consensus algorithm is used where the validators stake their reputation and it is much more difficult for the attacker to control 51% of the nodes in the network, thus minimizing the risk of potential cyberattacks.

4.5.2.4 Confidentiality, Non-Repudiation and Traceability

DiabeticChain ensures confidentiality through the use of cryptographic techniques, where each patient's data is encrypted before it is stored off-chain. The use of blockchain inherently ensures non-repudiation. All transactions, including data access requests, are logged immutably on the blockchain, preventing any party from denying their actions. Patients can track all data-sharing transactions, knowing who accessed their data and for what

purpose. The system guarantees traceability of the data, with a complete record of all access to patient data – when, by whom, and for what purpose, saved on the blockchain and viewable by both patient and healthcare providers. This increases accountability and data integrity in the healthcare system.

4.5.2.5 Authentication and Authorization

For authentication, only registered users (patients, data consumers, and data controllers), are able to interact with the system. Users are authenticated via secure smart contracts, which handle user authentication prior to accessing data. Authorization is handled by the Access Policy Smart Contract (APSC), where patients have the access to their data. The patients' data is governed by the patients themselves, who determine the rules of access for data consumers and when their diabetic data can be used by data consumers.

4.5.3 Performance Evaluation

To check DiabeticChain's performance, we used 4 virtual machines (VMs), hosted on Microsoft Azure, to establish a private blockchain with the help of Go-Ethereum, which is an open-source Ethereum client enabling permissioned private blockchain networks. Hyperledger Caliper, an open-source benchmarking tool for evaluating blockchain applications' performance, was employed to conduct multiple tests validating DiabeticChain's effectiveness. Table 4.1 displays the private blockchain and testing environment configurations. We tested two primary types of blockchain operations: Transaction and Query operations, using four performance metrics to evaluate DiabeticChain: Transaction throughput, Query throughput, Transaction latency, and Query latency [115]. In comparison to similar systems, DiabeticChain exhibits competitive performance, but it is important to consider differences in evaluation tools and setups, which may introduce variability in performance.

Table 4.1: Go-Ethereum and Testing Environment Configurations

Factor	Setting
Nodes	4 VMs with Ubuntu 20.04 on Microsoft Azure cloud, every VM equipped with 2 Intel CPUs and 8 GB RAM.
Blockchain	3 peer nodes 1 validator node Go-Ethereum v1.10.11
Consensus	PoA (Clique Protocol)
Smart Contracts	Solidity
Benchmarking Tool	Hyperledger Caliper v0.5.0

We conducted 10 test rounds and varied the transactions per second (tps) gradually from 100 to 1000 with an increase of 100 tps with every round and hence the send rate increased linearly from 99.38 tps to 748.22 tps for query operations and from 100.1 tps to 839.76 tps for transaction operations.

Figure 4.4 demonstrates the relationship between throughput and latency for query operations. As the send rate (TPS) increases from 99.38 TPS to 748.22 TPS, the average query throughput reaches 201.7 TPS, with the maximum throughput peaking at 243.54 TPS. The average query latency was stable at 0.246 seconds, with minor fluctuations as the send rate increased, ranging between 0.16 and 0.35 seconds.

Figure 4.5 illustrates the throughput and latency for transaction operations, where more complex blockchain operations are involved. The average transaction throughput

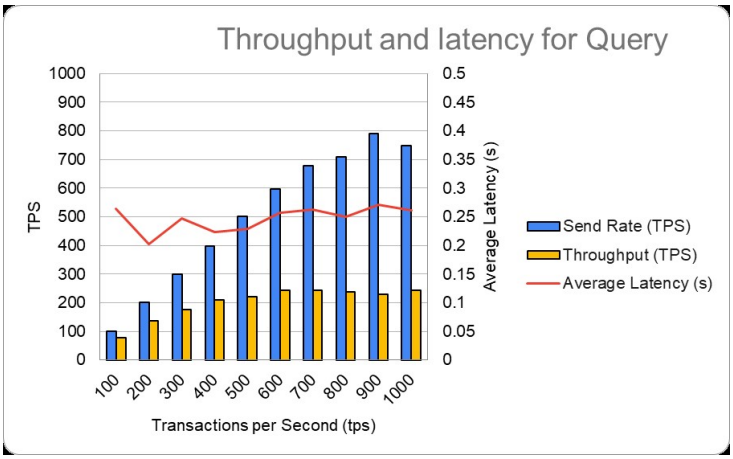


Figure 4.4: Throughput and Latency for Query

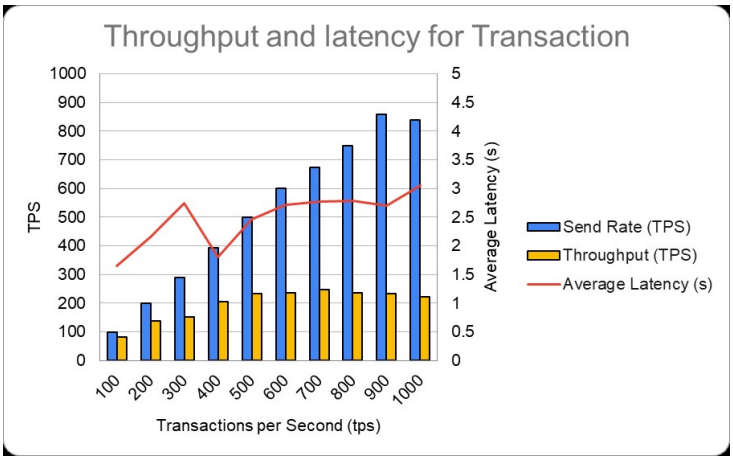


Figure 4.5: Throughput and Latency for Transaction

was recorded at 200.07 TPS, with a maximum of 246 TPS at a send rate of 673.94 TPS. As expected, transaction latency was higher, averaging 1.31 seconds, and ranged from 0.88 to 1.75 seconds as the send rate increased from 100.1 TPS to 839.76 TPS.

4.6 Summary

Sensitive diabetic data is generated in large amounts and needs to be shared effectively among stakeholders since diabetes requires significant self-care and substantially burdens the patients with processes like continuous glucose monitoring and carbohydrate counting. The controlled access and sharing of diabetic data play a crucial role in benefitting healthcare professionals to take better patient care and researchers to effectively study disease patterns. We addressed the current challenges associated with the privacy, security and controlled sharing of sensitive diabetic data, thereby improving patient care and advancing research. Our research identifies these laying the foundation for the novel solutions DiabeticChain offers. We recognized the current limitations of the traditional client–server and cloud-based approaches such as privacy, security concerns, data fragmentation, authorization, lack of interoperability and data integrity and hence proposed a private permissioned blockchain-based solution for sharing of diabetic data. DiabeticChain, our blockchain-based solution to diabetic data sharing, provides a novel patient-centric

approach, leveraging access control mechanisms and smart contracts written in solidity to facilitate secure and efficient sharing of diabetic data among stakeholders while addressing concerns regarding privacy, security, data fragmentation, authorization and data integrity. We implemented a proof of concept of the architecture proposed using a privately permissioned Ethereum blockchain and analysed the system's performance where the results depicted an average Query throughput of 201.7 tps and an average Transaction throughput was 200.07 tps. The average Query latency was 0.24 seconds and the average Transaction latency was 1.31 seconds proving DiabeticChain to be scalable and efficient. In comparison with the existing blockchain-based healthcare systems, we propose a blockchain-based architecture that uses a one-time access URL generated as per the access policy of the data owner, improving privacy, security and efficiency of diabetic data sharing. The proposed architecture will have a potential impact on patient care and research in the context of diabetes management as well as integration and application in other healthcare domains by enhancing the efficiency of the data sharing process, ensuring patient control over their data, and enabling privacy and data security.
However, DiabeticChain's initial design still faces scalability and interoperability limitations; hence, Chapter 5 introduces enhancements using IPFS, Polygon, and HL7/FHIR to address these issues.

Chapter 5

Enhancing Interoperability and Scalability in DiabeticChain: An Integrative Architecture Using IPFS, Polygon, and HL7/FHIR

This chapter extends DiabeticChain with IPFS for storage, Polygon for scalable execution, HL7/FHIR for semantic interoperability and zk-SNARKs for privacy-preserving consent. It outlines the architecture and implementation and analyses performance and security against declared goals.

5.1 Introduction

Blockchain technology has emerged as a promising solution to security and privacy challenges in healthcare, offering distributed trust, immutability, and auditable record-keeping [144, 145]. Crucial diabetic patient data such as fasting blood glucose (FBG), hemoglobin A1c (HbA1c), continuous glucose monitoring (CGM) readings, blood pressure, medical history, prescriptions, and lab reports can be securely stored and managed on a blockchain ledger [146]. Using this disruptive technology, the routinely collected healthcare data can be securely managed on the blockchain by the various stakeholders like patients, caregivers, physicians, healthcare insurance companies, researchers and pharmaceutical companies [147]. In our earlier work, we introduced DiabeticChain, a patient-centric, permissioned blockchain system that empowered patients to control access to their diabetic data through smart contracts. The system ensured user-centric control, data integrity, and privacy by allowing patients to grant or revoke consent for data access [148]. Although the design of the DiabeticChain system elevated security and privacy standards, it also demonstrated the importance of some additional challenges associated with the large volumes and complexities of healthcare data.

The proliferation of IoT devices, which include implantable CGMs, smart insulin pens, and wearable sensors, generates vast amounts of real-time health data each second [146, 149]. While this health data is vital for the practice of precision medicine and for early interventions, it creates significant burdens on blockchain networks. Increasingly, the transaction volume on blockchain networks increases the latency and lengthens the time required to confirm transactions [147]. Healthcare stakeholders require seamless and nearly real-time access to this data to support timely diagnoses and treatment decisions while maintaining records that are secure, consistent and highly accessible [150]. However, another obstacle to progress remains. Interoperability between diverse EHR systems that utilize disparate data standards, and the fact that most existing blockchain-based healthcare solutions do not provide strong HL7/FHIR-compliant API interfaces, limits

cross-institutional collaboration and supports holistic patient care by creating data silos. Therefore, there exists a pressing need for a scalable, privacy-preserving and interoperable architecture that will support the constant flow of IoT-generated health data, interact with external healthcare systems, and support patient-controlled fine-grained access to their data.

In this expanded research, a completely redesigned architecture that addresses the limitations mentioned above by integrating (i) zk-SNARKs to prove patient consent without exposing sensitive attributes, (ii) hybrid encryption using AES-256-GCM and RSA to encrypt HL7/FHIR-compliant medical records, (iii) decentralized IPFS storage to maintain tamper-resistant record keeping, and (iv) a Polygon-based execution layer to facilitate low-latency and high-throughput smart contract transactions was created. Additionally, we incorporated a systematic performance analysis utilizing Hyperledger Caliper to measure the throughput and latency of the system over 40–50 rounds to obtain statistically robust benchmarking results.

5.1.1 Background and Challenges

DiabeticChain architecture introduced a patient-centric system that implemented blockchain technology to transform the diabetic healthcare management system and gain both efficiency as well as security in the task of sharing the data. The issues and challenges encountered by conventional data sharing system, such as lack of security and privacy, data fragmentation, data integrity and complete patient ownership of related data, are addressed and thus tackled by the system.The original architecture adopted a hybrid model, combining blockchain-based metadata management with an off-chain database to store sensitive health data. This approach reduced blockchain storage overhead while ensuring immutability and verifiability. Cryptographic techniques such as public key cryptography are utilized to minimize the risk of security attacks and access control mechanisms either grant or revoke the consent of patient to access their respective data. Benefits of blockchain technology are leveraged by DiabeticChain system to preserve the privacy of patient's data, maintain the anonymity of the data and ensure that the patient owns his diabetic data fully, instead of the central authority of the data. The proposed system embedded considerable impact on the patient care and is both feasible as well as efficient option to be integrated with the existing architectures of healthcare systems.

Current implementation of DiabeticChain system encounters several challenges such as scalability issues, interoperability concerns and inefficiency in the task of storing diabetic healthcare data. With the proliferation of IoT devices such as continuous glucose monitors (CGMs), smart insulin pens, and wearable sensors, vast amounts of real-time diabetic health data are generated every second. In addition to creating significant data volume stress on the blockchain network, the increased data volume results in greater transaction latency and longer confirmation times unless the scalability of the blockchain network is addressed. Increasing data volumes associated with diabetic-related healthcare may result in an overburdened current healthcare system, resulting in slower transaction processing times. Additionally, the lack of common, standard formats for storing and communicating diabetic-related data, limits the ability to transfer diabetic data from DiabeticChain to external healthcare systems, resulting in additional interoperability issues. Interoperability restricts the quality care of patients in the healthcare sector due to the presence of numerous data standards [151]. Multiple and often incompatible EHR standards create data silos that degrade the quality of patient care and limit coordinated

treatment. Another challenge may arise due to reliance on a centralized off-chain database for storing sensitive patient data. While DiabeticChain was a secure and efficient solution for diabetic data management, there is room for further enhancement to enable the seamless exchange of health data with other healthcare systems, accommodation of increasing data volume, and optimization of the process of data handling.

The intrinsic characteristics of blockchain technology like consensus protocol, transparency and immutability, make it difficult for a large number of simultaneous transactions to take place on a blockchain [152]. As the number of nodes working simultaneously on the blockchain increase, the system should enable a higher transaction throughput with low latency. The performance of the system needs to be evaluated with increasing number of nodes to meet the scalability requirements and integrate the system in a real-world setting. With an increasing complexity of healthcare ecosystems, the different healthcare providers, platforms and entities must seamlessly interact with each other to share information, enhancing patient care. However, the design and development of blockchain systems occurs independently of each other, which makes it difficult for the different blockchain systems to share information with each other [153]. This limitation makes it difficult for blockchain systems to achieve interoperability between the different systems. A hybrid model for storing data provides a level of security for stored data; however, it also creates limitations in terms of data availability, potential single point of failure, efficient data retrieval and true decentralization. Therefore, there is a need to address the inefficiencies in data handling and data storage.

Therefore, the above limitations require an advanced architecture that is able to: (i) manage increasing data volumes with low-latency transaction processing, (ii) support standardized, interoperable data formats, and (iii) eliminate centralized dependency to ensure that the data storage is robust, fault tolerant, and reliable. This research addresses these needs through a dedicated privacy layer using zk-SNARK-based consent verification, attribute-based encryption (ABE) policies for fine-grained access control, decentralized IPFS storage for tamper-resistant record management, and the integration of the Polygon layer-2 to provide high-throughput, low-latency smart contract execution. Overall, these enhancements create a resilient, interoperable, and regulatory compliant environment for the management of diabetic data.

5.1.2 Objectives and Contributions

In the course of this research, the goal is to create and test an improved, privacy-protecting, and scalable blockchain-based architecture for exchanging diabetic information, as prior versions have failed to address their limitations. Based upon the base version of DiabeticChain, this project proposes a system that will include state-of-the-art cryptographic techniques, decentralized data storage, and performance benchmarking to guarantee practical deployment capability.

The main contributions of this work are as follows:

- A new secure architecture based on zk-SNARK-based consent verification, hybrid AES-RSA encryption and IPFS-based storage for managing medical data in a manner that preserves patient privacy.
- Integration of HL7/FHIR-compliant data transformation to allow for semantic interoperability between different electronic health record (EHR) systems.

- Deployment on Polygon to obtain both high transaction rates and low transaction latency, to support very large scale, real-time data exchange.
- Comprehensive performance evaluation using Hyperledger Caliper across 40-50 benchmarking rounds, including throughput and latency metrics.
- Alignment with Health Insurance Portability and Accountability Act (HIPAA)/General Data Protection Regulation (GDPR) by supporting auditable consent revocation, immutable logging, and robust defenses against common blockchain attacks (e.g., 51% attacks, replay attacks).

This enhanced version of DiabeticChain provides a robust, future-proof approach to diabetic healthcare data sharing, combining security, scalability, and interoperability in a single cohesive system.

The rest of this chapter is outlined as follows: Section 5.2 describes the technical background of the concepts used for enhanced DiabeticChain. Section 5.3 discusses the integrative architecture of enhanced DiabeticChain. The implementation of the proposed architecture and the details of its performance evaluation are outlined in Section 5.4. Section 5.5 presents detailed results showing the performance improvements of the proposed enhanced DiabeticChain architecture. Section 5.6 summarizes the chapter.

5.2 Technical Background

This section discusses the background of the concepts utilised in enhanced DiabeticChain, namely blockchain and smart contracts, and the need for interoperable and scalable healthcare blockchain architectures.

5.2.1 Interoperable and Scalable Blockchains

As a number of blockchains are emerging and the technology is maturing rapidly, the necessity of interoperability is being highly discussed [154, 155]. Interoperability is defined as the capability of various entities to collaborate effectively by exchanging and sharing data [156]. Until recently, blockchains did not prioritize the idea of interoperability, resulting in isolated data silos [157].

Over time, Health Level Seven (HL7) has introduced many standards to ease the sharing of medical data. Among these standards, FHIR stands out as a significant breakthrough as it focuses on addressing interoperability through human-readable message content and can be seamlessly integrated with hospital Electronic Medical Record (EMR) systems [158].

Our thesis introduces a dedicated Interoperability Layer with HL7/FHIR-compliant APIs and a FHIR Mapping Engine to transform clinical records into standardized FHIR resources before encryption and storage. This makes sure that semantic and structural compatibility stays with various EHR systems.

Although there is considerable study of blockchain technology, its scalability represents an important barrier to adoption [159]. Performance suffers when the number of nodes and transactions in a blockchain system grows because each node has to reach agreement on the state of the blockchain. For example, current public blockchains such as Ethereum and Bitcoin can process only 7-20 transactions per second, with blocks confirmed by a network of thousands of computers taking minutes or even hours to confirm [160–162].

This thesis addresses those limitations by using a hybrid storage model. In this model, only the pointers to the encrypted records (CIDs) are stored on-chain, along with the encrypted records being stored on IPFS. Also, since zk-SNARK proof allow for privacy-preserving access control checks (i.e., no information is revealed about the contents of the encrypted records), it maintains privacy without sacrificing performance.

5.2.2 Zero-Knowledge Proofs in Privacy-Preserving Systems

Zero-Knowledge Proofs (ZKPs) are cryptographic protocols that enable one party (the prover) to convince another party (the verifier) that a statement is true without revealing any additional information beyond the validity of the statement itself. [163, 164]
ZKPs are widely used in privacy-preserving applications such as identity verification, anonymous credentials, and confidential transactions. Among the different ZKP constructions, zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) have gained popularity due to their efficiency and small proof size. zk-SNARKs allow verification of proofs without interaction between prover and verifier and can be executed on-chain with relatively low computational overhead.

In our work, zk-SNARKs (Groth16) are used to verify that a healthcare provider has valid consent from a patient without disclosing sensitive attributes such as patient identity or consent duration details. This ensures privacy-preserving authorization and prevents replay or forgery attacks while maintaining high performance.

5.2.3 Attribute-Based Encryption for Fine-Grained Access Control

Attribute-Based Encryption (ABE) is a public-key cryptographic paradigm that enables encryption and decryption based on a set of attributes rather than individual identities[165, 166]. In ABE, data owners can define an access policy specifying which attributes are required to decrypt a given ciphertext, thus enabling fine-grained, context-aware access control. There are two primary forms of ABE: Key-Policy ABE (KP-ABE) where policies are embedded in the decryption keys, and Ciphertext-Policy ABE (CP-ABE) where policies are embedded in the ciphertext.

Our thesis employs CP-ABE to encrypt HL7/FHIR-compliant medical records such that only healthcare providers with matching attributes (e.g., role = physician, purpose = treatment, consent = valid) can decrypt the data. This method complies with HIPAA and GDPR requirements by enabling policy updates and applying least-privilege access without re-encrypting all data.

5.3 Integrative DiabeticChain Architecture

5.3.1 Overview

The new enhanced DiabeticChain architecture presents an overall structure for managing diabetic healthcare data, including a comprehensive, privacy enhancing and scalable architecture for diabetic healthcare data management. The enhanced DiabeticChain architecture is designed to overcome the deficiencies in the original design in terms of scalability, interoperability and privacy and includes a rigorous performance analysis of the architecture in order to meet the requirements of deploying the architecture in a real world

setting. In general, the system uses zk-SNARK based methods to verify patient consent, hybrid encryption (AES-256-GCM, RSA and Ciphertext-Policy Attribute-Based Encryption), and decentralized file storage through IPFS to provide confidentiality, integrity and tamper resistance of patient's medical records. In addition, the system includes a FHIR mapping engine to map all health records into HL7/FHIR resources prior to encrypting and storing them, thereby enabling semantic interoperability among different electronic health record systems.

To meet the performance requirements of real time data sharing of healthcare data, the architecture uses Polygon as a Layer-2 execution environment. This enables the architecture to achieve greater transaction throughput and lower confirmation latency than using traditional Layer-1 blockchains. Additionally, a systematic testing and benchmarking methodology using Hyperledger Caliper was used to execute 40–50 independent runs of tests and measure both throughput and latency to obtain statistically significant performance metrics. The system provides a unified solution for compliance with HIPAA and GDPR regulations.

5.3.2 Architectural Layers

This system architecture consists of multiple layers that are separated to support scalability, privacy, interoperability and security. The layers are described below:

5.3.2.1 User Layer

- The primary stakeholders of this layer are Patients and Healthcare Providers, that interact with the system.
- Patients may grant or revoke consent to share their health records with third party organizations. In addition to grant/revoke consent, patients may monitor who has accessed their health records.
- Healthcare providers will request access to patient's health records to assist in treatment, diagnosis or research purposes. Third parties will only have access to the health records if they have received consent from the patient.
- Two main interactions occur at this level; patient grants consent and organizations request access to patient's health records.

Figure 5.1 presents the enhanced DiabeticChain design as a layered architecture, which makes the separation of concerns explicit across user interaction, privacy, security, interoperability, and blockchain enforcement. In particular, it highlights how consent verification and cryptographic enforcement are handled in the privacy/security layers, while HL7/FHIR-based normalization supports semantic interoperability across heterogeneous EHR systems. This layered view is used throughout the chapter to explain where each component operates and how end-to-end auditability is preserved while keeping clinical payloads off-chain.

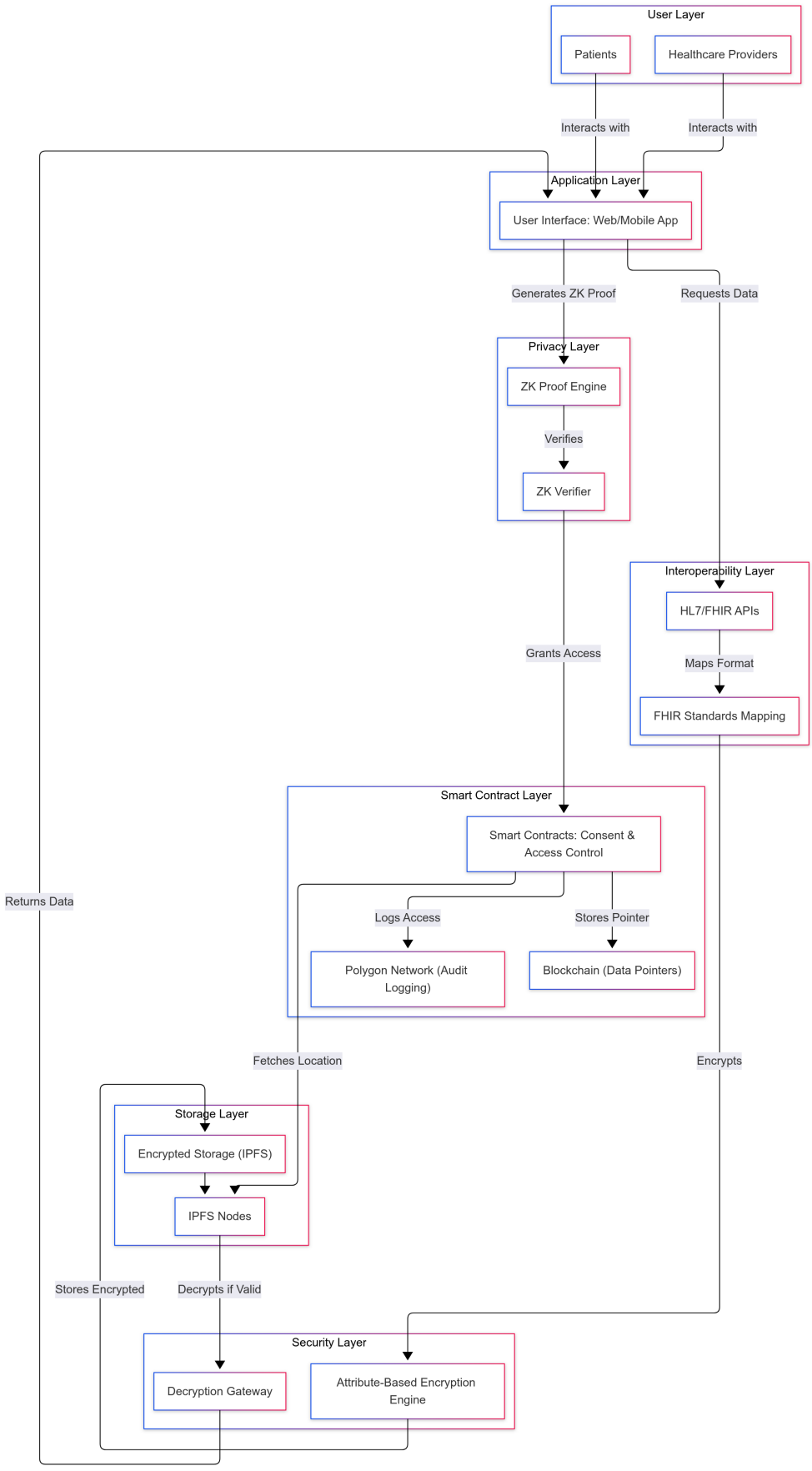


Figure 5.1: Layered Architecture of Enhanced DiabeticChain

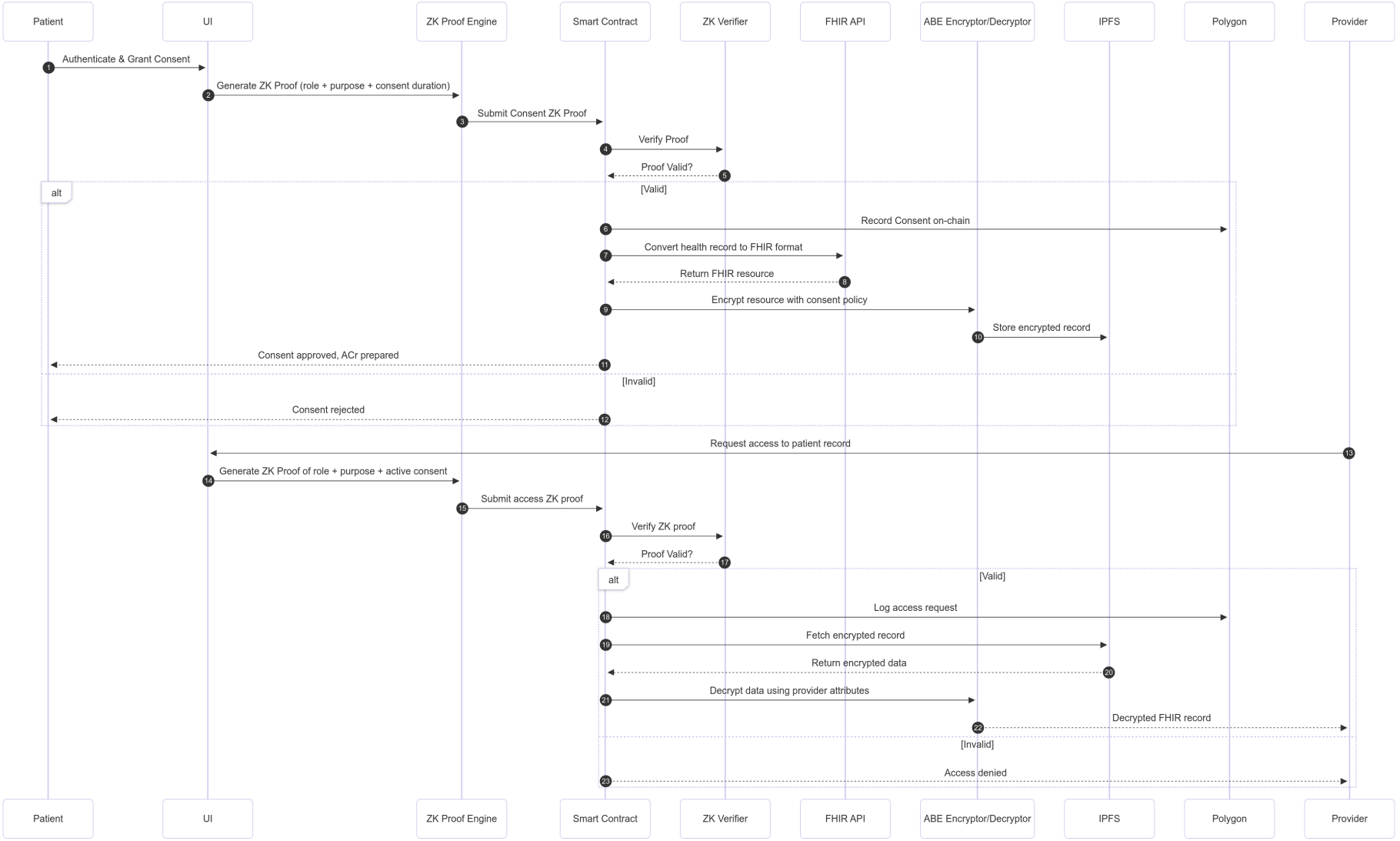


Figure 5.2: High-level Workflow of Consent Registration and Access Enforcement

To make the cross-layer interaction clearer, Figure 5.2 depicts the high-level sequence of consent registration and subsequent access enforcement. The diagram separates the write path (consent creation, record encryption and off-chain storage, and on-chain metadata/consent updates) from the read path, where an access request is validated against the stored policy before any decryption is permitted. This sequence view complements the layered architecture by showing how smart contracts, off-chain storage, and gateway services coordinate to preserve confidentiality and produce an auditable access trail.

5.3.2.2 Application Layer (User Interface (UI): Web/Mobile App)

- Application layer provides a secure web and mobile interface for patients to authenticate their accounts, provide consent to share their health records and initiate data sharing with third parties.
- It serves as the entry point for third parties to request access to patient's health records.
- This layer hides the blockchain complexity and shows a patient-friendly experience, also making sure every request is signed digitally and then shared to the privacy layer.

5.3.2.3 Privacy Layer (ZK Proof Engine / ZK Verifier)

- Privacy layer introduces zero-knowledge proof (zk-SNARK) consent verification for privacy-preserving validation.
- It generates proofs using ZK Proof Engine that encode patient's role, purpose of consent and validity period without revealing any sensitive information.
- The layer verifies the proof created by the ZK Verifier on-chain, prior to recording consent or accessing patient's health records.
- This approach prevents data leaks, supports non-repudiation and responds to reviewer comments concerning the requirement for cryptographic consent validation.

5.3.2.4 Interoperability Layer (HL7/FHIR APIs & Mapping)

- Interoperability Layer provides semantic interoperability by converting health data into HL7/FHIR-compliant resources prior to encryption and storage.
- It provides HL7/FHIR APIs to enable seamless interaction with hospital Electronic Health Record (EHR) systems and other external applications.
- The layer maps heterogeneous data formats into FHIR standards to enable seamless exchange of data between multiple institutions.
- It also ensures cross-system compatibility by enforcing standardization at this level to eliminate data silos.

5.3.2.5 Smart Contract Layer (Consent & Access Control)

- This layer contains all on-chain logic and access decisions:
  - All patient consents after zk-proof validation are recorded, updated, and revoked using ConsentManager contract.
  - Access Control Logic validates whether access requests match the active consent policy before returning data pointers.
  - All consent events and access transactions are immutably logged on Polygon to support regulatory compliance and traceability using Audit Logging.
- It combines policy enforcement with low-cost, high-throughput execution provided by the Polygon network.

5.3.2.6 Storage Layer (Encrypted Storage & IPFS Nodes)

- After standardization of data, it is stored and encrypted in a decentralized way:
  - Encrypted Storage (IPFS): AES-256-GCM is used for encryption of health record, and RSA/CP-ABE encrypts the symmetric key based on the access policy.
  - IPFS Nodes: Used to Store the encrypted content and provide tamper resistance and availability.
- It stores only the content identifier (CID) of the encrypted content on-chain to reduce blockchain storage overhead and improve performance.

5.3.2.7 Security Layer (Decryption Gateway & ABE Engine)

- The security layer applies fine-grained access control during data retrieval:
  - The layer delivers the encrypted AES key to requester via decryption gateway.
  - The Attribute-Based Encryption (ABE) engine makes sure that only valid users with attributes matching the organization's policy can decrypt the data.
- Security layer supports end-to-end confidentiality and covers GDPR's principle of 'least privilege' data access.

5.3.3 High-Level Workflow

There are 2 major phases to the complete, end-to-end workflow of the enhanced DiabeticChain architecture: (i) Consent Registration and (ii) Access Enforcement; this is represented in Figure 5.2 as the interaction between the different layers of the system and the actors within those layers, starting at patient authentication to secure record retrieval.

5.3.3.1 Phase 1: Consent Registration

1. Authentication and Consent Provision: The workflow commences upon the completion of the patient authentication via the user interface (UI) and specification of the consent parameters such as the role of the requester, purpose of use and duration of validity.
2. Zero-Knowledge Proof Generation: The Privacy Layer uses the Zero-Knowledge Proof Engine to create a cryptographic zero-knowledge proof (zk-SNARK), to attest that valid consent was provided without revealing the patient's identity or raw consent data.
3. Verification & Recording On-Chain: The proof is sent to the Smart Contract layer and validated as correct by the ZK verifier. Upon validation, the Consent & Access Control contract creates an immutable consent event on the Polygon network recording the event of consent being given thereby providing transparency and auditability.
4. Standardizing FHIR & Encrypting: The smart contract communicates with the Interoperability layer converting the clinical record to HL7/FHIR-compliant format and passes the resultant resource to the Security layer, then, the ABE Engine encrypts the resource using fine-grain attribute-based policies generated from the consent.
5. Decentralized Storage: The encrypted resource is stored on IPFS and the CID of the resource is recorded on-chain so that future retrieval is possible. The workflow concludes by creating an Access Credential (ACr) bound by policy which the patient may then share with authorized parties.

5.3.3.2 Phase 2: Access Enforcement

1. Initiating Access Requests: An authorized healthcare provider or other authorized entity, initiates an access request through the user interface (UI) and submits a zk-proof demonstrating compliance with the consent policy (role, purpose & validity period).
2. Verifying zk-proofs & Logging Events: Once Submitted, the zk-proof is sent to the smart contract where the zk-verifier verifies the validity. If verified, an immutable log entry is created for the access event on Polygon.
3. Retrieving the Encrypted Health Record: The smart contract retrieves the encrypted FHIR resource from IPFS based upon the CID stored during the initial access request.
4. Decrypting Data Based Upon Attributes: The encrypted AES key is only decrypted by the Decryption Gateway if the attributes associated with the providers meet the requirements outlined in the consent policy. The ABE Engine provides enforcement for the attribute-based decryption process to ensure fine-grained access to the stored health record.
5. Delivery of Health Records: The authorized healthcare provider receives the FHIR-compliant health record in clear-text once the decryption process has successfully completed. In cases where the zk-proof was invalid or the providers attributes do not meet the consent policies, access will be denied.

This high-level workflow ensures that no plain text health records are exposed outside the trust boundary between the patient and their authorized healthcare provider. The workflow also includes cryptographic verification of all critical steps and the creation of an immutable audit trail. The use of zk-SNARKs removes the requirement to reveal sensitive

metadata on-chain, and ABE combined with IPFS ensures only authorized parties can decrypt stored health records, therefore aligning our design with privacy regulations such as HIPAA and GDPR.

5.4 Implementation

This section describes enhanced DiabeticChain's implementation; the design of the smart contracts, zk-SNARK circuits, the encryption modules, and the integration scripts for the new DiabeticChain. All of the source code is open-source and can be located at our GitHub repository [167]. Subsections will cover the development environment (toolchain), the contract logic, the proof creation pipeline, the storage mechanism, the interoperability layer, and the functional verification test results.

5.4.1 Overview

This section will provide a general overview of how the DiabeticChain framework has been enhanced with regard to implementation. To enable independent verification of the results, the system has been implemented as part of a modular and reproducible process. All source code (smart contracts, zk-SNARK circuits, integration scripts) are available on our GitHub repository [167].

To ensure a high-quality, and therefore, reliable development environment, we have used Node.js/JavaScript toolchain, along with Hardhat for deploying and testing smart contracts, and combining cryptographic and storage libraries to securely store and manage decentralized data. Table 5.1, lists the primary tools and versions of each used in this work.

5.4.2 Smart Contracts Implementation

The on-chain implementation of enhanced DiabeticChain is performed via two main smart contracts: ConsentManager and Groth16Verifier. Together these contracts enable the enforcement of consent workflows (based upon cryptography-verifiable consent), the management of revocations, and the immutable recording of all events to the Polygon blockchain.

The most critical operation is the registration of patient consent. As described in Algorithm 5, the process starts by verifying a zk-SNARK proof on-chain to confirm that the consent parameters (role, purpose, validity window) satisfy the deployed policy. Only after successful verification is the consent hash computed and persisted, thereby preventing duplicates or tampering.

Algorithm 5 registerConsent Function

Require: proof, publicSignals, encryptedCID
Ensure: ConsentRegistered event if successful
1: assert Groth16Verifier.verifyProof(proof, publicSignals) == TRUE
2: consentHash ← hash(publicSignals.policyParams)
3: assert consents[consentHash] == NULL
4: consents[consentHash] ← Consent(encryptedCID, publicSignals.expiry, ACTIVE)
5: emit ConsentRegistered(consentHash, encryptedCID, publicSignals.expiry)

Table 5.1: Core Tools and Libraries Used in the Implementation

Tool / Library	Version	Purpose
Node.js	20.x LTS	JavaScript runtime environment
Hardhat	2.26.1	Smart contract compilation, deployment, and testing
@nomicfoundation/hardhat-toolbox	6.1.0	Gas reporting, debugging, and extended test utilities
Circom / SnarkJS	0.7.5	zk-SNARK circuit compilation (Groth16) and proof generation
`ipfs-http-client`	56.0.3	IPFS communication and file upload
`web3.storage`	4.5.5	Pinning and persistent storage of encrypted FHIR files
AES-256-GCM + RSA	N/A	Hybrid encryption for payload and key encapsulation
`dotenv`	17.2.2	Environment variable management for reproducible deployments
`web3`	1.3.0	Interaction with blockchain (Polygon testnet/local fork)

This algorithm guarantees that no consent can be recorded without passing zk-SNARK verification, thereby preventing forged or replayed consents.

Revocation of consent is shown in Algorithm 6, where patient can mark an active consent as revoked at any period of time. Once consent has been revoked, it will reflect on-chain immediately such that any subsequent access requests will fail.

Algorithm 6 revokeConsent Function

Require: consentHash
Ensure: ConsentRevoked event if successful
1: assert consents[consentHash].status == ACTIVE
2: consents[consentHash].status ← REVOKED
3: emit ConsentRevoked(consentHash)

The last major operation is processing access requests. In Algorithm 7, it is illustrated how the healthcare provider's zk-proof is verified, the current consent status is evaluated, and the encrypted record pointer (CID) is returned only when there exists an active consent policy for the access request.

Algorithm 7 requestAccess Function

Require: proof, publicSignals
Ensure: AccessGranted event + CID if successful
1: assert Groth16Verifier.verifyProof(proof, publicSignals) == TRUE
2: consentHash ← hash(publicSignals.policyParams)
3: assert consents[consentHash].status == ACTIVE
4: assert currentTime < consents[consentHash].expiry
5: emit AccessGranted(consentHash, consents[consentHash].cid)
6: **return** consents[consentHash].cid

5.4.3 zk-SNARK Proof Pipeline

The key element of enhanced DiabeticChain's privacy protection is the zero-knowledge proof pipeline, that lets patients and healthcare professionals prove their adherence to consent rules while protecting their private information from being visible on-chain. The proof pipeline is based on the Groth16 proving system and includes four phases: circuit compilation, trusted setup, proof generation and on-chain verification.

Groth16 requires a circuit-specific trusted setup that generates a proving key (pk) and a verification key (vk). In our prototype, this setup is executed once for the consent circuit, the resulting vk is embedded in the `Groth16Verifier` contract to support on-chain verification, while pk is kept off-chain and used by the client application to construct proofs. Since Groth16 security assumes that the setup trapdoor is not retained by any single party, a practical deployment should perform this setup as a multi-party ceremony across the participating healthcare organizations (e.g., hospitals and regulators) to reduce reliance on a single trusted entity. Finally, because pk and vk are tightly bound to the circuit, any circuit update (or suspected compromise) requires regenerating keys and rotating the on-chain vk through a controlled contract upgrade or governance-approved update, while deprecating older key versions to avoid verification ambiguity.

Algorithm 8 describes the off-chain process of creating zk-SNARK proofs (for role, purpose, time validity), to be run in the local client environment to ensure that no private data leaves the client environment.

Algorithm 8 zk-SNARK Proof Generation

Require: circuit.circom, witnessData (policyParams, requesterAttrs)
Ensure: Proof (π) and Public Signals (y)
1: Compile circuit.circom → Rank-1 Constraint System (R1CS) constraints
2: (provingKey, verificationKey) ← snarkjs.setup(R1CS)
3: witness ← calculateWitness(witnessData, R1CS)
4: π ← snarkjs.groth16.prove(provingKey, witness)
5: y ← witness.publicOutputs
6: **return** (π, y) for submission to smart contract

Once the proof is generated, the Groth16Verifier contract performs verification on-chain, as shown in Algorithm 9.

In case the proof is correct, then the contract will continue executing (for example register consent or grant access). If not, the transaction will revert, this prevents unauthorized or false access. With these mechanisms, the security of the system is ensured through cryptographic correctness and makes it impractical for attackers to go around

Algorithm 9 On-Chain Proof Verification

Require: Proof (π), Public Signals (y)
Ensure: Boolean (valid / invalid)
1: Load verificationKey from contract storage
2: result ← pairingCheck(π, verificationKey, y)
3: **if** result == TRUE **then**
4: **return** VALID
5: **else**
6: revert("Invalid Proof")

consent enforcement mechanisms by providing incorrect proofs. Algorithms 8 and 9 together provide a complete and effective zk-SNARK solution that effectively prevents data exposure, produces provable assurances, and effectively combines with the consent management system logic presented.

5.4.4 Encryption and Storage Pipeline

To ensure the entire confidentiality of the medical record for the diabetic patient (end-to-end), enhanced DiabeticChain has introduced a hybrid encryption and distributed storage pipeline. It guarantees that the sensitive FHIR resources are always encrypted regardless of how or where they are stored. The pipeline uses AES-256-GCM for symmetric encryption of the payload and RSA (or attribute-based encryption, ABE) for securely sharing the symmetric key with authorized parties. Encrypted payloads are stored on IPFS, and only their Content Identifiers (CIDs) are recorded on-chain, thus minimizing blockchain storage requirements. Algorithm 10 describes the encryption and upload procedure executed whenever a new patient record is registered.

Algorithm 10 FHIR Resource Encryption & Storage

Require: FHIR_Record `(R)`, `Patient_PublicKey`, `ConsentPolicy`
Ensure: CID of Encrypted Record
1: Generate symmetric key `K_AES` ← `random(256-bit)`
2: Encrypt record: `C` ← `AES-256-GCM-Encrypt(R, K_AES)`
3: Encrypt `K_AES` using `Patient_PublicKey (or ABE policy)`: `K_enc` ← `RSA/ABE-Encrypt(K_AES, ConsentPolicy)`
4: Bundle payload: `P` ← {`C`, `K_enc`, `metadata`}
5: Upload `P` to IPFS: `CID` ← `ipfs.add(P)`
6: **return** `CID` for on-chain storage in `ConsentManager`

This process ensures that even if the IPFS network is publicly accessible, no unauthorized entity can read the patient record without satisfying the decryption policy. Data retrieval is handled by a complementary process, shown in Algorithm 11, which decrypts the record only if the requester's attributes match the consent policy.

Together, Algorithms 10 and 11 guarantee confidentiality, integrity, and fine-grained access control. Only those who meet the authorization policy will be able to decrypt the AES key; every time an attempt to access these resources is made, the event generated by the smart contract will be recorded in such a way as to maintain the ability to audit fully.

Algorithm 11 Secure Retrieval & Decryption

Require: `CID`, `Requester_Attributes`
Ensure: Decrypted FHIR_Record (if authorized)
1: Fetch encrypted payload `P` from IPFS using `CID`
2: Verify access policy: **assert** `ABE-Satisfy(Requester_Attributes)`
3: Decrypt symmetric key: `K_AES` ← `RSA/ABE-Decrypt(P.K_enc)`
4: Decrypt record: `R` ← `AES-256-GCM-Decrypt(P.C, K_AES)`
5: **return** plaintext `FHIR_Record (R)`

5.4.5 Integration with HL7/FHIR

Enhanced DiabeticChain incorporates an Interoperability Layer as a result of its ability to provide semantic interoperability. Semantic interoperability means that each time patient records are shared among different types of healthcare systems they do not lose their meaning nor do they cause structural ambiguity. The Interoperability Layer performs data normalization, resource transformation, and validation using HL7/FHIR APIs, which are used for encryption and storage of patient records. The conversion of the patient's medical record into a compliant FHIR resource (i.e., Patient, Observation, Condition) prior to encryption allows authorized stakeholders to parse and use the data in a standardized format. This helps reduce friction when integrating new electronic health records (EHRs) and facilitates portability of the patient's medical record across hospitals, insurers, and research institutions. Algorithm 12 outlines the transformation and validation steps that occur when the Interoperability Layer processes the patient's medical record.

Algorithm 12 HL7/FHIR Data Normalization

Require: Raw Clinical Data `(D)`
Ensure: Validated FHIR Resource `(F)`
1: Map raw fields in `D` → FHIR schema attributes
2: Construct resource object: `F` ← `new FHIR.ResourceType(mappedData)`
3: Validate schema compliance: **assert** `FHIR.validate(F)` == `TRUE`
4: Attach metadata: `F.meta.policy` ← `ConsentPolicy`
5: **return** standardized FHIR resource `(F)`

Schema validation at this point also prevents the propagation of poorly formatted patient records to the decentralized network since malformed patient records are rejected from being encrypted and stored in the network. In addition, including the patient's consent policy as part of the FHIR resource's meta field allows downstream systems to apply policy-based decision making when they retrieve a patient's medical record. Using APIs to drive the integration of new FHIR profiles and to perform HL7/FHIR standard updates will greatly enhance interoperability and extensibility within the system. Future versions of HL7/FHIR standards can be integrated with minimal modifications to the system.

5.4.6 Functional Verification

Functional verification of this work has been accomplished by a unit testing suite implemented utilizing the Hardhat framework. The unit testing suite verifies the correctness, security, and overall robustness of the smart contract functionality, governing the consent registration, access control, and encrypted data management. Table 5.2 unit testing

Table 5.2: Summary of Unit Test Coverage

Test Scenario	**Expected Outcome**
Valid consent registration	Consent successfully recorded on-chain
Retrieval & payload decryption	Original file integrity preserved
Consent with tampered proof	Rejected (invalid zk proof)
Expired consent	Rejected
Access with mismatched purpose	Access denied
Correct re-encryption and CID update	CID updated and mapped correctly
Duplicate registration (same proof hash)	Rejected
Multiple registrations with identical zk input	Rejected
Full access request with valid proof	Encrypted data retrieved and decrypted

represents a critical component in providing a measure of assurance prior to performing the large-scale performance benchmarking. The testing suite contains nine individual testing scenarios which represent a mix of both valid and adversarial user interactions including; (i) Successful registration of a patient's consent utilizing a valid zk-SNARK proof, (ii) Prevention of registering a duplicate entry for the same patient, (iii) Rejection of attempting to register a duplicate entry for the same patient utilizing a tampered or expired zk-SNARK proof, (iv) Correctly encrypt and decrypt a patients FHIR record, and (v) Proper enforcement of access control based upon a patient's granted consent to use their medical information for a specific intended purpose. The test results, shown in Figure 5.3, indicate that each functional requirement outlined in this thesis has been met with a 100% pass rate and demonstrate the reliability of the on-chain and off-chain logic that make up the proposed architecture of this work.

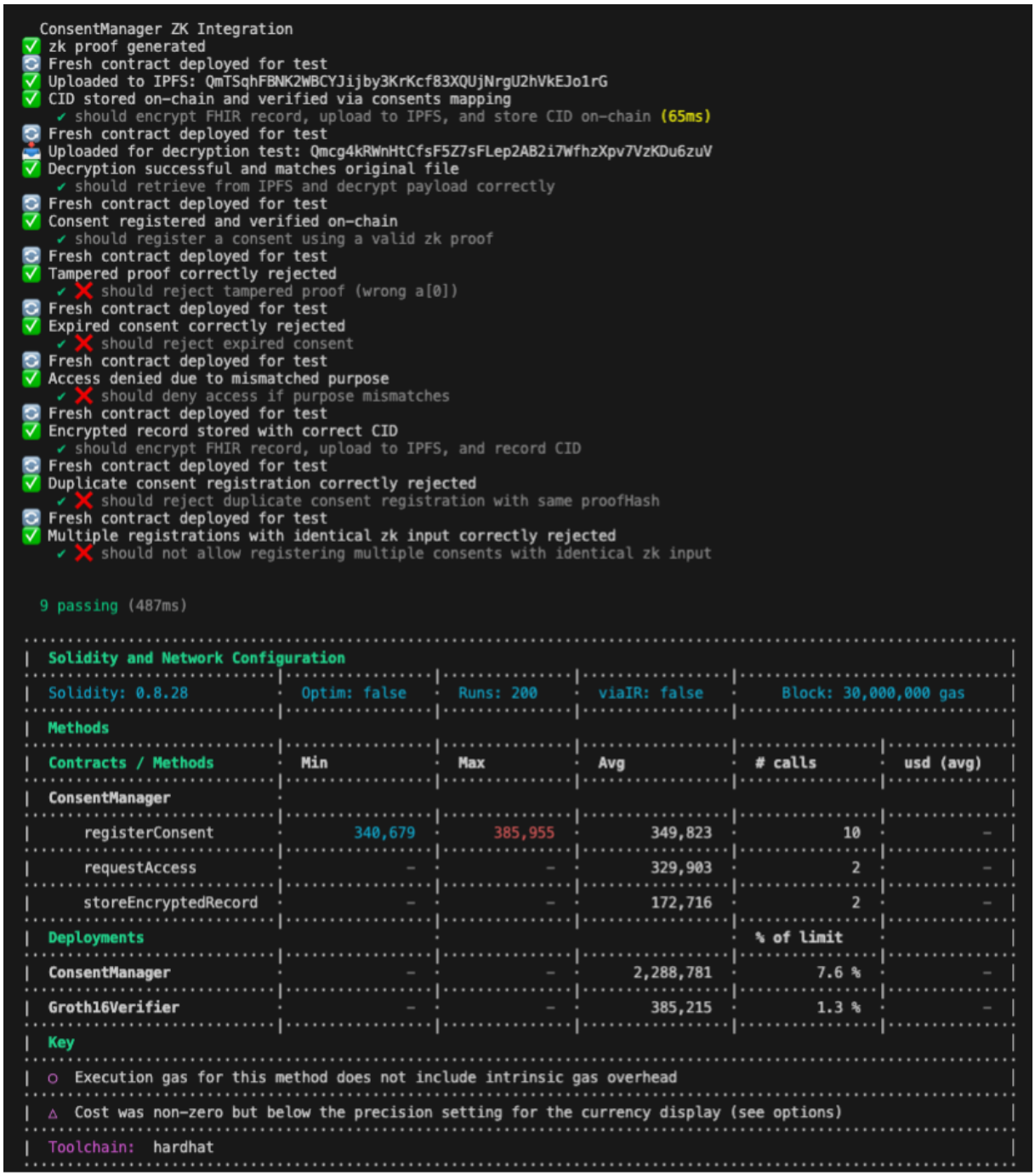


Figure 5.3: Successful Execution of All Consent, Encryption and Access-control Scenarios in the Unit Test Results

5.5 Evaluation, Results, and Discussion

This next part of the chapter describes an experimental assessment of enhanced DiabeticChain. It will describe performance metrics from our experiments, analyze the results of these experiments, and discuss the security implications of the results. We report the throughput, latency and success rate of transactions and queries on a private blockchain using controlled benchmarks of transactions and queries as workload increases. These results will be explained based upon the goals of the system design with respect to scalability limits, resilience characteristics, and the potential security issues that would need to be considered by deployers of such systems in real world healthcare settings.

5.5.1 Experimental Setup

5.5.1.1 Test Environment

An evaluation of an SUT (System Under Test) was made on a completely private, fully-compatible version of an Ethereum network for smart contracts. The SUT was designed to allow both deployment and execution of smart contracts in addition to provide full transactional finality while also allowing a controlled environment for block production. Hyperledger Caliper (version 0.6.0) was used to generate workloads, collect metrics of performance (throughput and latency), and measure the systems resource utilization. Workloads for each experiment were run on 4 servers with 12-core CPUs and 8 GBs of RAM. These four servers had ample processing capabilities to ensure that no bottleneck existed due to the computational capabilities of the servers and therefore allowed us to concentrate solely on how the blockchain would behave under various levels of load.

5.5.1.2 Workload Configuration

The Hyperledger Performance & Scale Working Group define the following blockchain functions [168]:

- **Transactions (state-changing operations):** Any changes to the global blockchain state; for example, registering or revoking a patients' consent.
- **Queries (read-only operations):** Accessing blockchain data that does not change the blockchain's state; for example, obtaining a patients' current consent status.

As stated above, because each transaction requires consensus participation, block inclusion and gas fees to be computed prior to validation, it is expected to have higher latency and lower throughput than queries when subjected to increasing workloads due to the need to reach agreement among all validators and to include the new block in the chain.

Two different workload configurations were therefore created to reflect this difference:

- **Transaction Workload**
  - **Number of Rounds:** 50
  - **Transactions per Round:** 1000
  - **TPS Ramp-Up:** TPS increases from 5 TPS in the first round to 250 TPS in the last round (increases by 5 TPS per round).

  - **Rationale:** The TPS will increase gradually over time to simulate real-world growth and will stress the consensus layer under an increasingly heavy workload to show at what point throughput saturates and latency grows.

- **Query Workload**
  - **Number of Rounds:** 50
  - **Queries per Round:** 1000
  - **TPS Ramp-Up:** TPS increases from 100 TPS in the first round to 5000 TPS in the last round (increases by 100 TPS per round).
  - **Rationale:** Reads are significantly less expensive and can scale much better than writes because reads do not require consensus. A larger TPS range was therefore chosen to measure its performance limit and to demonstrate the expected throughput disparity between state-changing and read-only operations.

This test plan enables us to examine enhanced DiabeticChain's capability to process high write volumes (for example, continuous consent updates) and large-scale read requests (for example, quickly accessing patient consent files), and realistically model potential healthcare scenarios. Additionally, comparing the performance of the two types of transactions provide insight into the system's scalability and help diagnose bottlenecks in consensus processing and state access.

5.5.2 Performance Results and Analysis

- **Transactions Workload**
  The performance analysis of enhanced DiabeticChain in terms of benchmarking the blockchain transaction processing performance, for 50 sequential rounds of operation demonstrated a scalable transaction processing capability when it processed 5 to 250 TPS sequentially. Therefore, the testing resulted in 50,000 write operations performed on a private blockchain as illustrated in Figure 5.4. The performance of DiabeticChain demonstrated almost perfect linear scalability until the point of approximately 240 TPS. After that point, the performance began to plateau and indicate that either the rate of producing blocks, or the capacity of the mempool, had reached saturation. As the actual throughput closely matched the theoretical y=x line up to this point of saturation, it also illustrates that enhanced DiabeticChain will continue to operate reliably without causing significant bottlenecks to process increasingly larger workloads.

  As shown by Figure 5.5, there were three discrete performance ranges observed:

  Optimal Zone (0–100 TPS): A 100% success rate for all transactions with steady-state latency (approximately 8.9 ms) were achieved demonstrating that enhanced DiabeticChain operates reliably in moderate workload conditions.

  Stable Zone (101–240 TPS): Average success rate of 95.5% was maintained while a single instance of failure occurred at round 30 (150 TPS), where 181 transactions failed (18.1% failure rate). However, the system immediately returned to a normal state of operation in subsequent rounds, illustrating its high degree of resiliency to transient stress.

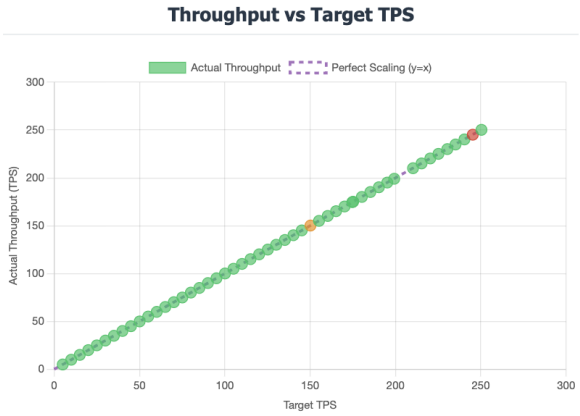


Figure 5.4: Throughput vs. Target TPS for Transactions

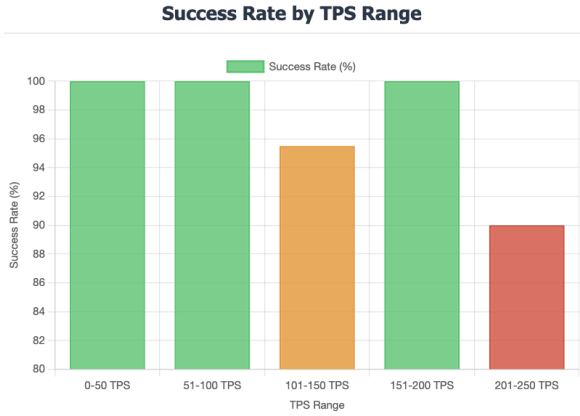


Figure 5.5: Success Rate by TPS Range for Transactions

Critical Zone (241 – 250 TPS): Significant deterioration in performance was observed where the success rate fell to 90%, and 846 transactions failed at round 49 (84.6% failure rate), marking a hard limit to performance.

An interesting observation made from examining Figure 5.6 is the observed counter-intuitive decrease in latency for higher TPS levels. It is hypothesized that this behavior may be due to batching/scheduling optimization techniques being applied under heavy load conditions or because the failed transactions are not included in the latency calculations. This behavior is consistent with previous studies regarding blockchain performance, where network saturation results in a reduction in queuing delays for successful transactions, however this occurs at the cost of increased failure rates.

As shown in Figure 5.7, the majority of failed transactions occur in two instances of stress—rounds 30 and 49—identifying the boundaries of stable to critical performance zones. Additionally, the full recovery of the system after the initial failure event provides evidence that enhanced DiabeticChain can tolerate transient increases in load, provided the increase does not exceed the saturation threshold.

In conclusion, the findings illustrate that the sustainable operating range for enhanced DiabeticChain is approximately 145 TPS for mission-critical use cases, and with proper monitoring and retry mechanisms, it can achieve acceptable performance to approximately 200 TPS. Beyond 240 TPS, the performance severely degrades resulting in un-

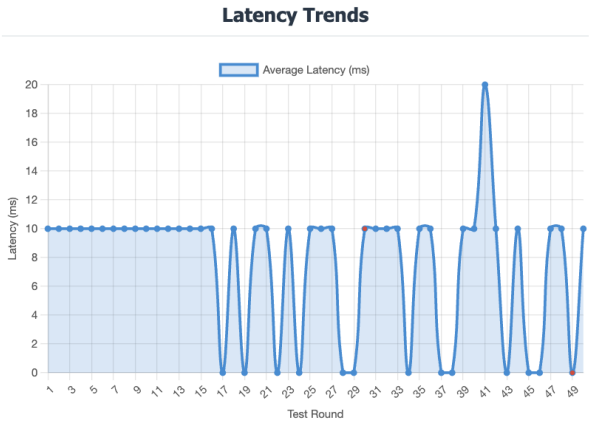


Figure 5.6: Latency Trends for Transactions

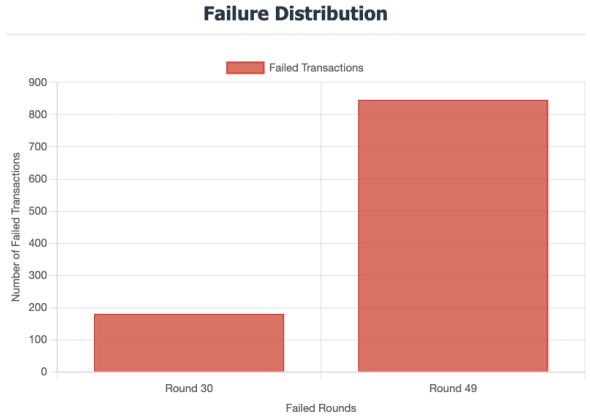


Figure 5.7: Failure Distribution for Transactions

acceptably low transaction reliability. Thus, the results of this study provide valuable guidance for both capacity planning and system tuning for production environments.

- **Query Workload**
  The benchmarking on enhanced DiabeticChain's queries were run for 50 rounds sequentially with increased target rates until reaching 5,000 transactions per second (TPS). The aggregate number of read-operations performed were 50,000 read operations. As seen in Figure 5.8, although there were no failures (100% success rate), the measured throughput leveled-off at approximately 687 TPS. A bottleneck in the Remote Procedure Call (RPC) layer or read-path constraints is likely responsible for this upper limit. Since queries are read-only and do not include blocks, this bottleneck is unlikely to be caused by consensus latencies.

  In contrast to the previous transaction-based workloads where the performance decreased after 240 TPS due to consensus limitations, the query workloads demonstrated perfect success for each of the target loads tested (Figure 5.9). This illustrates the resiliency of the read-paths within the system when subjected to an extreme synthetic load.

  The latency trend (Figure 5.10), like before, indicated zero-latency responses (≈ 0 ms average response time) for all rounds, which demonstrates the ability of the system to provide read access with very little delay even at its maximum throughput (saturation).

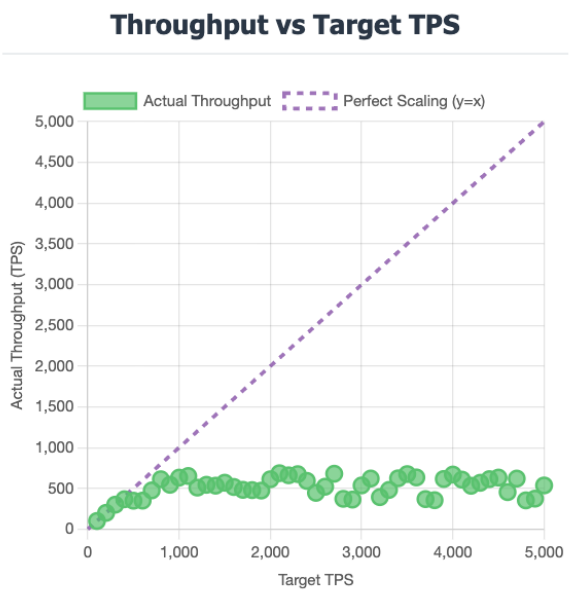


Figure 5.8: Throughput vs. Target TPS for Queries

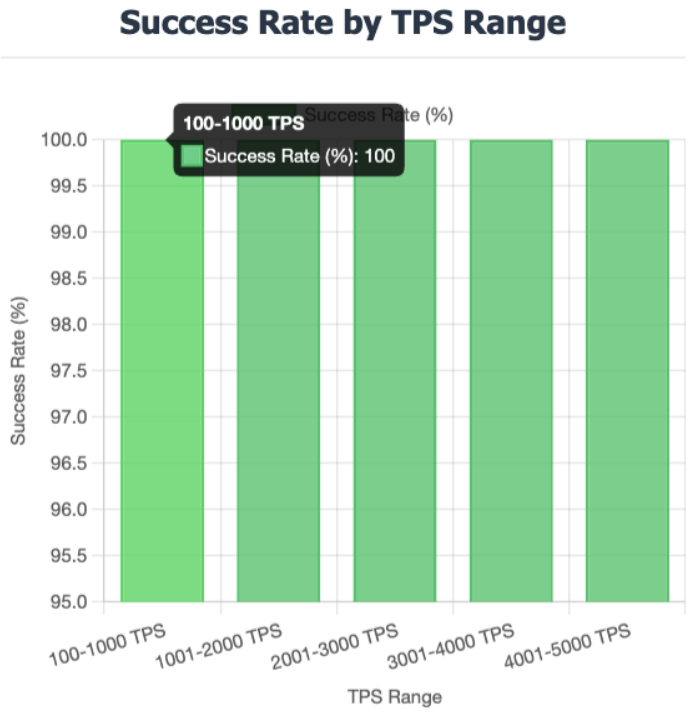


Figure 5.9: Success Rate by TPS Range for Queries

This type of behavior is also important for clinical use cases where responsiveness to patient data retrieval needs to occur regardless of how much the system is utilized.

Finally, Figure 5.11, confirms there were no failed rounds; therefore, query underperformance is solely an issue of throughput efficiency and not reliability.

Taken together, the results of this set of tests clearly indicate that enhanced DiabeticChain's query services are extremely reliable; however, they have a fixed performance ceiling of approximately 687 TPS. When planning for production deployments, a safe

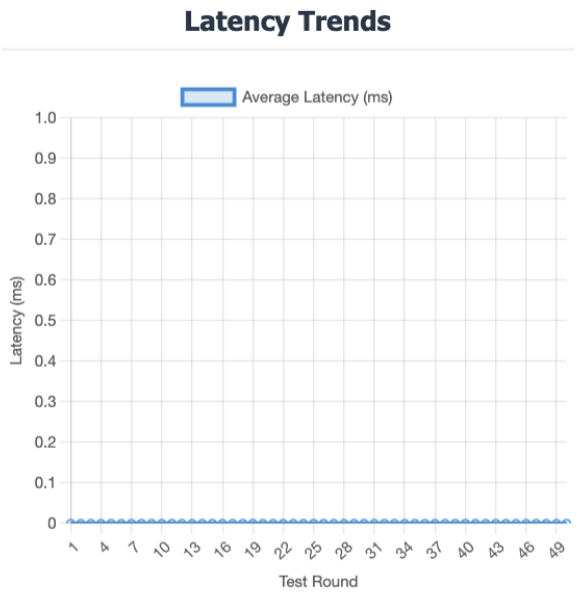


Figure 5.10: Latency Trends for Queries

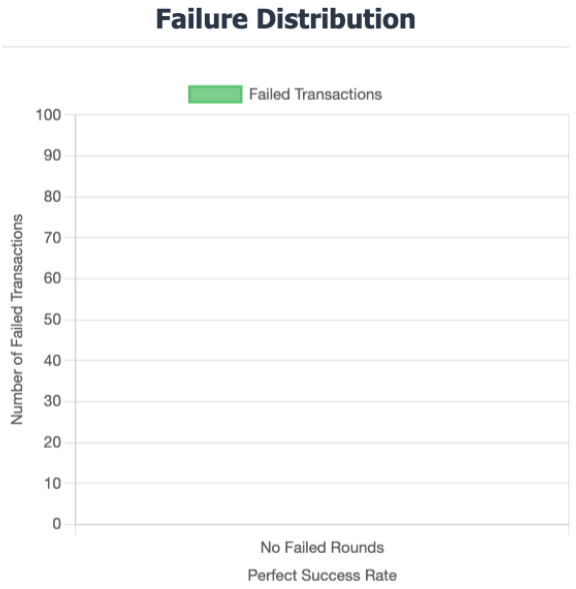


Figure 5.11: Failure Distribution for Queries

operating range would be to target ≤ 600 TPS to allow for some margin between the maximum possible throughput and the maximum allowed throughput. To increase the query service throughput beyond this point, either horizontal scaling of nodes or load-balancing of RPCs will need to be implemented.

5.5.3 Threat Model and Security Analysis

In this section we present an analysis of the threat model for the enhanced DiabeticChain as well as possible attack vectors that can occur at the application and network layers, and the methods available to mitigate these threats and the remaining risk. Figure 5.12 consolidates the adversarial assumptions used in our analysis and visualizes the main

trust boundaries in the system. It distinguishes threats that primarily target the smart-contract/application layer from those that arise at the network/consensus layer, which is important because the available mitigations differ across these two levels.

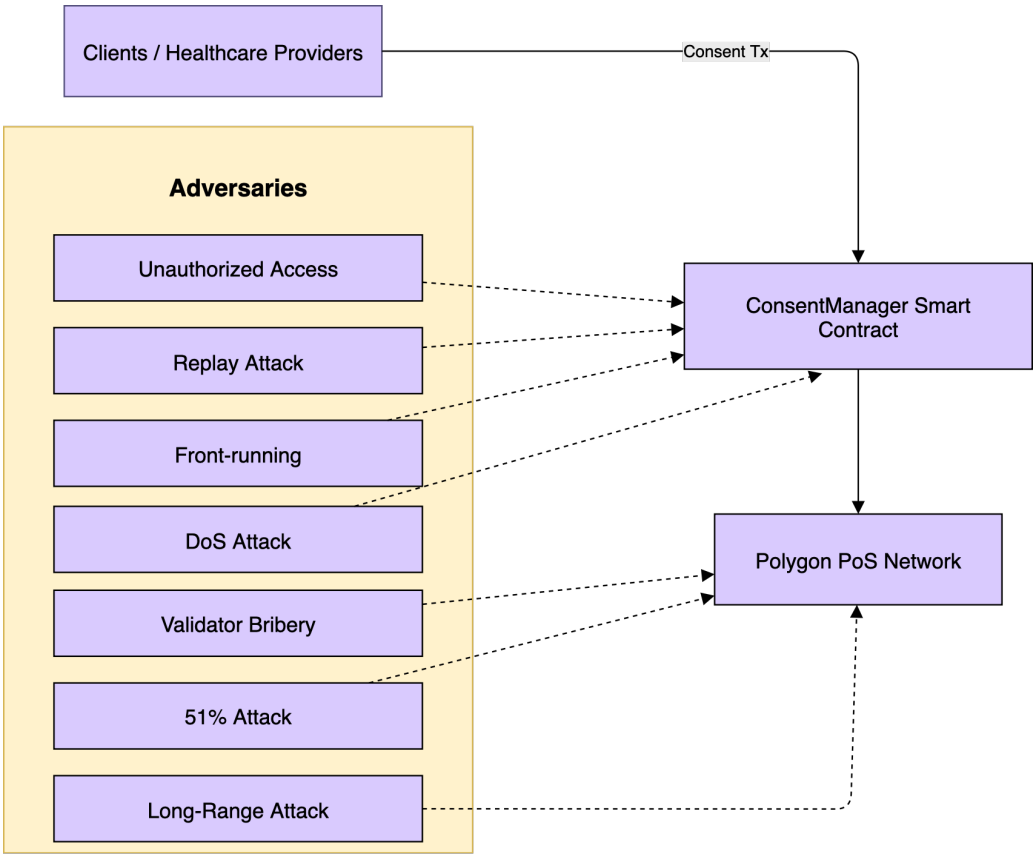


Figure 5.12: Threat Model for Enhanced DiabeticChain

5.5.3.1 Adversary Model

Guided by the trust boundaries in Figure 5.12, we assume the following: (i) patient/provider endpoints protect their private keys, if an endpoint is compromised, the attacker can act with that user's privileges (treated as a residual risk), (ii) off-chain storage (e.g., IPFS/gateways) is untrusted for confidentiality and integrity, so security must not rely on it being honest, and (iii) the permissioned validator set is composed of known entities where an adversary may corrupt a limited fraction of validators but does not sustain majority control of consensus. Under these assumptions, the adversary can observe on-chain transactions and pending mempool activity, submit transactions, call public contract functions, and attempt replay/front-running and DoS at the application layer, while attempting censorship/reordering at the network layer within the stated limits. Accordingly, we separate attacks that primarily target the smart-contract/application layer from those that arise at the network/consensus layer, since the mitigations differ across these two levels. In terms of potential attacks on the smart contract layer, there are four primary types of attacks:

- **Unauthorized Access:** Attempting to bypass role-based permissions or exploit misconfigurations to modify consent records without authorization. This may include privilege escalation by compromised keys/credentials or abusing exposed contract functions, leading to incorrect consent states and loss of patient control.

- **Replay Attacks:** Reusing previously valid transactions (or signed requests) to trigger state changes multiple times and create an invalid system state. Without robust freshness checks (e.g., nonces, deadlines), an attacker may re-apply old consent updates or access approvals, undermining auditability and revocation.
- **Front-running:** Observing pending transactions in the mempool and injecting competing or malicious transactions that are executed first (e.g., by paying higher fees). This can be used to preempt consent updates, reorder access requests, or exploit time-sensitive operations, causing unfair ordering and potential policy violations.
- **Denial-of-Service (DoS):** Flooding the contract or its supporting RPC/middleware with excessive transactions or expensive calls to reduce availability for legitimate users. In a clinical workflow, such congestion can delay consent updates and data-access operations, impacting timeliness even if confidentiality remains intact.
  Potential attacks at the network and consensus layer include the following three types of attacks:
- **Validator Bribery:** Incentivizing validators to reorder, censor, or selectively include transactions for economic gain. Such behavior can delay consent revocations, prioritize adversarial access attempts, or facilitate MEV-style manipulation that weakens fairness and availability guarantees.
- **51% Attack:** Acquiring majority influence over block production (e.g., majority stake/validators) to override the consensus mechanism and reorganize chain history. This can enable rollbacks of recorded consent events or access logs, reducing trust in the immutability and integrity of the audit trail.
- **Long-Range Attacks:** Rewriting historical checkpoint information when older validator keys are compromised and then used over time to forge an alternative chain history. This is particularly relevant to PoS-style settings where past keys may leak, potentially enabling retroactive manipulation of consent timelines unless strong finality/checkpointing is enforced.

5.5.3.2 Mitigations

Enhanced DiabeticChain will use a multi-layered approach to defend against the described threats:

- **End-to-End Encryption:** Patient data is first encrypted by Attribute-Based Encryption (ABE) before it is stored in IPFS; therefore, if someone tries to access the data, the only people who can unlock the data are those who have the correct attributes as healthcare providers; this method prevents an attacker from accessing patient data through unauthorized means, an inside job, or data loss.
- **Zero-Knowledge proofs (ZKPs):** ZKP allows patient consent verification without releasing the patients' sensitive identifiers on-chain. Therefore, ZKP provides protection for patients' identities and helps prevent inference or linkage attacks.
- **Access Control & Replay Protection:** The ConsentManager smart contract uses role-based permissions and transaction nonces to limit access to write operations and limit the potential of malicious parties to reuse previous transactions.
- **Resistance to front-running:** By off-chain signing, encrypting payload, and making atomic on-chain commitments, Enhanced DiabeticChain can limit the visibility of the adversary in the mempool and make it difficult for the adversary to reorder transactions.
- **Denial-of-service (DoS) Resistance:** High gas costs create a financial deterrent for adversaries to attempt DoS attacks by sending spam transactions; additionally, the

middleware level can be used to throttle incoming traffic during a flood attack.
- **Threat Mitigation of Validators:** Polygon PoS has slashing conditions built into its design; therefore, attempting to bribe or collude with validators creates an economic disincentive for them to engage in such activities.
- **Prevention of Long-Range Attacks:** Checkpointing of Polygon to Ethereum L1 creates chain finality and makes rewriting history of the blockchain unfeasible.

5.5.3.3 Residual Risks

Although there are measures in place to mitigate some of these risks, others remain:
- **Economic Threats:** A well funded attacker may attempt to bribe validators or acquire enough staked tokens to censor transactions, or to reorganize the blockchain.
- **Scaling Issues:** While high transaction volumes will slow down the network and increase fees, they will not stop it entirely.
- **Security at the Endpoint:** Users' ability to protect their wallet and private key is still vulnerable to attack as no protocol can completely prevent it from occurring.
- **Availability of Services:** Despite being expensive, a network level DoS (Denial of Service) attack could still temporarily disrupt the network.

5.6 Summary

The contribution of this research is an enhanced version of the DiabeticChain framework. This version is a substantially improved version of the previous DiabeticChain architecture and a further advancement of blockchain technology used for managing data within healthcare applications. An important aspect of this new framework is semantic interoperability with healthcare information standard frameworks, specifically HL7/FHIR standards using a customized mapping/API layer to ensure that all patient records will be standardized to allow portability among disparate and heterogeneous healthcare systems. Furthermore, privacy for patients is protected through the utilization of zk-SNARKs to provide zero-knowledge consent verification without publishing any of the patient's sensitive information on-chain. In addition, confidentiality is ensured through a hybrid encryption system (AES-256-GCM + RSA/ABE), providing end-to-end encryption of all patient data placed on IPFS, even when utilizing untrusted storage nodes.

Evaluation of our enhanced DiabeticChain indicates the ability to support high transactional throughput rates (up to 240 TPS prior to saturation) and very low latency for all query-based operations; supporting use cases for real-time patient data sharing in operational environments. Overall, these findings confirm the enhanced DiabeticChain supports both increased scalability and auditing capabilities, while meeting compliance regulations like GDPR and HIPAA, allowing for legitimate treatment of data (data minimization) and controlled access during the entire life cycle of data.

With the enhanced architecture in place, it is essential to ensure its smart contracts are secure; Chapter 6 introduces TotalSol, a multi-layer static analysis framework for detecting vulnerabilities in the system's smart contracts. Since DiabeticChain's core healthcare logic is encoded in Solidity smart contracts, including user registration (USC), consent-policy storage and validation (APSC), metadata anchoring for off-chain clinical records (MDSC), and access-ticket based authorization (AMSC); any flaw in contract logic or access control can directly weaken patient privacy and consent enforcement. To strengthen assurance prior to deployment, these DiabeticChain contracts are treated as first-class

audit targets and are analyzed using TotalSol as a dedicated static-analysis step in the development workflow. TotalSol runs multiple complementary analyzers over the same contract codebase and consolidates their results into a single, deduplicated report, helping to surface high-impact issues (e.g., reentrancy, arithmetic errors, and access-control weaknesses) that are especially critical in patient-centric healthcare systems. In this way, TotalSol operationalizes the thesis's security objective by providing an explicit, repeatable check on the contracts that implement consent and data-access control.

Chapter 6

TotalSol: A Multi-Layer Static Analysis Method for Vulnerability Detection in Ethereum-based Smart Contracts

The chapter introduces TotalSol, a new multi-layered static-analysis approach to detect vulnerabilities in smart contracts running on the Ethereum blockchain platform. The chapter provides an overview of the methodology employed in this study and the data set used to test it, compares the performance of TotalSol with other existing (baseline) analyzers of smart contracts and discusses the results obtained from the testing phase. This study aims to provide the necessary support for the secure usage of smart contracts in the healthcare domain.

6.1 Introduction

Blockchain is a decentralized system based on Distributed Ledger Technology (DLT) used to store data in a secure manner, while providing transparent access to said data, and immutable storage to said data [169]. Blockchain was originally created for use with digital currency, specifically Bitcoin, however since the creation of Ethereum Smart Contract Technology, blockchain has expanded into a number of areas outside of digital currencies. A smart contract is a self-executing digital agreement which is directly coded onto a blockchain. The smart contract can automatically verify, enforce and enable the provisions contained within the terms of the smart contract without the need for an intermediary. Smart contracts are much more than simply a digital version of an existing agreement; they are designed to take action when predetermined conditions have been met, thus automatically enforcing the terms of the agreement, including a tamper-proof record of all transactions [170]. Once deployed, a smart contract will operate as intended and will automatically execute the previously determined actions such as transferring funds/assets, releasing funds or issuing tokens without requiring human interaction. In addition to minimizing threats, the inherent reliability of smart contracts also creates a trusted environment; smart contracts therefore serve as the backbone of decentralized systems that provide many different types of automated, trusting exchanges between parties [171]. In contrast, the contract platforms such as Ethereum are open in nature for everyone to participate.

Execution of Ethereum and Bitcoin is vulnerable to manipulation attempts by arbitrary adversaries, unlike traditional permissioned networks where the threat is lower. They allow network participants to decide which transactions to accept, how to order transactions, and set the block timestamp, among other things. Contracts based on any

of these sources must be aware of the subtle semantics of the underlying platform and explicitly be protected against manipulation [172]. Unlike classic distributed applications, which can be patched when errors are discovered, contracts are irreversible and immutable. There is no way to repair a broken contract, regardless of its value, without reversing the blockchain. Therefore, it is important to think about the accuracy of the contracts before deployment. In recent years, various techniques have been proposed for analyzing vulnerabilities in smart contracts. All current solutions rely on static analysis and symbolic execution to verify the security and correctness of smart contracts. These are complex and require a lot of time to get analyzed. They have limited applicability as they do not process constructs like loops, have a high false positive rate, do not easily scale to large contracts and do not address all types of known vulnerabilities [173]. The techniques we propose in this thesis will allow complete automated analysis of smart contracts, using static techniques to reduce the number of false positives, and handle the entire syntax of smart contracts. Also, the tool developed will significantly take lesser time to operate. The tool is specially made to handle large amounts of real-world smart contracts.

6.2 Security Challenges in Ethereum Smart Contracts

a) **Reentrancy Attacks**: Reentrancy attack occurs when a smart contract interacts with an external contract that can call back into the original contract before the first function completes. This can lead to multiple withdrawals before the balance is properly updated. A famous instance of this attack was the 2016 DAO hack, which exploited reentrancy vulnerability [174].

b) **Integer Overflow/Underflow**: These errors happen when arithmetic operations go beyond the maximum or minimum value of the data type, causing unpredictable behavior. To prevent this, use of the Safe Math library ensures safe and controlled arithmetic operations. Attackers can use up the gas needed for operations, causing functions to fail and thus making the contract unusable. Contracts may be overloaded with operations that exceed block gas limits, leading to DoS.

c) **Front-Running**: This happens when attackers observe pending transactions and submit their own transactions with higher gas fees, getting their own transactions processed first and manipulating the results. Front-running is common in decentralized exchanges, where attackers see large trades and jump ahead to profit from price fluctuations.

d) **Phishing and Social Engineering**: Attackers use tactics like fake websites, emails, or impersonations to trick users into interacting with malicious contracts or addresses. This can result in an unauthorized transfer of assets.

e) **Unchecked Call Return Values**: When a function makes external calls without verifying if they succeeded, it can cause unexpected problems if the call fails. If a contract sends Ether without checking the return value, it might lose funds if the fallback function of recipient contract fails.

f) **Insecure Randomness**: Since the blockchain data is public, generating random numbers in a smart contract is hard, which can result in making it predictable.

This can undermine the fairness of games, lotteries, or any features that depend on unpredictable outcomes.

6.3 Proposed Methodology

This section outlines the workflow used to design, implement, and evaluate the proposed smart contract vulnerability analyzer. To ensure the evaluation reflects current, real-world practice, the dataset is built from verified smart contracts deployed on the Ethereum mainnet. Many publicly available smart contract datasets are either synthetic/non-mainnet or tied to older Solidity versions. Therefore, a Smart Contract Downloader is adopted [175] which uses the Etherscan API [176] to retrieve contract source code directly from the mainnet. This ensures the collected data corresponds to already deployed contracts. The downloader executes in two phases to obtain the desired dataset: first, an Etherscan account is created and API key(s) are generated to access the API. Next, a CSV file containing smart contract addresses is provided (e.g., from Etherscan's open-sourced contract lists). The addresses in the CSV are processed sequentially, each address is queried via the Etherscan API, which returns the corresponding contract source code. The source code is then downloaded locally in standard JSON format [177].
Figure 6.1 summarizes the end-to-end collection pipeline used in this study, from contract acquisition to the initial screening steps that prepare the dataset for analysis. In particular, it highlights how downloaded contracts are normalized and checked for basic structural and compilation validity before they enter the subsequent parsing and vulnerability-detection stages. This ensures that later results are produced on a consistent corpus of Solidity code rather than on incomplete or incompatible sources.

6.3.1 Parsing Phase

In the parsing phase, the downloaded .json files are parsed by using a json parser and are converted into solidity files (.sol) format. The solidity files are modified to ensure compatibility with the Solidity compiler, such as updating pragma statements and fixing version-related issues that may occur while compilation of the files. The modified solidity files are compiled to check for errors. If some contracts can't be compiled, they are discarded. The final output consists of the Solidity contracts that were successfully downloaded and parsed from the Ethereum mainnet.

6.3.2 TotalSol: A Vulnerability Detector

Several existing tools, such as Aderyn, Slither, Mythril, Oyente and Securify have made strides in this field, each specializing in different aspects of vulnerability detection. However, relying on a single static analyzer often leads to incomplete assessments, as a single tool cannot detect the full spectrum of possible vulnerabilities.
This thesis introduces TotalSol as a full-fledged contract vulnerability evaluation tool utilizing multi-static analyzers in parallel. This methodology allows for an increased number of evaluated vulnerabilities while reducing the number of false negatives. It provides an efficient and accurate way to evaluate the on-chain security of smart contracts [178].

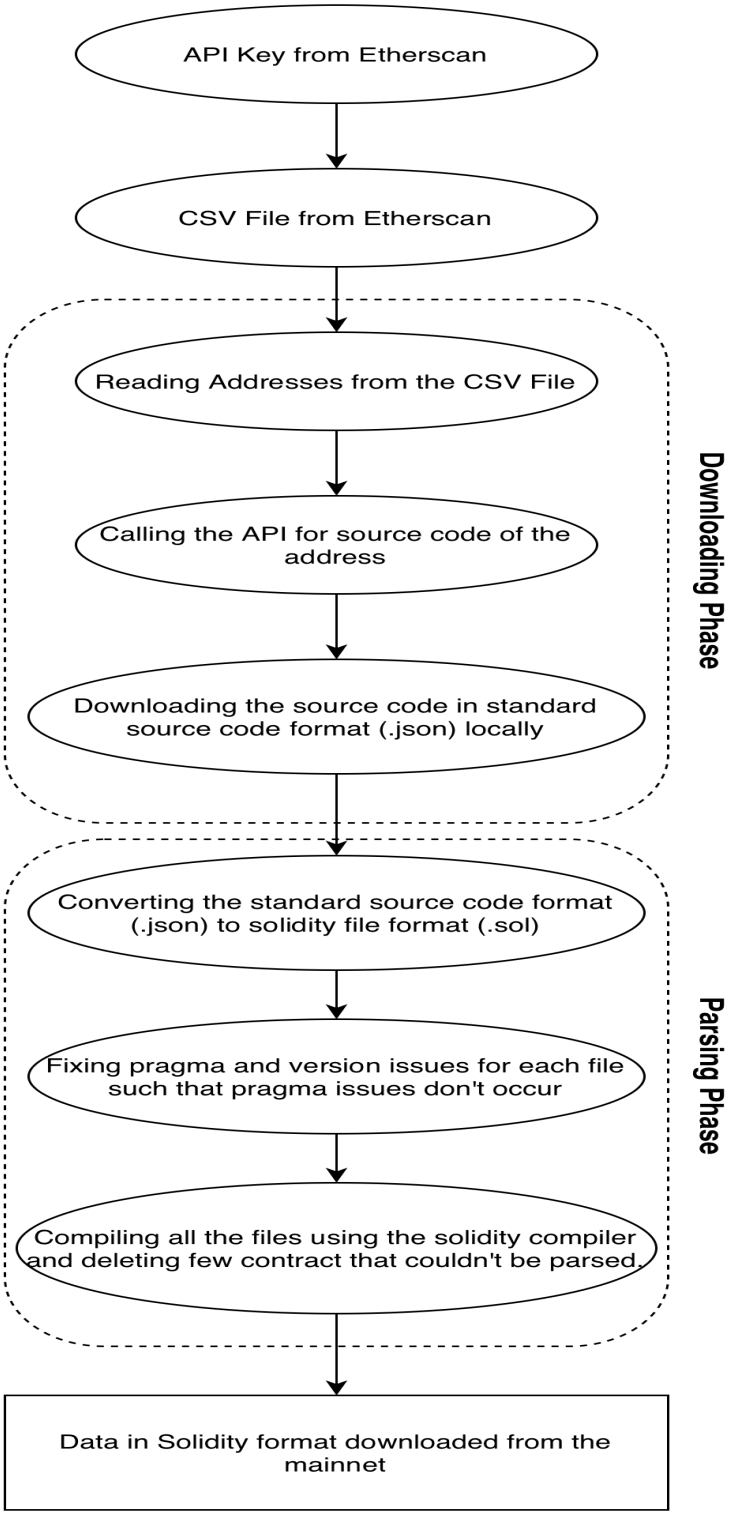


Figure 6.1: Downloading and Parsing Phases of Smart Contract

6.3.3 TotalSol Architecture

Written in C++, TotalSol is a static-analysis tool which looks at combining various smart contract security analyzers into one single framework. Its architecture allows modularity and scalability, allowing the system to pass the same code to different analyzers through a unified input format. This allows seamless operation of different analyzers. This design also allows for easy incorporation of new analyzers as the smart contract and blockchain security area evolves and more vulnerabilities are reported. TotalSol is specially made for analyzing thousands of contracts together [179].

One of the foremost goals of TotalSol is ease of use, allowing a developer to plug in new tools or customize some existing ones without much hassle and without making any substantial changes to the core architecture. This ensures that this tool remains future-proof to stay relevant, given that Solidity would develop and with it, new security issues would be generated. TotalSol uses a very simple yet effective way of finding the bugs from a smart contract. The following flowchart presents an overall construction of TotalSol, showing the clear flow from input files to bug-reporting:

To clarify how TotalSol orchestrates multiple analyzers and produces a single actionable output, Figure 6.2 presents the architecture at a glance. The diagram emphasizes the separation between (i) contract ingestion and preprocessing, (ii) parallel execution of complementary analyzers, and (iii) aggregation of findings into a consolidated report. This view is helpful for interpreting the implementation choices discussed next and for understanding where scalability and extensibility are introduced in the framework.

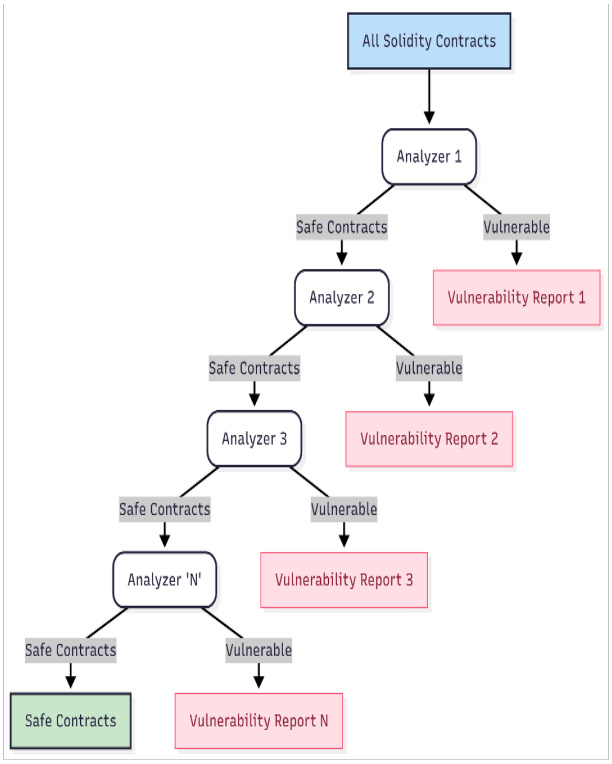


Figure 6.2: TotalSol Architecture

6.3.3.1 Input Methods

TotalSol can use different input formats. The inputs that are currently supported include:

- **Solidity Files (.sol):** The primary input format that allows developers to directly analyze their smart contract source code.
- **JSON Configuration Files:** Smart contracts source code downloaded from the internet is sometimes in standard JSON format. TotalSol can accept and process these files for analysis.
- **Other Custom Formats:** Depending on the specific user or tool, TotalSol supports other input formats.

6.3.3.2 Analyzing Methods and Tools

TotalSol is created on the basis of the three key principles of quality - reliability, accuracy, and speed. These factors were maintained by using only those analyzers that have been both well-maintained and can scale. Each analyzer has its own role within the tool to enable vulnerabilities to be found as quickly as possible and to minimize false negatives and to keep false positives to a minimum. TotalSol includes three primary analyzers - Slither, Aderyn, SmartCheck - and a fourth (Gas Gauge) as an optional tool for optimizing gas usage for smart contracts. In order to maintain the highest level of accuracy while simplifying the process, TotalSol uses a combination algorithm of the outputs of the integrated analyzers it utilizes; this algorithm minimizes the occurrence of false negatives. The system will proceed through a sequence of steps in which each analyzer identifies vulnerabilities and flags the smart contract for subsequent evaluation [180]:
1. **Slither (Analyzer 1)**:
Slither works as the first stage of detection. Swift analysis and low false positives make it ideal. Slither specifies in detecting general Solidity vulnerabilities such as reentrancy, integer overflows, access control issues, and more. The low false positive rate ensures that when other tools analyze the codes, it will be less probable to get a false-positive instance. TotalSol ensures the identification of evident vulnerabilities at the earliest possible stage. When Slither is run in the front, it helps to filter out problematic contracts.
2. **Aderyn (Analyzer 2)**:
Once Slither has finished with its analysis, whenever smart contracts escape, its ambit of vulnerability will fall to being deeper examined by Aderyn. Aderyn shines in the detection of more sophisticated and newer vulnerabilities which might elude tools such as Slither. Traversing through the Abstract Syntax Trees (AST) to catch possible vulnerabilities is Aderyn's foremost strength in providing an elaborated picture of those contracts. The logical vulnerabilities and providing the correct coding practices are spotted quite worryingly by Aderyn by checking the structure, flow and state variables of the contract. Aderyn boasts about the plethora of detectors they host. It is absolutely the case of lots of upgrades and constant maintenance that make Aderyn's detectors most robust and updated, thus much better in terms of searching for new and possibly unnoticed vulnerabilities. Smart contracts which pass through Aderyn unscathed move on to the next stage of analysis [179].
3. **SmartCheck (Analyzer 3)**:
SmartCheck is the third analyzer in the architecture of TotalSol and is designed to analyze coding patterns for vulnerabilities. SmartCheck excels in finding niche vulnerabilities like Denial of Service (DoS) by external contract, Costly Loop, unchecked external calls and timestamp dependence, which other analyzers might miss. Its strength lies in identifying coding patterns that can lead to subtle yet critical vulnerabilities in smart contracts.

SmartCheck analyzes how contracts interact with external contracts, ensuring that no hidden vulnerabilities arise from third-party dependencies. If no critical vulnerabilities are identified by SmartCheck, the contract is marked as safe for deployment. Otherwise, it is flagged and reported as vulnerable [181].

4. **Gas Gauge (Optional Analyzer)**:

Gas Gauge is an optional tool used to analyze gas consumption in smart contracts, and focuses on identifying Out-of Gas Denial of Service (DoS) vulnerabilities and optimizing gas usage efficiency. Developers can choose to skip this step, as Gas Gauge relies heavily on Truffle and can increase the execution time significantly. Gas Gauge lately had a problem in detecting problems on latest smart contract versions. However, for projects where gas optimization and vulnerabilities are a must, Gas Gauge provides valuable insights into how contracts can be optimized.

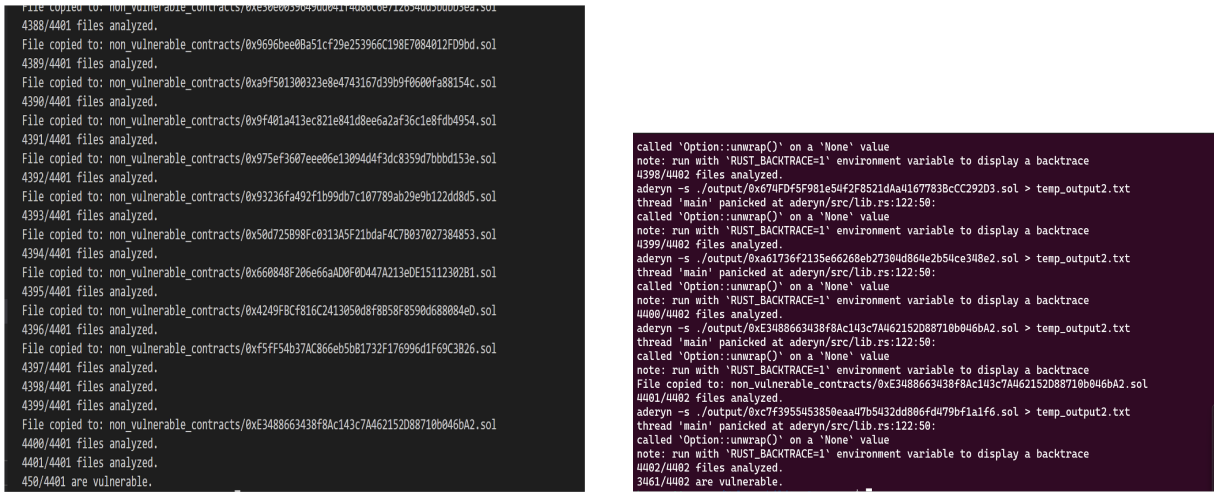
Algorithm 13 Algorithm

1: The Tool takes input of Smart Contract in .sol format.
2: The Tool passes the contract to the analyzer1.
3: If analyzer1 finds vulnerability, then the code is reported vulnerable.
4: Else if analyzer1 does not detect any vulnerability then it is flagged as safe and sent for further analysis.
5: The Contracts that are flagged safe are then analyzed by analyzer2.
6: The analyzer2 flags the Contract in the same way as in step 3, 4
7: The Contract passes n numbers of analyzers before being finally marked as safe.

Algorithm 14 SmartContractAnalysis

```
1: procedure SmartContractAnalysis(directory)
2:     contracts ← load_contracts(directory)
3:     non_vulnerable_contracts ← [ ]
4:     safe_contracts_count ← 0
5:     n ← number of analyzers
6:     for all contract in contracts do
7:         result ← analyzer1(contract)
8:         if result == "vulnerable" then
9:             report_vulnerability(contract)
10:        else
11:            copy_contract(contract, "non_vulnerable_contracts")
12:    for i = 2 to n do
13:        for all contract in contracts do
14:            result ← analyzer_i(contract)
15:            if result == "vulnerable" then
16:                report_vulnerability(contract)
17:                remove(non_vulnerable_contracts, contract)
18:    safe_contracts_count ← len(non_vulnerable_contracts)
19:    vulnerable_contracts_count ← len(contracts) - safe_contracts_count
```

Pseudo Code of Smart Vulnerability Detection in Ethereum Based Smart Contracts



(a) Slither **(b)** Aderyn

Figure 6.3: Baseline Analyzer Outputs Used for Comparison

6.4 Results and Discussion

This section compares TotalSol on a handpicked set of vulnerable Ethereum smart contracts to existing static analyzers and measures its performance against the latter. We merge qualitative data from analyzer outputs with a quantitative overview of detection effectiveness and analysis time (Table 6.1). The comparison is focused on the vulnerability classes that are significant in practice for healthcare DApps (e.g., reentrancy, arithmetic correctness, and access control), where false negatives directly correspond to security risk. The presentation is the same evidence-first as the rest of the thesis: illustrative results, runtime and coverage analysis, shortcomings, and auditor practical implications.

6.4.1 Qualitative Comparison Across Analyzers

Figure 6.3 shows characteristic outputs from Aderyn and Slither on the same instances of the contracts. In a number of instances, the baseline tools identify overlapping but incomplete slices of the problem space; one might detect possible reentrancy while the other identifies gas-costly patterns or suspicious access paths. While informative in isolation as stand-alone reports, these views necessitate manual deduplication and differences reconciliation by the reviewer.

6.4.2 Consolidation via TotalSol

Figure 6.4 shows the combined TotalSol report following multi-analyzer consolidation. The pipeline normalizes tool-to-tool outputs, resolves redundancies, and promotes consensus discoveries to a unified, deduplicated perspective. Operationally, this amalgamation lowered back-and-forth when triaging and simplified the ability to trace a vulnerability from raw pattern to an actionable fix note.

6.4.3 Detection Rate and Analysis Time

Table 6.1 consolidates tool-wise detection rate and mean analysis time on the vulnerable-contract suite. Standalone analyzers are still useful for rapid, targeted scans, but their

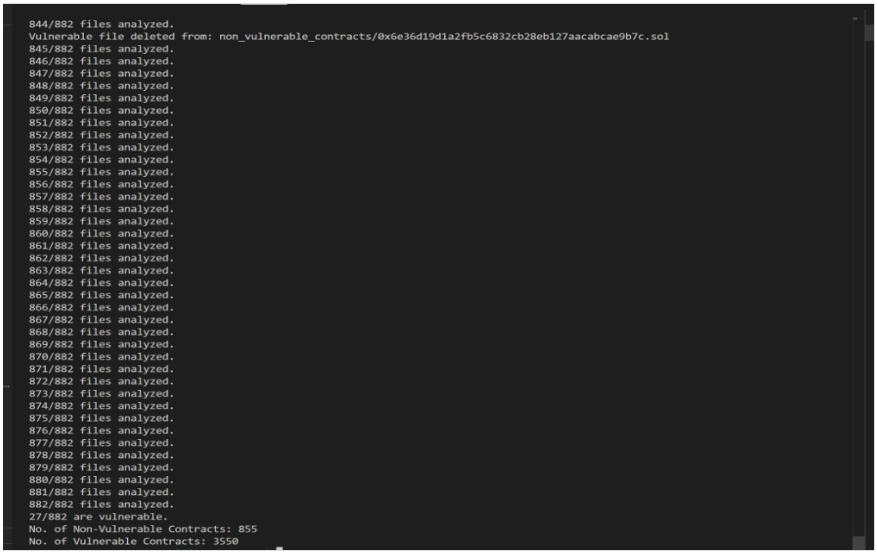


Figure 6.4: TotalSol Consolidated Report After Multi-analyzer Aggregation

Table 6.1: Tool-wise Detection Rate and Time Analysis for Vulnerable Contracts

Tools	Vulnerable Contracts Detected	Total-Contracts Tested	Detection (%)	Time Taken
Slither	3227	4402	73.3%	46 minutes
Aderyn	3461	4402	78.6%	44 minutes
SmartCheck	450	4402	10.2%	60 minutes
Mythril	2779	4402	63.1%	120 minutes
TotalSol	3550	4402	80.7%	90 minutes

detection envelopes vary. Coordinating them via TotalSol fills gaps that emerge if tools are executed separately. The framework sacrifices moderate scheduling overhead in favor of significant reduction in findings missed, which is beneficial in high-stakes audits.

6.4.4 Representative Evidence and Narrative Fit

The results in Figures 6.3 and 6.4 are cropped to show foreground category-level flags and important callouts. Prioritizing the baselines and then the consolidated report captures the chronology of the narrative: per-tool detections, normalization and grouping, and one audit narrative that bridges issues to locations in the code. It facilitates readers to check for themselves how single warnings are reconciled.

6.4.5 Interpretation and Practical Impact

Two observations are most pertinent to auditors. First, coverage: uniting analyzers increases the surface area of vulnerability analyzed per contract and lowers the chance of false negatives on key classes like reentrancy, arithmetic, and access control. Second, efficiency of operation: one, deduplicated report consolidates review by removing collation by hand between tools. Combined, these characteristics enable the chapter's objective of enhancing smart-contract assurance in patient-focused blockchain installations with high confidentiality, integrity, and availability standards.

6.4.6 Limitations and Threats to Validity

Results depend on the selected contract set and versions of specific analyzers. Detection profiles can change with new releases or with contracts that implement upcoming Solidity features. Static analysis—orchestrated or otherwise—cannot emulate all runtime behavior; fuzzing and dynamic testing are still complements to environment-dependent paths. Enhancements are understood mostly as improved first-pass confidence rather than a replacement for dynamic techniques.

6.4.7 Practical Takeaway

TotalSol's multi-layer methodology provides stronger practical confidence than any single analyzer by consolidating detections and triaging. The evidence in Figures 6.3 and 6.4, and Table 6.1 suggests using TotalSol as a first-pass audit step, followed by specialist tools and dynamic testing where needed, is effective.

6.5 Summary

As blockchain technology continues to experience fast changes and as smart contracts continue to play their critical role in decentralized systems, their security guarantees should not be taken lightly. Though currently available tools for smart contract analysis such as Mythril, Slither, Aderyn and SmartCheck are useful in the task of giving insights, but they fall short in terms of detecting a wide range of vulnerabilities with speed and precision. TotalSol is unique among the current analytical tools because it brings together the many static analyzers into one solution. The results are higher efficiencies, fewer false negatives, and greater scalability when analyzing smart contracts. Additionally, through the use of multi-layered analysis, TotalSol fills significant gaps in the current analysis methodologies and offers a much better and scalable assessment of contract security. Although the TotalSol architecture is a highly advanced analysis framework, there is still opportunity for future development in terms of adding new functional abilities and capabilities to TotalSol. A potential area of investigation to enhance TotalSol's functionality in the future is to include dynamic analysis (such as fuzzing) into the TotalSol framework. As mentioned earlier, fuzz testing can simulate how users interact with a system, which may help to detect vulnerabilities that were not identified through static analysis alone. Therefore, the integration of fuzz testing with static analysis will provide a more complete assessment of the smart contract and will increase the total security of blockchain systems. Ultimately, enhancing the functionality of TotalSol with fuzz testing will result in a more versatile and powerful tool for smart contract auditing.

In the final chapter, we synthesize the thesis's findings, discuss limitations and future research directions, and examine the broader societal impact of the proposed patient-centric blockchain solutions.

Chapter 7

Conclusion, Future Scope and Social Impact

This chapter sums up the contributions of this thesis, addresses its limitations and scope for future work, and defines the wider social impact of patient-centric blockchain systems.

7.1 Conclusion

This thesis investigated a variety of blockchain-based architectures that are patient-centric for the management of data in the healthcare environment; these architectures have also been developed with a view toward addressing many of the issues related to security, privacy, and interoperability in addition to scalability for the use of information in healthcare. We found the significant problems associated with traditional centralized healthcare models, including poor data management practices (data loss), inadequate management of the consent process for patients, fragmented medical histories for individual patients, and poor interoperability between different healthcare organizations.

1. This work included an extensive literature review in order to provide an overview of blockchain-based healthcare architectures that currently exist, and to identify current research gaps (in particular in terms of security, privacy, interoperability, and systems for managing patient consent). The literature review provided the background for the subsequent architecture proposals that are presented in this thesis.

2. DiabeticChain is a novel blockchain designed for providing support for diabetic health data management. DiabeticChain uses smart contracts to manage dynamically the consent of individuals with diabetes and utilizes decentralized storage through IPFS. In comparison to the traditional permissionless public blockchain (such as Ethereum), the DiabeticChain blockchain significantly improves the integrity, security, privacy, and control of the individual over their personal health data.

3. Enhanced DiabeticChain combines scalability, interoperability, and privacy through its improved structure. Additions of note are zk-SNARK consent validation, hybrid AES-256-GCM+RSA with fine-grained CP-ABE, IPFS (CIDs on-chain), and a specialized HL7/FHIR layer. Polygon Layer-2 minimizes latency and increases throughput; Caliper testing achieved sustained 200 TPS (plateau 240) with high-speed reads (687 TPS) and near-zero latency, with audit logs and consent revocation supporting HIPAA/GDPR compatibility.

4. TotalSol is a new multi-layer static analysis framework designed for the identification of Ethereum-based smart contract vulnerabilities. TotalSol is of paramount significance in securing blockchains by effectively pinpointing essential vulnerabilities, including reentrancy, integer overflow, underflow, and access control vulnerabilities. Its application ensures the reliability, security, and strength of blockchain applications in the healthcare sector.

Chapter 2 gave the necessary tools and technologies, thus forming a basic technical foundation for architecture development and deployment on blockchain technology. Detailed descriptions of the used development stacks, storage solutions, and testing environments helped to ensure the replicability and transparency of the proposed frameworks.

A systematic literature review in Chapter 3 indicated that there are significant research gaps, including poor or inadequate informed consent processes, lack of strong interoperability, and vulnerability to security breaches. The literature review provided a solid base for the solutions developed in later chapters of this thesis.

Chapter 4 outlines the original DiabeticChain model that can improve patient confidentiality, safety of healthcare data, and enable patients to have better control over their personal health information. Following the design, Chapter 4 demonstrated how the DiabeticChain system (with Ethereum smart contracts) was effectively tested, which led to verification of both the functionality and usability of the system in real-world healthcare environments.

Subsequent innovation in Chapter 5 addresses concerns of interoperability, scalability, and confidentiality. DiabeticChain has been enhanced by the addition of an HL7/FHIR layer, a Polygon Layer-2, zk-SNARK based consent mechanisms, CP-ABE based access controls, and end-to-end encryption with IPFS. Also, Caliper measured an average of approximately 240 TPS and approximately 687 TPS read operations with compliance.

Ultimately, Chapter 6 discusses TotalSol, which provides additional security aspects for blockchain-based healthcare systems. As TotalSol has the capability to identify weaknesses, it can be an important tool for providing a trustworthy system for the deployment of smart contracts in blockchain-based healthcare systems.

To summarize, this research is a significant step towards the development of a blockchain-based healthcare model, that ensures privacy and is secure, scalable, interoperable and centered on the patient. In addition to addressing existing limitations identified in the literature, the proposed solutions in this thesis will provide a suitable platform for future research and represent a revolution toward decentralizing and patient-centric healthcare systems.

7.2 Future Scope

Increased use of digital healthcare delivery systems and patient-centric healthcare will increase the need for additional research related to blockchain-enhanced frameworks. Although this thesis has made a notable contribution to improving data security, data privacy, interoperability, and scalability within blockchain based healthcare delivery systems; there are many other areas that could be researched to enhance and evolve the existing framework.

1. **Integration of Advanced Privacy-Preserving Techniques:** While the current work utilizes zk-SNARK-based consent verification (Groth16), hybrid AES-256-GCM + RSA encryption and fine-grained CP-ABE, there is scope for more

advanced privacy-protecting techniques such as Differential Privacy and Homomorphic Encryption in future work. These have the potential to make patient data entirely secure during processing and sharing, even in distributed systems.

2. **Contextual Access Control and Adaptive Consent:** Still an unexplored area is how consent should be managed dynamically. The focus of the upcoming research could be on developing context-adaptive consent models based on clinical context, emergency conditions, and user preferences. Future works may also include fine-grain revocation of accesses and short-term sharing mechanisms, as for example, notifications in real-time.

3. **Interoperability Beyond Standards:** Although this thesis used HL7/FHIR for the purpose of providing interoperability between applications and services, it is important to continue research on cross-blockchain data-interoperability — i.e., exchanging data between multiple blockchains (as e.g., Ethereum, Hyperledger Fabric, Polygon). If bridges of interoperability are developed, they would allow for larger collaboration at the level of ecosystems by different healthcare service providers and research institutions.

4. **Enhanced Performance Optimization:** DiabeticChain and its enhanced deployment, which includes the use of IPFS and Polygon, have been demonstrated to scale effectively in the test cases; but we expect that a national or worldwide real-world deployment of our system would require improved performance. Potential methods for improving performance include gas optimization, state channels, and sharding, all of which may contribute to a significant increase in throughput and reduction in latency.

5. **AI and IoT Integration for Predictive Healthcare:** Use of decentralized data in AI models will result in enhanced blockchain architectures for predictive diagnosis (for example, predicting complications related to diabetes) and will provide proactive patient care through use of IoT real-time monitoring devices and secured blockchain-based storage of data.

6. **Integration of Legal and Regulatory Framework:** Implementing formal verification tools for smart contracts to enforce compliance with smart contract regulations such as GDPR, HIPAA, and national health data laws, is one right direction to follow. These tools will assist in creating an environment that supports the adoption of technology by healthcare providers and organizations due to the alignment of technology with regulatory body requirements.

7. **Models of Universal Identity and Reputation:** Establishing a globally decentralized identity (DID) system, along with systems of reputation for both patients and healthcare professionals to promote increased levels of trust, accountability, and control regarding who can access their data. We propose integrating DID standards (such as W3C) and verifiable credentials into existing architecture through research.

8. **Quantum-Resilient Cryptography and Readiness:** Plans to implement a hybrid (PQC + Classic) transition to post-quantum cryptography (KEMs and signature schemes), can be done to evaluate the potential performance impacts on

throughput, latency, gas, and verification. This will consider the usage of quantum-resistant proof systems (e.g. STARKs/hash-based) as alternatives to pairing-based SNARKs.

7.3 Social Impact

The findings of this study show how blockchain technology may help improve patient control over, access to, and confidentiality of their patient health information so they have a transparent, accessible, and secure way to experience the healthcare system.

1. Patient-owned medical records will give patients control over who has access to their medical records; which gives them greater autonomy as a patient and will provide continuity of care for chronic conditions such as diabetes.

2. The use of scalable networks (e.g., Polygon) and the minimal amount of central infrastructure needed will allow the system to extend to remote and resource-poor locations and assist in closing the gap in unequal access to secure healthcare.

3. Blockchain's immutable ledger enables actions to be traceable and prevents tampering or fraud from occurring, and the greater level of transparency will contribute to rebuilding the trust that exists between patients, providers, payers, and regulators.

4. Consent-based data sharing will allow for larger scale studies and public health initiatives while protecting the confidentiality of individual patient data; allowing for collaborative and innovative approaches to conducting ethically responsible research.

5. Smart contract access controls will ensure compliance to applicable laws concerning data protection and ethics; promoting the growth of accountable digital health systems that respect patient rights.

6. Improved security practices via integrated verification and authentication will increase the overall cybersecurity position of the healthcare ecosystem as it continues to face growing threats.

This thesis marks the culmination of an extensive analysis of patient-centered, blockchain-based healthcare systems. This thesis identifies the architecture and design elements necessary for creating a scalable, interoperable, and secure digital healthcare ecosystem. The concepts and solutions described above are intended to aid in the development of healthcare systems that protect patient confidentiality, empower patients, and promote trust among all stakeholders involved in the healthcare delivery process.

Appendix A

List of Publications

International Journals

1. D. K. Mishra and P. S. Mehra, "DiabeticChain: A novel blockchain approach for patient-centric diabetic data management," *The Journal of Supercomputing*, 2024. DOI: 10.1007/s11227-024-06589-6.

2. D. K. Mishra and P. S. Mehra, "Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda," *Computers in Industry*, vol. 122, art. 103290, 2020. DOI: 10.1016/j.compind.2020.103290.

3. D. K. Mishra and P. S. Mehra, "Enhancing interoperability and scalability in Diabetic Chain: An integrative architecture using IPFS, Polygon, and HL7/FHIR," *Journal of Computer and Electrical Engineering*, 2025. DOI: 10.1016/j.compeleceng.2025.110835.

International Conferences

1. D. K. Mishra and P. S. Mehra, "Blockchain-based Patient-Centric Healthcare Architecture: A Secure and Efficient Approach for Medical Data Sharing," in *Proc. 14th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, Jul. 6–8, 2023, pp. 1–7. DOI: 10.1109/ICCCNT56998.2023.10307004.

2. D. K. Mishra and P. S. Mehra, "TotalSol: A Multi-Layer Static Analysis Method for Vulnerability Detection in Ethereum Based Smart Contracts," in *2025 Int. Conf. on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*, Greater Noida, India, 2025, pp. 945–950. DOI: 10.1109/IC3ECSBHI63591.2025.10990789.

References

[1] I. Keshta and A. Odeh, "Security and privacy of electronic health records: Concerns and challenges," *Egyptian Informatics Journal*, vol. 22, pp. 177–183, 7 2021.

[2] L. B. Harman, C. A. Flite, and K. Bond, "Electronic health records: Privacy, confidentiality, and security," *Virtual Mentor*, vol. 14, pp. 712–719, 2012.

[3] S. O. Zandieh, K. Yoon-Flannery, G. J. Kuperman, D. J. Langsam, D. Hyman, and R. Kaushal, "Challenges to ehr implementation in electronic- versus paper-based office practices," *Journal of General Internal Medicine*, vol. 23, p. 755, 6 2008.

[4] Lehigh Valley Health Network, "Lehigh valley health network issues cyber incident notification." `https://www.lvhn.org/news/lehigh-valley-health-network-issues-cyber-incident-notification`, June 2023. Official incident notice (BlackCat/ALPHV).

[5] Government of India, Ministry of Health and Family Welfare, "Cyber-attack in aiims (lok sabha unstarred question no. 1837)." `https://sansad.in/getFile/lokabhaquestions/annex/1710/AU1837.pdf?source=pqals`, December 2022. Parliamentary reply: five physical servers affected; most e-Hospital functions restored after two weeks.

[6] N. A. ElSayed, G. Aleppo, V. R. Aroda, R. R. Bannuru, F. M. Brown, and et al., "6. glycemic targets: Standards of care in diabetes—2023," *Diabetes Care*, vol. 46, no. Supplement_1, pp. S97–S110, 2023.

[7] N. A. ElSayed, G. Aleppo, V. R. Aroda, R. R. Bannuru, F. M. Brown, and et al., "7. diabetes technology: Standards of care in diabetes—2023," *Diabetes Care*, vol. 46, no. Supplement_1, pp. S111–S127, 2023.

[8] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *Journal of medical systems*, vol. 40, 10 2016.

[9] M. Ayaz, M. F. Pasha, M. Y. Alzahrani, R. Budiarto, and D. Stiawan, "The fast health interoperability resources (fhir) standard: Systematic literature review of implementations, applications, challenges and opportunities.," *JMIR medical informatics*, vol. 9, p. e21929, 7 2021.

[10] HL7 International, "Fhir overview — hl7 fhir specification (r5)," 2025.

[11] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," *Journal of the American Medical Informatics Association*, vol. 24, no. 6, pp. 1211–1220, 2017.

[12] A. Dubovitskaya *et al.*, "Action-ehr: Patient-centric blockchain-based electronic health record data management for cancer care," *J Med Internet Res*, vol. 22, Aug 2020.

[13] A. N. Gohar, S. A. Abdelmawgoud, and M. S. Farhan, "A patient-centric healthcare framework reference architecture for better semantic interoperability based on blockchain, cloud, and iot," *IEEE Access*, vol. 10, pp. 92137–92157, 2022.

[14] M. George and A. M. Chacko, "Meditrans—patient-centric interoperability through blockchain," *International Journal of Network Management*, vol. 32, pp. 1–18, May 2022.

[15] J. Kaye, E. A. Whitley, D. Lund, M. Morrison, H. Teare, and K. Melham, "Dynamic consent: A patient interface for twenty-first century research networks," *European Journal of Human Genetics*, vol. 23, no. 2, pp. 141–146, 2015.

[16] F. Albalwy, A. Brass, and A. Davies, "A blockchain-based dynamic consent architecture to support clinical genomic data sharing (consentchain): Proof-of-concept study," *JMIR Med Inform*, vol. 9, no. 11, 2021.

[17] A. A. Zaidan, B. B. Zaidan, A. Al-Haiqi, M. L. Kiah, M. Hussain, and M. Abdulnabi, "Evaluation and selection of open-source emr software packages based on integrated ahp and topsis," *Journal of Biomedical Informatics*, vol. 53, pp. 390–404, 2 2015.

[18] K. Abouelmehdi, A. Beni-Hssane, H. Khaloufi, and M. Saadi, "Big data security and privacy in healthcare: A review," *Procedia Computer Science*, vol. 113, pp. 73–80, 1 2017.

[19] D. D. V., "Medical internet of things and big data in healthcare," *hir*, vol. 22, no. 3, pp. 156–163, 2016.

[20] A. I. Stoumpos, F. Kitsios, and M. A. Talias, "Digital transformation in healthcare: Technology acceptance and its applications," *International Journal of Environmental Research and Public Health*, vol. 20, 2 2023.

[21] N. Duong-Trung, H. X. Son, H. T. Le, and T. T. Phan, "On components of a patient-centered healthcare system using smart contract," in *ACM International Conference Proceeding Series*, pp. 31–35, 2020.

[22] L. H. Newman, "Ransomware attacks have entered a 'heinous' new phase." `https://www.wired.com/story/ransomware-tactics-cancer-photos-student-records`, March 2023. Wired magazine article on BlackCat ransomware escalation tactics, incl. LVHN incident.

[23] IBM, "Benefits of blockchain - ibm blockchain," 2023. Accessed: Apr. 25, 2023.

[24] N. R. Pradhan, S. S. Rout, and A. P. Singh, "Blockchain based smart healthcare system for chronic -illness patient monitoring," *3rd International Conference on Energy, Power and Environment: Towards Clean Energy Technologies, ICEPE 2020*, 3 2021.

[25] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *IEEE Access*, vol. 6, pp. 32700–32726, June 2018.

[26] H. Kurdi, S. Alsalamah, A. Alatawi, S. Alfaraj, L. Altoaimy, and S. H. Ahmed, "Healthybroker: A trustworthy blockchain-based multi-cloud broker for patient-centered ehealth services," *Electronics (Switzerland)*, vol. 8, June 2019.

[27] K. Kanagi, C. C. Y. Ku, L. K. Lin, and W. H. Hsieh, "Efficient clinical data sharing framework based on blockchain technology," *Methods Inf Med*, vol. 59, pp. 193–204, Dec. 2020.

[28] T. Fatokun, A. Nag, and S. Sharma, "Towards a blockchain assisted patient owned system for electronic health records," *Electronics (Switzerland)*, vol. 10, pp. 1–14, Mar. 2021.

[29] D. T. Harrell *et al.*, "Technical design and development of a self-sovereign identity management platform for patient-centric health care using blockchain technology," in *Blockchain Healthc Today*, vol. 5, 2022.

[30] S. A. H. Mohsan *et al.*, "Decentralized patient-centric report and medical image management system based on blockchain technology and the inter-planetary file system," *Int J Environ Res Public Health*, vol. 19, Nov. 2022.

[31] "Announcing hyperledger besu – hyperledger foundation," 2019. Accessed: Apr. 21, 2023.

[32] A. S. de Pedro, D. Levi, and L. I. Cuende, "Witnet: A decentralized oracle network protocol," tech. rep., 2017.

[33] M. T. D. Oliveira *et al.*, "Towards a blockchain-based secure electronic medical record for healthcare applications," in *IEEE International Conference on Communications*, May 2019.

[34] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016*, pp. 25–30, Sept. 2016.

[35] A. Khurshid *et al.*, "Designing and testing a blockchain application for patient identity management in healthcare," *JAMIA Open*, vol. 4, July 2021.

[36] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *J Med Syst*, vol. 43, Nov. 2018.

[37] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." Online.

[38] H. Wu, Y. Shang, L. Wang, L. Shi, K. Jiang, and J. Dong, "A patient-centric interoperable framework for health information exchange via blockchain," in *ACM International Conference Proceeding Series*, pp. 76–80, Dec. 2019.

[39] I. Bashir, *Mastering Blockchain*, vol. 161. Packt, 2017. Accessed: Apr. 16, 2023.

[40] G. Lodha, M. Pillai, A. Solanki, S. Sahasrabudhe, and A. Jarali, "Healthcare system using blockchain," in *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, pp. 274–281, May 2021.

[41] B. Zou, F. Nie, L. Xiao, T. Zhang, and C. Zhu, "Meda: Using blockchain for patient-controlled medical data auditing in institutions," *Proceedings - 22nd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGrid 2022*, pp. 297–306, 2022.

[42] S. Cherednichenko, "Designing a blockchain architecture: Types, use cases, and challenges," 2023. Accessed: Apr. 21, 2023.

[43] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.

[44] Ethereum Foundation, "Proof-of-stake (pos)." `https://ethereum.org/developers/docs/consensus-mechanisms/pos/`, 2024. Accessed Sep 24, 2025.

[45] Web3 Foundation / Polkadot Docs, "Proof of stake consensus (npos)." `https://docs.polkadot.com/polkadot-protocol/architecture/polkadot-chain/pos-consensus/`, 2024. Accessed Sep 24, 2025.

[46] Ethereum Foundation, "Proof-of-stake: rewards and penalties." `https://ethereum.org/developers/docs/consensus-mechanisms/pos/rewards-and-penalties/`, 2024. Accessed Sep 24, 2025.

[47] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *CRYPTO 2017*, 2017.

[48] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *CRYPTO 2015*, 2015.

[49] Chia Network, "Proof of space." `https://docs.chia.net/chia-blockchain/consensus/proof-of-space-1.0/`, 2025. Accessed Sep 24, 2025.

[50] Chia Network, "Plotting basics." `https://docs.chia.net/reference-client/plotting/plotting-basics/`, 2025. Accessed Sep 24, 2025.

[51] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OSDI '99*, 1999.

[52] E. Buchman, *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains*. PhD thesis, University of Guelph, 2016.

[53] Hyperledger Foundation, "Using pbft consensus (hyperledger sawtooth)." `https://sawtooth.splinter.dev/docs/1.2/pbft/using-pbft-consensus.html`, 2020. Accessed Sep 24, 2025.

[54] Hyperledger Fabric Docs, "The ordering service (hyperledger fabric) — raft." `https://hyperledger-fabric.readthedocs.io/en/release-2.2/orderer/ordering_service.html`, 2024. Accessed Sep 24, 2025.

[55] Slimcoin Project, "Slimcoin: A peer-to-peer crypto-currency with proof-of-burn." `https://raw.githubusercontent.com/slimcoin-project/slimcoin-project.github.io/master/whitepaperSLM.pdf`, 2014.

[56] Counterparty Docs, "What is xcp? (proof-of-burn)." https://docs.counterparty.io/docs/basics/what-is-xcp/, 2024. Accessed Sep 24, 2025.

[57] Counterparty Docs, "Protocol specification — proofofburn." https://docs.counterparty.io/docs/advanced/protocol/, 2024. Accessed Sep 24, 2025.

[58] Linux Foundation / Hyperledger, "Hyperledger sawtooth (archived): Proof of elapsed time (poet)." https://lf-hyperledger.atlassian.net/wiki/display/sawtooth/, 2024. Accessed Sep 24, 2025.

[59] LF Decentralized Trust Labs, "Proof of elapsed time 2 (poet2)." https://lf-decentralized-trust-labs.github.io/labs/archived/sawtooth-poet2.html, 2023. Accessed Sep 24, 2025.

[60] Intel, "Intel® software guard extensions (intel® sgx)." https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/software-guard-extensions.html, 2025. Accessed Sep 24, 2025.

[61] Ethereum Foundation, "Proof-of-authority (poa)." https://ethereum.org/developers/docs/consensus-mechanisms/poa/, 2024. Accessed Sep 24, 2025.

[62] V. Buterin *et al.*, "Eip-225: Clique proof-of-authority consensus protocol." https://eips.ethereum.org/EIPS/eip-225, 2017. Accessed Sep 24, 2025.

[63] VeChain Foundation, "Vechain whitepaper 3.0." https://www.vechain.org/assets/whitepaper/whitepaper-3-0.pdf, 2023. Accessed Sep 24, 2025.

[64] Gnosis Chain Docs, "The merge (gnosis chain): Transition from poa to pos." https://docs.gnosischain.com/about/specs/consensus/, 2024. Accessed Sep 24, 2025.

[65] Edureka, "Different types of blockchain and why we need them," 2023. Accessed: Apr. 24, 2023.

[66] R. Stevanovic, "Blockchain from scratch: Permissioned and permissionless blockchains," 2023. Accessed: Apr. 24, 2023.

[67] M. T. Oliveira *et al.*, "Towards a performance evaluation of private blockchain frameworks using a realistic workload," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp. 180–187, Feb. 2019.

[68] "Holochain - a framework for distributed applications," 2020. Accessed: Jun. 2020.

[69] "Hyperledger fabric – hyperledger foundation," 2023. Accessed: Apr. 21, 2023.

[70] "Eos.io technical white paper," 2023. Accessed: Apr. 21, 2023.

[71] R. Deviations, "The ripple protocol consensus algorithm," 2023. Accessed: Apr. 21, 2023.

[72] S. Nzuva, "Smart contracts implementation, applications, benefits, and limitations," 2019.

[73] H. Wu, Y. Shang, L. Wang, L. Shi, K. Jiang, and J. Dong, "A patient-centric interoperable framework for health information exchange via blockchain," in *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, pp. 76–80, ACM, Dec. 2019.

[74] M. Vinodhini and M. Prakash, "A patient-centric doctor referral model based on hyperledger chaincode," *CSITSS 2021 - 2021 5th International Conference on Computational Systems and Information Technology for Sustainable Solutions, Proceedings*, 2021.

[75] Y. Zhuang, L. R. Sheets, Y. W. Chen, Z. Y. Shae, J. J. P. Tsai, and C. R. Shyu, "A patient-centric health information exchange framework using blockchain technology," *IEEE J Biomed Health Inform*, vol. 24, no. 8, pp. 2169–2176, 2020.

[76] Solidity Team, *Solidity Documentation (v0.8.31)*. Solidity, 2025. PDF: https://docs.soliditylang.org/_/downloads/en/latest/pdf/.

[77] Truffle Suite, "Truffle documentation (archived)," 2025.

[78] Nomic Foundation, "Getting started with hardhat 3," 2025. Official Hardhat documentation.

[79] Truffle Suite, "Ganache documentation (archived)," 2025.

[80] MetaMask, "Ethereum provider api — metamask wallet api," 2025.

[81] Go Ethereum Team, "Getting started with geth," 2024.

[82] Chainlink Labs, "Chainlink documentation," 2025.

[83] MongoDB Inc., *MongoDB Manual*, 2025.

[84] Protocol Labs, "Content identifiers (cids) — ipfs docs," 2025.

[85] Hyperledger Foundation, "Introduction — hyperledger caliper," 2025.

[86] HL7 International, "Restful api — hl7 fhir specification (r5)," 2025.

[87] Polygon Labs, "Polygon pos — documentation hub," 2025.

[88] Ethereum Foundation, "Zero-knowledge proofs — ethereum.org," 2025.

[89] Electric Coin Company, "Zcash protocol specification," tech. rep., Zcash, 2024.

[90] J. Groth, "On the size of pairing-based non-interactive arguments," Cryptology ePrint Archive 2016/260, International Association for Cryptologic Research (IACR), 2016. EUROCRYPT 2016 version.

[91] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS)*, pp. 89–98, ACM, 2006.

[92] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, pp. 321–334, IEEE, 2007.

[93] V. C. Hu *et al.*, "Overview and considerations of access control based on attribute encryption," nist interagency/internal report (ir) 8450, National Institute of Standards and Technology, 2023. PDF: https://nvlpubs.nist.gov/nistpubs/ir/2023/NIST.IR.8450-upd1.pdf.

[94] M. J. Dworkin, "Recommendation for block cipher modes of operation: Galois/-counter mode (gcm) and gmac," NIST Special Publication 800-38D, National Institute of Standards and Technology, Gaithersburg, MD, November 2007. Official NIST recommendation (includes updates).

[95] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, and Z. Chen, "Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology," *PLoS One*, vol. 15, Dec 2020.

[96] L. Li, Z. Yue, and G. Wu, "Electronic medical record sharing system based on hyperledger fabric and interplanetary file system," in *ACM International Conference Proceeding Series*, pp. 149–154, Feb 2021.

[97] M. Tuler De Oliveira, L. H. A. Reis, Y. Verginadis, D. M. F. Mattos, and S. D. Olabarriaga, "Smartaccess: Attribute-based access control system for medical records based on smart contracts," *IEEE Access*, vol. 10, pp. 117836–117854, 2022.

[98] T. Kenaza, S. Messai, I. Debicha, and M. Sehaki, "A secure and interoperable architecture for blockchain/ipfs assisted electronic health record access control and sharing," 07 2023.

[99] C.-M. Chen, Z. Chen, S. Kumari, M. S. Obaidat, J. J. P. C. Rodrigues, and M. K. Khan, "Blockchain-based mutual authentication protocol for iot-enabled decentralized healthcare environment," *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 25394–25412, 2024.

[100] S. Liu, Z. Wang, and C.-M. Chen, "A blockchain-assisted drug management and authentication scheme in iomt," *IEEE Internet of Things Journal*, 2025. Early Access.

[101] T. Bai, Y. Hu, J. He, H. Fan, and Z. An, "Health-zkidm: A healthcare identity system based on fabric blockchain and zero-knowledge proof," *Sensors*, vol. 22, no. 20, p. 7716, 2022.

[102] G. S. Reen, M. Mohandas, and S. Venkatesan, "Decentralized patient centric e-health record management system using blockchain and ipfs," in *2019 IEEE Conference on Information and Communication Technology, CICT 2019*, Dec. 2019.

[103] H. Kordestani, K. Barkaoui, and W. Zahran, "Hapichain: A blockchain-based framework for patient-centric telemedicine," in *2020 IEEE 8th International Conference on Serious Games and Applications for Health, SeGAH 2020*, Aug. 2020.

[104] V. Mani, P. Manickam, Y. Alotaibi, S. Alghamdi, and O. I. Khalaf, "Hyperledger healthchain: Patient-centric ipfs-based storage of health records," *Electronics (Switzerland)*, vol. 10, pp. 1–14, Dec. 2021.

[105] M. U. CHELLADURAI, D. S. Pandian, and D. K. Ramasamy, "A blockchain based patient centric electronic health record storage and integrity management for e-health systems," *Health Policy Technol*, vol. 10, pp. 1–10, Dec. 2021.

[106] S. Alexaki, G. Alexandris, V. Katos, and E. N. Petroulakis, "Blockchain-based electronic patient records for regulated circular healthcare jurisdictions," in *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*, Oct. 2018.

[107] M. Chen *et al.*, "Blockchain-enabled healthcare system for detection of diabetes," in *Journal of Information Security and Applications*, p. 102771, May 2021.

[108] M. Y. Jabarulla and H. N. Lee, "Blockchain-based distributed patient-centric image management system," *Applied Sciences (Switzerland)*, vol. 11, pp. 1–20, Jan. 2021.

[109] C. Pighini *et al.*, "Syncare: An innovative remote patient monitoring system secured by cryptography and blockchain," in *IFIP Advances in Information and Communication Technology*, pp. 73–89, Springer Science and Business Media Deutschland GmbH, 2022.

[110] A. Taylor, A. Kugler, P. B. Marella, and G. G. Dagher, "Vigilrx: A scalable and interoperable prescription management system using blockchain," *IEEE Access*, vol. 10, pp. 25973–25986, 2022.

[111] A. Mudaliar *et al.*, "Blockchain-based user-centric electronic health record management system," in *2022 IEEE International Conference on Blockchain and Distributed Systems Security, ICBDS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022.

[112] R. A. Abutaleb, S. S. Alqahtany, and T. A. Syed, "Integrity and privacy-aware, patient-centric health record access control framework using a blockchain," *Applied Sciences (Switzerland)*, vol. 13, Jan. 2023.

[113] B. K. Rai, "Pcbehr: patient-controlled blockchain enabled electronic health records for healthcare 4.0," *Health Serv Outcomes Res Methodol*, vol. 23, pp. 80–102, Mar. 2023.

[114] F. A. Reegu, H. Abas, Y. Gulzar, Q. Xin, A. A. Alwan, A. Jabbari, R. G. Sonkamble, and R. A. Dziyauddin, "Blockchain-based framework for interoperable electronic health records for an improved healthcare system," *Sustainability 2023, Vol. 15, Page 6337*, vol. 15, p. 6337, 4 2023.

[115] "Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 152–167, 6 2022.

[116] V. N. K, V. M. T, and M. T, "A blockchain based patient consent management technique for electronic health record sharing," in *2022 IEEE 7th International*

Conference on Recent Advances and Innovations in Engineering (ICRAIE), pp. 215–218, IEEE, 2022.

[117] M. Petticrew and H. Roberts, "Systematic reviews in the social sciences: A practical guide," *Systematic Reviews in the Social Sciences: A Practical Guide*, 2008.

[118] N. K. Dewangan and P. Chandrakar, "Patient-centric token-based healthcare blockchain implementation using secure internet of medical things," *IEEE Trans Comput Soc Syst*, 2022.

[119] S. Jiang, H. Wu, and L. Wang, "Patients-controlled secure and privacy-preserving ehrs sharing scheme based on consortium blockchain," in *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, 2019.

[120] F. Zhao, J. Yu, and B. Yan, "Towards cross-chain access control model for medical data sharing," *Procedia Computer Science*, pp. 330–335, 2022.

[121] M. Kumar and S. Chand, "Medhypchain: A patient-centered interoperability hyperledger-based medical healthcare system: Regulation in covid-19 pandemic," *Journal of Network and Computer Applications*, vol. 179, Apr. 2021.

[122] A. P. Singh *et al.*, "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Trans Industr Inform*, vol. 17, pp. 5779–5789, Aug. 2021.

[123] L. Ismail, H. Materwala, and M. A. B. Khan, "Performance evaluation of a patient-centric blockchain-based healthcare records management framework," in *ACM International Conference Proceeding Series*, pp. 39–50, Association for Computing Machinery, 2020.

[124] B. Toshniwal, P. Podili, R. J. Reddy, and K. Kataoka, "Pacex: Patient-centric emr exchange in healthcare systems using blockchain," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2019*, pp. 954–960, Oct. 2019.

[125] H. Ghayvat, M. Sharma, P. Gope, and P. K. Sharma, "Sharif: Solid pod-based secured healthcare information storage and exchange solution in internet of things," *IEEE Trans Industr Inform*, vol. 18, pp. 5609–5618, Aug. 2022.

[126] H. Kurdi, S. Alsalamah, A. Alatawi, S. Alfaraj, L. Altoaimy, and S. H. Ahmed, "Healthybroker: A trustworthy blockchain-based multi-cloud broker for patient-centered ehealth services," *Electronics (Switzerland)*, vol. 8, June 2019.

[127] K. P. Satamraju and M. Balakrishnan, "A secured healthcare model for sensor data sharing with integrated emotional intelligence," *IEEE Sens J*, vol. 22, pp. 16306–16313, Aug. 2022.

[128] "Cryptographic technology blockchain and its applications," *Lecture Notes in Electrical Engineering*, vol. 560, pp. 14–33, 2019.

[129] S. Monga and D. Singh, "Mrbschain a novel scalable medical records binance smart chain framework enabling a paradigm shift in medical records management," *Sci Rep*, vol. 12, Dec. 2022.

[130] M. A. H. Wadud, T. M. A.-U.-H. Bhuiyan, M. A. Uddin, and M. M. Rahman, "A patient centric agent assisted private blockchain on hyperledger fabric for managing remote patient monitoring," in *Proceedings of 2020 11th International Conference on Electrical and Computer Engineering, ICECE 2020*, pp. 194–197, Institute of Electrical and Electronics Engineers Inc., 2020.

[131] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2 2018.

[132] H. Wang and Y. Song, "Secure cloud-based ehr system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 42, pp. 1–9, 8 2018.

[133] "How can we get blockchains to talk to each other? - ieee spectrum." Accessed: Aug. 30, 2023.

[134] F. F. Ozair, N. Jamshed, A. Sharma, and P. Aggarwal, "Ethical issues in electronic health records: A general overview," *Perspectives in Clinical Research*, vol. 6, p. 73, 2015.

[135] P. Uppamma and S. Bhattacharya, "Diabetic retinopathy detection: A blockchain and african vulture optimization algorithm-based deep learning framework," *Electronics (Switzerland)*, vol. 12, Feb 2023.

[136] B. K. Rai, S. Fatima, and K. Satyarth, "Patient-centric multichain healthcare record," *International Journal of E-Health and Medical Communications*, vol. 13, no. 4, 2022.

[137] S. M. Cleveland and M. Haddara, "Iot for diabetics: A user perspective," pp. 161–172, 2021.

[138] E. W. Karlson, N. T. Boutin, A. G. Hoffnagle, and N. L. Allen, "Building the partners healthcare biobank at partners personalized medicine: Informed consent, return of research results, recruitment lessons and operational considerations," *Journal of Personalized Medicine 2016*, vol. 6, no. 1, p. 2, 2016.

[139] A. Hevner and S. Chatterjee, "Design science research in information systems," pp. 9–22, 2010.

[140] Synopsys, "What is blockchain and how does it work?," 2023. Accessed: Apr. 15, 2023.

[141] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "A cooperative architecture of data offloading and sharing for smart healthcare with blockchain," *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2021*, 5 2021.

[142] P. Nanda, A. R. Cavalli, D. L. Fekete, and A. Kiss, "Toward building smart contract-based higher education systems using zero-knowledge ethereum virtual machine," *Electronics 2023, Vol. 12, Page 664*, vol. 12, p. 664, 1 2023.

[143] W. Li, C. Meese, H. Guo, and M. Nejad, "Blockchain-enabled identity verification for safe ridesharing leveraging zero-knowledge proof," *2020 3rd International Conference on Hot Information-Centric Networking, HotICN 2020*, pp. 18–24, 12 2020.

[144] K. P. Satamraju and M. Balakrishnan, "A secured healthcare model for sensor data sharing with integrated emotional intelligence," *IEEE Sens J*, vol. 22, no. 16, pp. 16306–16313, 2022.

[145] H. S. et al., "Blockchain technology in healthcare: A systematic review," *PLoS One*, vol. 17, no. 4, 2022.

[146] T. Robbins, S. N. L. C. Keung, S. Sankar, H. Randeva, and T. N. Arvanitis, "Diabetes and the direct secondary use of electronic health records: Using routinely collected and stored data to drive research and understanding," *Digit Health*, vol. 4, p. 205520761880465, 2018.

[147] M. Y. Jabarulla and H. N. Lee, "A blockchain and artificial intelligence-based, patient-centric healthcare system for combating the covid-19 pandemic: Opportunities and applications," *Healthcare (Switzerland)*, vol. 9, no. 8, 2021.

[148] D. K. Mishra and P. S. Mehra, "Diabeticchain: a novel blockchain approach for patient-centric diabetic data management," *The Journal of Supercomputing*, vol. 81, p. 166, 2025. Accepted: 7 Oct 2024.

[149] S. M. Cleveland and M. Haddara, "Iot for diabetics: A user perspective," in *XX (Eds.), Smart Health*, pp. 161–172, 2021.

[150] G. Subramanian and A. S. Thampy, "Implementation of blockchain consortium to prioritize diabetes patients' healthcare in pandemic situations," *IEEE Access*, vol. 9, pp. 162459–162475, 2021.

[151] Y. Singh, M. A. Jabbar, S. K. Shandilya, O. Vovk, and Y. Hnatiuk, "Exploring applications of blockchain in healthcare: road map and future directions," *Front Public Health*, vol. 11, 2023.

[152] Web3author, "Blockchain scalability solutions." `https://medium.com/@web3author/unblocking-the-blockchain-bottleneck-scalability-solutions`. Accessed: Jan. 25, 2024.

[153] "How can we get blockchains to talk to each other? - ieee spectrum." `https://spectrum-ieee-org.cdn.ampproject.org/c/s/spectrum.ieee.org/amp/blockchain-interoperability-2655912591`. Accessed: Sep. 08, 2023.

[154] R. Belchior, J. Süßenguth, Q. Feng, T. Hardjono, A. Vasconcelos, and M. Correia, "A brief history of blockchain interoperability," Oct 2023.

[155] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," in *ACM Computing Surveys (CSUR)*, vol. 54, Oct 2021.

[156] P. Wegner, "Interoperability," in *ACM Computing Surveys (CSUR)*, vol. 28, pp. 285–287, Mar 1996.

[157] H. Jin, X. Dai, and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *Proc Int Conf Distrib Comput Syst*, vol. 2018-July, pp. 1203–1211, Jul 2018.

[158] M. George and A. M. Chacko, "A patient-centric interoperable, quorum-based healthcare system for sharing clinical data," in *2022 International Conference for Advancement in Technology, ICONAT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022.

[159] D. Yang, C. Long, H. Xu, and S. Peng, "A review on scalability of blockchain," *ACM International Conference Proceeding Series*, pp. 1–6, Mar 2020.

[160] K. Cong, Z. Ren, and J. Pouwelse, "A blockchain consensus protocol with horizontal scalability," in *2018 IFIP Networking Conference IFIP Networking and Workshops, IFIP Networking 2018 - Proceedings*, pp. 424–432, Jul 2018.

[161] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: a survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[162] A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor, "Blockchain and scalability," in *Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security Companion, QRS-C 2018*, pp. 122–128, Aug 2018.

[163] C. P. Sah, M. Kaur, and G. Singh, "Efficiency of zero-knowledge proofs: A thorough review and analysis," in *2024 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA)*, (Bangalore, India), pp. 1–7, IEEE, 2024. Accessed: 13 Sep 2025.

[164] R. Lavin, X. Liu, H. Mohanty, L. Norman, G. Zaarour, and B. Krishnamachari, "A survey on the applications of zero-knowledge proofs," *arXiv preprint*, 2024.

[165] E. Meamari, H. Guo, C. Shen, and R. Zhang, "Data user-based attribute-based encryption," *arXiv preprint*, 2020.

[166] T. Prantl, T. Zeck, and S. Kounev, "Towards a cryptography encyclopedia: A survey on attribute-based encryption," *Journal of Surveillance, Security and Safety*, vol. 4, pp. 129–154, 2023.

[167] dkm0714, "Enhanced-diabetic-chain: Source code repository." `https://github.com/dkm0714/enhanced-diabetic-chain`, 2025. Accessed: Sep. 14, 2025.

[168] Hyperledger Performance and Scale Working Group, "Dlt metrics whitepaper (v1.01)." `https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/Hyperledger/Printables/HL_Whitepaper_Metrics_PDF_V1.01.pdf`. Accessed: 13 Sep 2025.

[169] S. Akca, A. Rajan, and C. Peng, "Solanalyser: A framework for analysing and testing smart contracts," in *2019 Asia-Pacific Software Engineering Conference (APSEC)*, pp. 482–489, 2019.

[170] L. Albshaier, A. Budokhi, and A. Aljughaiman, "A review of security issues when integrating IoT with cloud computing and blockchain," *IEEE Access*, vol. 12, pp. 109560–109595, Aug. 2024.

[171] Z. A. Khan and A. S. Namin, "Involuntary transfer: A vulnerability pattern in smart contracts," *IEEE Access*, vol. 12, pp. 62459–62479, Jan. 2024.

[172] L. S. H. Colin, P. M. Mohan, J. Pan, and P. L. K. Keong, "An integrated smart contract vulnerability detection tool using multi-layer perceptron on real-time solidity smart contracts," *IEEE Access*, vol. 12, pp. 23549–23567, Jan. 2024.

[173] D. Han, Q. Li, L. Zhang, and T. Xu, "A smart contract vulnerability detection model based on graph neural networks," in *2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, pp. 834–837, 2022.

[174] Z. A. Khan and A. S. Namin, "A survey of vulnerability detection techniques by smart contract tools," *IEEE Access*, vol. 12, pp. 70870–70910, May 2024.

[175] X. Yu, H. Zhao, B. Hou, Z. Ying, and B. Wu, "Deescvhunter: A deep learning-based framework for smart contract vulnerability detection," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2021.

[176] N. Li, Y. Liu, L. Li, and Y. Wang, "Smart contract vulnerability detection based on deep and cross network," in *2022 3rd International Conference on Computer Vision, Image and Deep Learning and International Conference on Computer Engineering and Applications (CVIDL and ICCEA)*, pp. 533–536, 2022.

[177] X. Li, J. Liu, X. Chen, and Q. Zhang, "A symbolic execution-based approach for smart contract vulnerability detection," in *2023 IEEE 6th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pp. 468–472, 2023.

[178] X. Yi, J. Wu, G. Li, A. K. Bashir, J. Li, and A. A. Alzubi, "Recurrent semantic learning-driven fast binary vulnerability detection in healthcare cyber physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 2537–2550, 2023.

[179] N. Songsom, W. Werapun, J. Suaboot, and N. Rattanavipanon, "The SWC-based security analysis tool for smart contract vulnerability detection," in *6th International Conference on Information Technology (InCIT 2022)*, pp. 74–77, 2022.

[180] Z. Yang and W. Zhu, "Improvement and optimization of vulnerability detection methods for Ethereum smart contracts," *IEEE Access*, vol. 11, pp. 78207–78223, July 2023.

[181] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "Smartcheck: Static analysis of Ethereum smart contracts," in *Proceedings of the International Conference on Software Engineering (ICSE)*, pp. 9–16, 2018.

Author Biography

Mr. Deepak Kumar Mishra is currently working as an Assistant Professor at the College of Vocational Studies, University of Delhi. He received his M.Tech in Computer Science and Engineering from Babasaheb Bhimrao Ambedkar University (Central University), Lucknow. He completed his Bachelor of Technology in Information Technology from Gautam Buddh Technical University, Lucknow . His areas of interest include Blockchain Technology, Artificial Intelligence, Database Systems, Compiler Design and Network Security.