applied sciences

MDPI

Article

Blockchain for the eHealth Sector—A Survey and Implementation

Alessandro Vizzarri * and Franco Mazzenga

Department of Enterprise Engineering, University of Rome Tor Vergata, 00133 Rome, Italy; mazzenga@ing.uniroma2.it

* Correspondence: alessandro.vizzarri@uniroma2.it

Abstract

Blockchain is one important building blocks of the Internet of the future, called Web3. The Blockchain technology supports a wide range of applications, spanning from Smart Cities and automotive industries, from agriculture to energy. The healthcare sector, in particular, has experienced a profound impact from blockchain-based technologies, paving the way for the development of true digital healthcare systems. By enabling secure and immutable data storage, and facilitating the sharing of this information among all nodes possessing a local copy of the distributed ledger, blockchain plays a vital role in the analysis of healthcare data. This paper provides a comprehensive survey of the main blockchain platforms utilized in the digital healthcare, integrated with a comparative analysis. In addition, the implementation of Innovative permissioned Blockchain for eHealth (IBEH) is presented and discussed in detail. IBEH addresses key challenges in digital health data management, including secure and controlled access to sensitive health information, ensuring data integrity and traceability, and secure sharing between different healthcare institutions and organizations. This is made possible by decoupling the application and blockchain layers and by a flexible, customizable, and easily deployable infrastructure. IBEH integrates the application-oriented and embedded layer with that of a blockchain network built with the MultiChain platform, which uses smart contracts with permissions, REST APIs, and RPC calls. The main features and its associated smart contracts within the healthcare domain are discussed. Finally, the analysis of performance is provided.

Keywords: blockchain; eHealth; smart contract; ethereum; performance analysis; correlation models

Check for updates

Academic Editor: Yujue Wang

Received: 25 June 2026
Revised: 14 August 2026
Accepted: 17 August 2026
Published: 25 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Blockchain technology is regarded as an important building block of the innovative model of the Internet of the future, named Web3 [1]. It represents a fundamental infrastructural shift with potential applications in different sectors. Its primary goal is to facilitate immutable transactions and enable the development of Decentralized Applications (dApps) without the presence of intermediaries. This technology has the potential to revolutionize outdated business models, making processes more efficient and reducing costs [2]. The concept of blockchain technology was first introduced in 2008 with the advent of Bitcoin. Since its inception, the scope of blockchain applications has expanded significantly beyond digital currencies. The fundamental advantage of blockchain lies in its decentralized architecture, which enables secure, transparent and tamper-resistant data storage, and transactional records, especially in the context of the future 6G networks [3]. In the healthcare sector,

blockchain technology plays a crucial role, particularly in managing data associated with patient Electronic Health Records (EHRs), as well as information related to physicians, specialists, and healthcare institutions. Blockchain has emerged as a fundamental component of modern Health Management Systems (HMS) [4] and its applications to this Sector are discussed in this paper. In particular, the main contributions of this paper are:

- to provide a survey of the main Healthcare Blockchain solutions,
- to present and describe one blockchain implementation for healthcare based on MultiChain platform,
- and to carry out performance comparison and analysis, including smart contract application.

The paper is organized as follows. Section 2 provides an overview of blockchain technology, including its main components, consensus mechanisms, key features, and layered architecture. Section 4 reviews the state of the art in the application of blockchain in the healthcare domain. Section 5 summarizes the most relevant blockchain-based initiatives currently available in the healthcare market. Section 6 describes the practical implementation of a permissioned blockchain using the MultiChain platform. Section 7 describes the approach adopted for the development of the MultiChain-based solution in the ehealth sector, including subcomponents and software codes. Section 8 reports the performance evaluation and related results. Finally, conclusions are drawn.

2. Problem Statement and the Role of Blockchain in the Health Sector

2.1. The Typical Issues in the Health Sector

Modern healthcare systems involve numerous, often independent, stakeholders, such as physicians, hospital staff, laboratory technicians, pharmacists, and others, who must often collaborate throughout the entire patient care journey. Despite the introduction of EHRs, several critical issues remain in the healthcare sector, including the fragmentation of clinical data, limited interoperability between IT systems, data security and traceability, and access control and patient consent management, among others. Because patient information is often fragmented and owned and managed by multiple stakeholders, it is often difficult to obtain a complete and up-to-date view of a patient's clinical picture. Furthermore, this data fragmentation is sometimes exacerbated by limited interoperability between healthcare information systems, which effectively hinders the efficient exchange of information. From a cybersecurity and information privacy perspective, healthcare and clinical data are of particular concern to hackers who aim to target healthcare information systems with cyberattacks, unauthorized access, and privacy violations. Furthermore, there is an increasing need to ensure data integrity and traceability, ensuring that the documentation produced is not modified or altered without the necessary authorization. Efficient and secure access management, combined with patient consent, ensures that only authorized users can access the requested data, in full compliance with current data protection regulations (GDPR and HIPAA). Overall, the critical issues listed reduce the efficiency of patient management, reducing collaboration between stakeholders, and risking compromising the effectiveness of treatment pathways. Based on these guidelines, blockchain is a technology capable of addressing these challenges, thanks to its distributed, transparent, and immutable infrastructure for the secure management of healthcare information. Blockchain operates as an overlay network that coordinates interactions between different information systems, which then continue to store and manage healthcare data within their own databases. Blockchain immutably stores the events and transactions associated with the healthcare data lifecycle, copying into the ledger the cryptographic hashes, metadata, and information regarding access and updates to all healthcare data. These blockchain features allow it to

simultaneously address various issues related to the management of patient healthcare data. The distributed and shared ledger supports interoperability between healthcare institutions that possess healthcare data, as well as allowing the entire history of actions performed on them to be tracked. Hashing data and documents ensures their integrity, preventing access and manipulation/alteration by unauthorized parties because any modification to the data results in the generation of a different hash from the previous one. If we also consider smart contracts as actions performed automatically based on the occurrence of a specific condition, then advanced mechanisms can be implemented to secure access control for stakeholders and patients, even to the extent of limiting or revoking access to their data by certain individuals. Finally, with decentralization, blockchain increases the infrastructure's resilience and data availability even in the event of malfunctions.

2.2. The Blockchain Capabilities for the Health Sector

Blockchain has gained considerable interest in the healthcare sector thanks to the multitude of capabilities it offers [5]. Healthcare Data Protection (HDP) is certainly one of the most important because, thanks to encryption and an untampered distributed ledger, it protects sensitive medical information from unauthorized access and cyberattacks. This allows for the secure management of all data in the Personal Health Record (PHR) Data Management (DRM). Indeed, with blockchain, all involved users (patients, doctors, healthcare professionals, etc.) can securely store, access, and manage their healthcare data while maintaining full control. Blockchain also enables the management of genomic data within the healthcare facility (Point of Care Genomics) and enables the secure sharing of stored genomic information among authorized healthcare professionals, thus promoting personalized and precision medicine. Furthermore, Electronic Health Record (EHR) management allows for the secure storage, management, and retrieval of data contained in electronic health records throughout a patient's healthcare journey. This improves the reliability and integrity of health data, as well as its sharing across different healthcare facilities (Interoperable Electronic Health Records), maintaining consistency and preventing unauthorized modifications. In addition to individual patient care, blockchain provides a valuable and useful contribution to public health by supporting disease and outbreak tracking. Thanks to transparent and traceable mechanisms, blockchain enables the collection and sharing of epidemiological data for health surveillance and the monitoring of epidemics and health emergencies. Finally, genomic data protection (Safeguarding Genomics) can also benefit from the introduction of blockchain, as it ensures secure access to highly sensitive and immutable data, supporting genomic research while protecting the privacy of subjects. For all these capabilities, the decentralization is one of the most important key features of blockchain and involves distributing control and decision-making among all nodes (participants) in the network, rather than allocating them to a single central authority. Unlike traditional centralized systems in which data and its exchange are managed by a central node, in blockchain, all nodes control the exchange of data, which is stored within a distributed ledger. Transactions between nodes exchanging data are validated through specific consensus mechanisms among the blockchain's validator nodes (miners), which thus ensure the transparency and reliability of transactions as well as the immutability of the ledger. Thanks to decentralization, there is no longer any dependency on a central node, which effectively represents a single point of failure. Indeed, in centralized systems, the malfunction or damage of the central authority can render services and data unavailable. With blockchain, the ledger immutably stores transactions, and each node in the network maintains a local copy, which is the same for all nodes. This makes the entire system more resilient to failures and cyberattacks. The decentralization enabled by blockchain also eliminates the need for intermediaries in data exchange between nodes. This reduces

management costs and data processing times, as nodes can interact directly with each other thanks to a peer-to-peer network topology.

3. Blockchain

The blockchain is a distributed digital ledger managed by a network of nodes (peers), comprising a sequence of blocks that store transaction data secured by cryptographic algorithms. Each block is appended in a way that prevents modification or deletion of previous blocks, preserving the integrity of the ledger. Blocks are cryptographically linked, making any alteration immediately detectable by the network of nodes. Validation of the block is achieved through a distributed consensus mechanism, obviating the need for a central authority. Key characteristics of blockchain include:

- Decentralization: Data is distributed across all nodes, each acting as both creator and validator.
- Immutability: Cryptographic protection ensures that stored data cannot be altered or deleted, maintaining ledger consistency.
- Security: The combination of decentralization and immutability ensures resilience against data corruption and guarantees transaction integrity.

Blockchain thus operates as a secure, transparent, and tamper-resistant framework for storing information. Its fundamental components include blocks, hash functions, network of nodes, transactions, and consensus and mining mechanisms. A transaction represents the fundamental unit of data within a blockchain, corresponding to an exchange between users. Each transaction undergoes validation and verification and, once confirmed, is stored immutably. Transactions are initiated using digital signatures that ensure both authenticity and integrity. Consensus mechanisms are used to validate transactions within a block and to finalize their inclusion in the ledger. The blockchain establishes an overlay network that connects all participating peers, as illustrated in Figure 1.

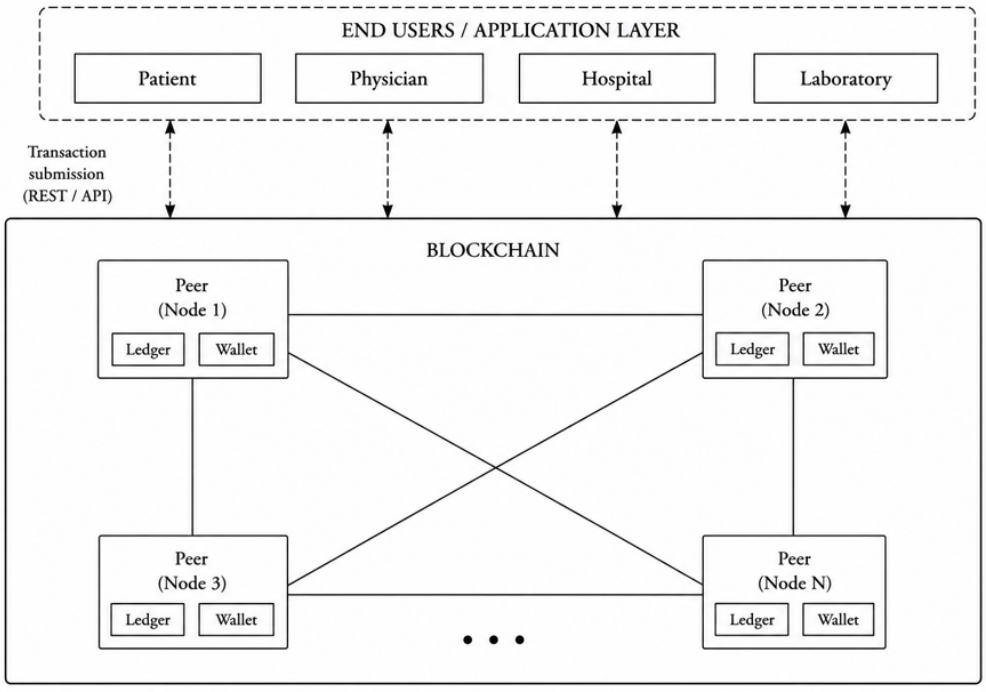


Figure 1. Blockchain as an overlay network.

From the end-user perspective, blockchain introduces several innovations, including user identification, decentralization, data verifiability, health data ownership, data availability, data security, transparency, trust, and smart contracts. Asymmetric cryptography,

based on public-private key pairs, enables secure digital identification and eliminates the need for central authorities, fostering trust across the network. The distributed nature of the blockchain improves fault tolerance and removes single points of failure, while its immutability guaranties that data cannot be modified without network consensus. These properties make blockchain particularly suitable for domains such as finance, law, and healthcare, where secure and transparent data management is critical. From the system architecture, the blockchain can be modeled by adopting a layered model. Each layer is characterized by specific functionalities to improve scalability, security, and efficiency. A detailed overview of layered architectures of the blockchain is provided in [6]. Beyond traditional financial applications, blockchain is recognized as an important enabling technology for next-generation distributed systems, including Internet of Things (IoT), cyber-physical systems, edge/cloud computing, and future 6G communication networks. In these environments, blockchain can be also integrated with Artificial Intelligence (AI) techniques. The blockchain can provide decentralized trust management, secure data sharing, and immutable transaction records, while AI can add optimal mechanisms for autonomous resource management, intelligent service orchestration, and distributed decision making [7]. In Figure 2, the typical blockchain layered architecture is depicted as a composition of six layers [8].

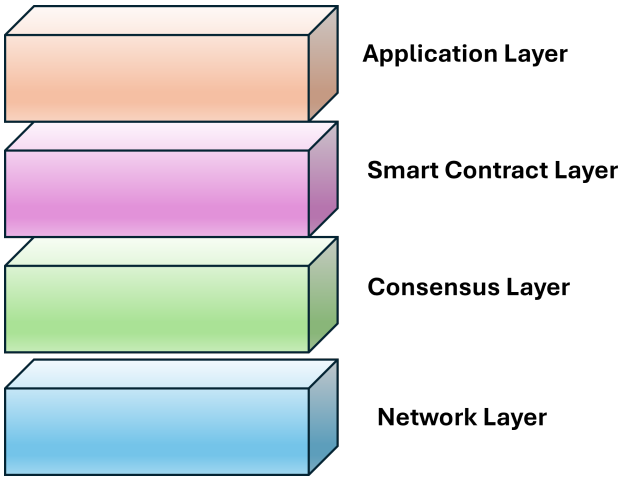


Figure 2. The Blockchain Layered Architecture.

Analyzing this model from the lowest level, we note the Network layer, which generally includes the Internet and provides the general communication tools on which the blockchain is based. Then comes the P2P (Peer-to-Peer) layer, consisting of the network of connected nodes connected in a peer-to-peer topology and protocols for the rapid and efficient exchange of information, such as flooding protocols [9,10]. The Cryptography layer is the next layer and includes the main cryptographic mechanisms that underpin the blockchain's security features [11] for ensuring data integrity, secure information exchange, as well as and the correct implementation of consensus mechanisms [12,13]. This layer includes public-key cryptography, digital signatures, and hash functions. Added to these layers is the Consensus layer, which manages the correct management of the consensus mechanisms necessary for validating transactions between nodes [14]. It can include various consensus mechanisms, including State Machine Replication (SMR) [15], proof-based protocols (such as Proof-of-Work or Proof-of-Stake) [16], and traditional Byzantine Fault Tolerant (BFT) algorithms [17]. Next is the Execution layer, which includes the operational elements necessary for the blockchain to function, such as virtual machines, blocks, transactions, and smart contracts [18]. Finally, the top level is the Applications layer, which includes the potential applications and services used by end users that rely on the

blockchain, such as smart contracts, decentralized applications (DApps), decentralized autonomous organizations (DAOs), and autonomous agents [19].

4. Blockchain for eHealth, State of the Art

The integration of blockchain technology with the healthcare systems holds significant potential to enhance the security, privacy, and interoperability of Electronic Health Records (EHRs). Recent studies have developed and assessed blockchain-based solutions for healthcare, leveraging its immutability, decentralization, and transparency while ensuring compliance with privacy and regulatory standards. As in [20], some use cases can be divided into four main categories: (i) EHR management, (ii) Supply chain management, (iii) Clinical trials, and (iv) Health insurance claims. In terms of secure management, storage, and sharing of EHRs through blockchains, Stamatellis et al. [21] proposed a Hyperledger Fabric-based framework for privacy-preserving EHR management. It can leverage the Idemix protocol in accordance with the General Data Protection Regulation (GDPR) and the relative anonymization requirements. In the same direction, Zaabar et al. [22] proposed HealthBlock, a secure blockchain-based healthcare data management system designed to ensure the integrity and privacy of healthcare data, especially in the case of different systems hosted by multiple hospital. In the context of scalable and privacy-friendly management of healthcare data, Kumari [23] has developed a blockchain-based, patient-centric system that integrates blockchain with the InterPlanetary File System (IPFS) called HealthRec-Chain. The solution is based on a scalable architecture and enables automated and efficient management of medical records while adhering to privacy protection principles through the adoption of appropriate encryption and access control mechanisms. Tran et al. [24] introduced MediChain, a secure decentralized medical data asset management system combining blockchain and smart contracts to ensure that only authorized entities can access medical data. Moreover, MediChain provides a transparent mechanism for guaranteeing th end-to-end traceability and origin of prescription drugs. Liu et al. [25] developed a blockchain-based framework for privacy-preserving medical data sharing within smart healthcare systems, using a cryptographic search function based on multi-keyword search cryptography (MK-IPSE) techniques. Fernandes et al. [26] defined the main components (Client, Certificate Authority (CA), storage, and Hyperledger component) and the related control interfaces and protocols within Hyperledger Fabric for secure data sharing and access control to healthcare data, in three different scenarios: patient visiting a hospital, patient visiting a Blockchain-connected heart clinic, and patient visiting a non-connected clinic. Nagori et al. [27] proposed an architecture based on the MultiChain platform for managing electronic health records (EHR) using data stream exchange, thus enabling more efficient access control. Krai et al. [28] employed a patient-centric multichain model that grants individuals dynamic control over data access and consent. They proposed a patient-centric multichain health record (PCMHR) that implements smart contracts on Ethereum and the Polygon framework with multiple blockchains. Another research direction focuses on decentralized identity management, including authentication, access Control, and consent Management Aziz Torongo and Toorani [29] produced a study on blockchain-based decentralized identity management for healthcare data management, highlighting the need for patients to maintain control of their personal healthcare data.

On the same topic related to the access control in electronic health records (EHR), Sharma et al. [30] proposed a permissioned blockchain network that enables the proactive and decentralized enforcement of access policies. The framework enforces predefined access rules via smart contracts, such as LSTM.

Nedaković et al. in [31] proposed VerifyMed 2.0, a blockchain-based smart healthcare framework that uses Hyperledger Fabric, which stores cryptographic hash values of data. This enables GDPR compliance.

A blockchain-based decentralized authentication management system that allows patients to control and track the sharing of their genomic data has been proposed by Glicksberg et al. in the context of genomic data sharing [32]. Several studies specifically address the security, privacy, and data integrity requirements within the sensitive healthcare context. Zhang et al. [33] identified the main vulnerabilities in data privacy, confidentiality, and susceptibility to cyberattacks in healthcare blockchains, and proposed relevant mitigation strategies for secure blockchain adoption. Cheng, Luo, and Chang [34] introduced an integrity model for eHealth based on a permissioned blockchain. Their solution offers a blockchain-based eHealth Integrity Model and allows for ensuring information integrity through the use of permissioned blockchains combined with off-chain information storage. Martin et al. [35] defined a TrustHealth framework to integrate blockchain with Trusted Execution Environments (TEEs) to protect sensitive healthcare data. These approaches demonstrate how blockchain can be combined with complementary privacy and security mechanisms to protect healthcare information not only during storage and sharing, but also during data processing and verification. In addition to the data storage and access management, blockchain represents an enabling technology distributed analytics and machine learning techniques so to preserve the privacy requirements.

Kuo and Ohno-Machado [36] introduced ModelChain, a predictive modeling framework to preserve privacy in the eHealth sector using private blockchains. Passerat-Palmbach et al. [37] investigated the role of federated learning and machine learning techniques for preserve privacy in the healthcare. They proposed a federated learning architecture for healthcare consortia based on the blockchain. Their solution can allow multiple institutions to share medical date and to perform the training activities in a collaborative way without managing sensitive data. Another research direction regards the necessity to integrate the blockchain with IoT and IoMT devices. These infrastructures generate and transmit medical data in a continuos way. Zhang et al. [38] also analyzed the integration of the blockchain with IoT technologies so to make more secure the data trasmission in real time, especially in the case of the patients to be monitored by remote. Finally, in terms of surveys and systematic reviews.

Roberts et al. [39] reviewed the most important blockchain applications in healthcare, different blockchain infrastructures, together with the definition of data privacy, system interoperability, and security in the eHealth sector. Singh et al. [40] surveyed the blockchain applications for enhancing the data privacy in ehealth sector. They underlined the role of the permissioned blockchain platforms for guaranteeing the data privacy and integrity in line with the regulatory compliance. Wang et al. [41] focused on the need to share electronic health records (EHRs) via blockchain, emphasizing the need for interoperability and suggesting future development opportunities for key healthcare applications. Finally, Kumar et al. [42] analyzed blockchain's contribution to healthcare data privacy and security, systematically organizing key directions for future research on creating secure and resilient healthcare systems capable of protecting patient data from cyber threats. Most of the blockchain technologies used in the applications listed previously are of the private type and based on the Hyperledger fabric. This fact is shown in Tables 1 and 2 also summarizing the key features of the analyzed blockchain-based medical solutions.

Table 1. List of papers on the Permissioned Blockchain Platforms in eHealth (Part 1).

Reference	Key Features	Private Blockchain Used
[21]	Utilizes Hyperledger Fabric to ensure healthcare data privacy. Patient privacy is preserved through encryption. Secure management of healthcare data.	Hyperledger Fabric
[22]	Decentralized EHR storage, secure access control, data integrity, enhanced privacy, and throughput/latency evaluation.	Hyperledger Fabric
[23]	Patient-centric model using IPFS and blockchain for scalable and private health data management.	Hyperledger Fabric
[24]	Decentralized medical data asset management. Focus on patient data control and integrity.	MediChain
[25]	Blockchain-based sharing of medical records in smart health environments. Privacy-centric design.	Hyperledger Fabric
[26]	Data access control and secure sharing using Hyperledger Fabric in eHealth contexts.	Hyperledger Fabric
[27]	MultiChain-based EHR management system enabling secure decentralized storage, patient-controlled data sharing, and predictive analytics.	MultiChain
[28]	Patient-centric multichain healthcare record providing patient-controlled EHR management, interoperability among authorized hospitals, IPFS-based storage, and smart contract-based data management.	Ethereum, Polygon
[29]	Decentralized identity management for healthcare systems, enabling patient-controlled access, secure authentication, selective disclosure, and verifiable credentials.	Hyperledger Fabric
[30]	Access control for electronic health records using permissioned blockchain. Implements decentralized control model.	Hyperledger Fabric
[31]	Privacy-preserving smart healthcare framework. Manages and shares medical data securely through an advanced cryptographic patient authentication.	Hyperledger Fabric

Table 2. List of papers on the Permissioned Blockchain Platforms in eHealth (Part 2).

Reference	Key Features	Private Blockchain Used
[32]	Decentralized system for consent management and secure sharing of genomic health data.	Hyperledger Fabric
[33]	Analyzes security and privacy in healthcare blockchain applications. Focuses on sensitive data protection. Includes vulnerability analysis.	Hyperledger Fabric
[34]	Ensures data integrity and transparency in healthcare systems via permissioned blockchain.	Hyperledger Fabric
[35]	Enhances eHealth security by integrating blockchain and trusted execution environments.	Hyperledger Fabric
[36]	Privacy-preserving predictive modeling for healthcare. Blockchain supports secure model sharing and execution.	Hyperledger Fabric
[37]	Uses blockchain to coordinate federated learning across healthcare providers securely.	Hyperledger Fabric
[38]	Blockchain integration into smart healthcare applications for secure data exchange.	Hyperledger Fabric
[39]	Systematic analysis of blockchain applications in health. Covers security, privacy, and technical design.	Hyperledger, Ethereum, etc.
[40]	Survey of blockchain-based approaches for health data security and privacy.	Hyperledger, Ethereum, etc.
[41]	Exploration of secure electronic health record sharing via permissioned blockchain.	Hyperledger Fabric
[42]	Proposes blockchain-based solutions to ensure medical data privacy and system trust.	Hyperledger Fabric

In the context of eHealth applications, the smart contracts can also increase the benefits deriving from the blockchain-based systems. In [43] smart contracts are used to manage consent for patient data processing, control access, and securely and tamper-proofly share prescriptions. They can also allow for the granting or revoking of authorizations. In [44] they enable the secure and unalterable management of medical image metadata. They can also manage user registration and data access, implementing a three-state authorization mechanism (pending, accepted, and rejected). In [45] smart contracts are used to control secure node access, as well as the execution and correct application of policies for the automated and secure verification and management of data related to telemedicine and IoT applications. Finally, as in evidenced in [46] smart contracts are used to define the right access policies for medical data, securely, automatically, and immutably verifying the requester's identity, reason, and duration before granting or denying access.

5. Blockchains for eHealth: Technologies and the Most Important Initiatives

In this Section we review the main blockchain technologies with their associated features, that have been considered in the most important initiatives in the ehealth sector. The review includes both general-purpose blockchain frameworks and blockchain platforms specifically developed for the healthcare sector. The objective is to provide an overview of the main blockchain-based solutions adopted and/or specifically developed for healthcare applications according to the following selection criteria:

- Presence in the scientific literature or in the target market;
- Applicability and/or specificity for the healthcare sector and Electronic Health Record (EHR) management;
- Availability of sufficient technical information to enable a meaningful comparative analysis.

The final comparison highlights the applicability of the selected blockchain platforms to healthcare environments by considering their main architectural, functional, and operational characteristics, as summarized below:

- **Blockchain type:** Public, private, or consortium (hybrid) blockchain architectures.
- **Licensing model and cost:** Open-source platforms (free) versus proprietary commercial solutions (paid).
- **Consensus mechanism:** The consensus protocol directly affects the security, resilience, transaction latency, and throughput of the blockchain platform. The following consensus mechanisms are considered: Proof of Work (PoW), Proof of Stake (PoS), Proof of Authority (PoA), Practical Byzantine Fault Tolerance (PBFT), Istanbul Byzantine Fault Tolerance (IBFT), Raft, Byzantine Fault Tolerance (BFT) (used in the Corda Notary service), Keyless Signature Infrastructure (KSI), and Proprietary Proof of Stake (Prop. PoS).
- **Smart contract support:** Smart contracts enable the implementation of automated healthcare workflows, access control mechanisms, and patient consent management.
- **Performance metrics:** The main Key Performance Indicators (KPIs) considered are transaction latency, throughput, expressed in Transactions Per Second (TPS), and scalability. These metrics are essential for evaluating the suitability of blockchain platforms for large-scale healthcare environments.
- **Privacy protection mechanisms:** Privacy preservation is achieved through technologies such as encryption, Role-Based Access Control (RBAC), Zero-Knowledge Proof (ZKP), private transactions, and data isolation, which contribute to protecting sensitive healthcare information.

- **Regulatory compliance:** The capability of each blockchain platform to support compliance with healthcare regulations related to the protection of personal and sensitive data, including the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR).
- **Implementation cost:** The evaluation also considers the expected deployment costs, including software licensing, infrastructure requirements, and operational expenses.

5.1. Hyperledger

Hyperledger is an open-source blockchain incubator initiated by the Linux Foundation in 2015 and supported by major industry players, including IBM, Intel, JP Morgan, and Cisco [47]. Its objective is to foster cross-industry collaboration for the development of blockchain and distributed ledger technologies (DLTs), with a focus on enhancing system performance and reliability. In October 2018, Hyperledger partnered with the Ethereum Enterprise Alliance (EEA) to accelerate enterprise-level technology integration [48]. The Hyperledger project encompasses multiple frameworks, with Hyperledger Fabric being the most versatile and widely adopted, and the focus of this study. Fabric enables the development of blockchain applications tailored to specific business requirements and is based on a modular architecture that allows customization of features, including the consensus protocol, to suit particular use cases. Unlike most blockchains, including Ethereum, which follow an "order-execute" architecture, Hyperledger Fabric employs an "execute-order-validate" model, whereby transactions are executed prior to reaching consensus on their ordering, thus filtering out inconsistent transactions. A key innovation in Fabric is the categorization of peers into three types, Endorser, Committer, and Acceptor, which ensures that information sharing is restricted to designated network participants. The primary feature of Hyperledger Fabric is its private, permissioned blockchain, where all participants possess a digital identity (ID) that defines their access permissions and roles (e.g., member, administrator, client, or peer) and is verified by a trusted authority [49]. The Membership Service Provider (MSP) governs the rules for validating identities within an organization. Hyperledger Fabric supports multiple ledgers, referred to as Channels, which can operate concurrently. Each Channel comprises a world state and a blockchain recording concatenated transaction blocks. Channels provide private communication sub-networks for members, ensuring confidential transactions. Transactions within a Channel require authentication and authorization for all participating actors. Transactions in Hyperledger Fabric can be classified into two types:

- Deploy Transaction: Creates a new chaincode (smart contract) and takes a program as a parameter. Upon successful execution, the chaincode is installed on the peers within the Channel.
- Invoke Transaction: Executes a function of an existing chaincode, reading or modifying the world state.

Chaincode (smart contracts) in Hyperledger Fabric can be implemented using multiple programming languages, including Java, Node.js, and Go. The "Execute-Order-Validate" transaction flow in Hyperledger Fabric differs from the traditional "Order-Execute" model employed in public blockchains such as Ethereum, offering enhanced performance, privacy, and scalability. The process begins when a client application generates a transaction proposal summarizing the intended ledger changes. The proposal is signed by the client and submitted to endorsing peers, selected according to a predefined endorsement policy. These peers simulate the transaction logic using the chaincode and generate a read/write set, documenting the data affected without immediately altering the ledger. Upon endorsement, the peers return the signed proposal to the client. After collecting the required endorsements, the client submits the transaction to the ordering service, which sequences

transactions into blocks. Unlike public blockchains, the ordering service in Hyperledger Fabric does not perform validation, but only determines the order of transactions. Ordering can utilize consensus protocols such as Kafka, Raft, or Byzantine Fault Tolerant (BFT) mechanisms. The blocks are subsequently distributed to committing peers, which validate transactions against the endorsement policy and perform multi-version concurrency control (MVCC) checks to detect conflicts and prevent double-spending. Valid transactions are appended to the ledger, and an event notification is sent to the client. This "Execute-Order-Validate" workflow significantly improves throughput by enabling parallel transaction execution, as execution is decoupled from ordering and validation. The modular architecture of Hyperledger Fabric supports privacy features and fine-grained access control, making it particularly suitable for enterprise applications in healthcare, finance, and supply chain management. Its permissioned model ensures strong security, data integrity, and efficient consensus, distinguishing it from fully decentralized public blockchains.

5.2. Ethereum

Ethereum was first introduced by Vitalik Buterin in a 2013 article [50] with the objective of creating a blockchain capable of executing decentralized applications. Unlike specialized blockchains, Ethereum is general-purpose and programmable through smart contracts, enabling a wide range of use cases. Ethereum can be conceptualized as a global computer where smart contracts are executed in a decentralized, continuous, and uncensored manner. Its blockchain is public and permissionless, initially employing Proof of Work (PoW) as the consensus algorithm [51], and transitioning to Proof of Stake (PoS) on 15 September 2022 [52]. To prevent attacks or programming errors in smart contracts—such as infinite loops that could overload the blockchain—Ethereum introduced a resource metering mechanism called "gas." Gas serves as the computational fuel for contracts, quantifying the resources required to perform operations on the blockchain. Each operation has a predefined gas cost, as detailed in the Ethereum Yellow Paper by G. Wood [53]. Every transaction in Ethereum specifies two parameters: the gas price and the gas limit, which defines the maximum gas that can be consumed. If a transaction requires more gas than the specified limit, execution halts, and all gas is consumed. In contrast, any unused gas is returned to the sender. Ethereum supports two types of accounts: externally owned accounts (EOAs), controlled by private keys and capable of sending and receiving transactions, and contract accounts, which are associated with smart contracts and execute predefined code upon receiving a transaction. Smart contracts are typically developed in the Solidity programming language [54] and executed on the Ethereum Virtual Machine (EVM), which ensures that contracts run consistently across all full nodes in a secure, isolated environment. The Ethereum Request for Comment (ERC) standards define specific functionalities for smart contracts. Among these, ERC-20 [55] is widely used for fungible tokens, particularly in initial coin offerings (ICOs), while ERC-721 [56] defines non-fungible tokens (NFTs), which are unique and indivisible.

5.3. MedRec

MedRec is a blockchain-based system specifically designed for the secure and decentralized management of EHRs [57]. Originally developed by researchers at the MIT Media Lab, MedRec aims to enhance patient data ownership, interoperability, and access management, while preserving security and privacy. Leveraging Ethereum blockchain technology, MedRec constructs a patient-centric and tamper-resistant ledger of medical records, in contrast to conventional EHR systems that rely on centralized databases vulnerable to breaches and interoperability challenges. Due to the volume of data and privacy concerns, MedRec does not store actual medical records directly on the blockchain. Instead, it functions as a

metadata management system, recording data access logs, permissions, and cryptographic references to off-chain storage systems that maintain the actual medical data. This approach preserves patient privacy while harnessing the immutability, security, and decentralization inherent to blockchain technology. Patients control access to their medical history through a smart contract-based access control mechanism central to MedRec. By granting rights to healthcare providers via Ethereum smart contracts, patients act as the ultimate custodians of their data. These contracts also record access policies and generate auditable, time-stamped logs of all interactions with patient information, ensuring transparency and accountability. Initially operating under Ethereum's PoW consensus mechanism, MedRec has introduced a novel incentive structure tailored for the healthcare sector. In this model, hospitals, academic institutions, and insurers participate in mining transactions to maintain the blockchain. Rather than receiving monetary rewards, these stakeholders gain privileged access to anonymized health data for research purposes, creating a mutually beneficial relationship between blockchain security and medical research advancement.

5.4. BurstIQ

BurstIQ is a blockchain platform specifically designed for the healthcare sector, providing secure data sharing, identity management, and AI-powered insights [58]. The platform implements a fully compliant, decentralized architecture that allows organizations to share and analyze health data while maintaining strict privacy and security standards, in contrast to traditional EHR solutions. By leveraging blockchain, artificial intelligence, and big data, BurstIQ enhances healthcare operations, improves patient care, and enables secure data monetization. BurstIQ employs a hybrid blockchain architecture that supports permissioned access while preserving immutability and data integrity, addressing the scalability and privacy limitations of public blockchains. Rather than storing raw patient data on-chain, the blockchain maintains smart contracts and cryptographic metadata that govern ownership rights, data-sharing policies, and access permissions. The blockchain serves as a trust layer to monitor and verify access events, while patient records are securely stored in off-chain repositories compliant with GDPR and Health Insurance Portability and Accountability Act (HIPAA) regulations. A central feature of BurstIQ is "LifeGraphs," an AI-enhanced functionality that generates dynamic health profiles for individuals, devices, and organizations. LifeGraphs integrate medical history, genetics, lifestyle factors, medications, and data from IoT devices to provide real-time analytics and personalized healthcare recommendations. BurstIQ employs a modified PoS consensus mechanism optimized for enterprise applications. Compared to traditional PoW blockchains, this approach ensures high throughput, low latency, and energy efficiency, enhancing scalability. In addition, the platform leverages smart contracts to facilitate consent management, regulatory compliance, and secure data exchange among research institutions, insurers, healthcare providers, and patients.

5.5. IBM Blockchain

IBM Blockchain is a permissioned, enterprise-grade blockchain platform designed to support secure, transparent, and efficient transactions across industries such as healthcare, supply chain management, and finance [59]. Built on Hyperledger Fabric, IBM Blockchain offers a highly flexible, scalable, and regulatory-compliant solution for enterprises that require privacy, interoperability, and governance over distributed networks. In the healthcare sector, IBM Blockchain facilitates secure data exchange, prescription traceability, patient identity verification, and fraud prevention. It ensures compliance with regulations such as HIPAA and GDPR, while enabling collaboration among hospitals, pharmaceutical companies, insurers, and research institutions through an auditable and tamper-resistant ledger.

Unlike public blockchains such as Ethereum or Bitcoin, IBM Blockchain operates as a permissioned network, allowing only authorized participants to join, read, and write transactions. The platform leverages Hyperledger Fabric's modular architecture to provide customizable features tailored to enterprise-specific requirements.

- Channel-based privacy: Enables organizations to create private sub-networks (channels) where only authorized participants can access specific transaction data.
- Smart contracts (chaincode): Automate business logic for multi-party agreements, enforce compliance, and ensure secure data access.
- Off-chain storage: Sensitive healthcare data is stored securely in external databases (such as CouchDB, cloud storage, or on-premise solutions), while the blockchain maintains an immutable record of access logs and cryptographic hashes.

IBM Blockchain employs a pluggable consensus mechanism, enabling organizations to select between Byzantine Fault Tolerant (BFT) and Crash Fault Tolerant (CFT) algorithms according to their specific performance and security requirements. This flexibility is particularly advantageous for high-throughput enterprise applications, providing low-latency transaction finality. Unlike traditional PoW or PoS mechanisms, IBM Blockchain leverages Hyperledger Fabric's endorsement-based consensus. Transactions are processed through three distinct stages:

1. Execution: To generate a simulated outcome, smart contracts are carried out by assigned endorsing peers.
2. Ordering: To maintain uniformity throughout the network, the ordering service groups transactions into blocks without actually executing them.
3. Validation and Commitment: Transactions are recorded on the distributed ledger after being verified by endorsement standards.

For healthcare applications requiring real-time data access and regulatory compliance, the deterministic transaction processing, scalability, and efficiency of this architecture provide significant advantages. Additionally, it reduces computational overhead, enhancing overall system performance and responsiveness.

5.6. Solve.Care

Solve.Care is a blockchain-based platform designed to enhance payment processing, administrative efficiency, and care coordination within the healthcare sector [60]. Unlike traditional EHR systems, which are often fragmented and inefficient, Solve.Care leverages decentralized blockchain technology to establish an automated, transparent, and patient-centered healthcare ecosystem. The platform addresses critical challenges such as fraud prevention, high administrative costs, and delayed reimbursements by enabling secure data exchange, instant payments, and smart contract-driven automation. By integrating blockchain, smart contracts, and digital identity management, Solve.Care facilitates seamless communication among patients, insurers, and healthcare providers while ensuring compliance with data protection regulations such as GDPR and HIPAA. Solve.Care operates on a permissioned blockchain, ensuring that transactions are accessible and validated only by authorized parties. Utilizing Care.Wallets, a blockchain-based personal health management solution, patients and healthcare providers can securely exchange health data through cryptographic references and permission-based access, without storing raw patient information directly on the blockchain. The platform is structured around three core components:

- Care.Wallet: A decentralized healthcare application that enables users to schedule appointments, manage payments, and control access to their medical information.

- Care.Cards: Configurable smart contracts that automate various healthcare processes, including medication adherence, referral management, and claim settlement.
- Care.Protocol: A decentralized interoperability layer that ensures seamless communication among different payment processors and healthcare organizations.

Solve.Care automates administrative workflows using Ethereum-based smart contracts, reducing fraud, errors, and inefficiencies in billing and claims processing. The platform also offers SOLVE tokens, a native digital currency that incentivizes network participation and facilitates cross-border payments among healthcare providers. A hybrid blockchain architecture underpins Solve.Care, combining a public Ethereum-based layer for token transactions and smart contract execution with a private, permissioned layer to ensure compliance with healthcare regulations. The platform incorporates multiple mechanisms to guarantee security and data integrity:

- Decentralized Identity Management: Patients retain full control over their data, granting or denying access through their Care.Wallet.
- Role-Based Access Control: Only authorized insurers and healthcare professionals can access medical records, subject to patient approval.
- End-to-End Encryption: All data exchanges and transactions are securely encrypted to prevent unauthorized access.

5.7. EHRData

EHRData is a blockchain-based healthcare data management platform designed to enhance the security, interoperability, and usability of EHRs [61]. Unlike traditional EHR systems, which often suffer from security vulnerabilities, limited patient control, and data fragmentation, EHRData leverages blockchain technology to create a patient-centric, tamper-resistant, and interoperable healthcare data network. The platform empowers patients with full ownership and control over their medical information while enabling real-time data exchange among public health organizations, insurers, researchers, and healthcare providers. By integrating blockchain, smart contracts, and decentralized identity management, EHRData enhances healthcare efficiency, transparency, and regulatory compliance. EHRData employs a permissioned blockchain architecture to safeguard data privacy and integrity, restricting access and transaction validation to authorized participants. Instead of storing actual medical records on-chain, the platform maintains metadata and cryptographic hashes, while sensitive patient data is securely stored off-chain. Key components of the EHRData system include:

- Patient-Centric Data Ownership: Through a decentralized identity system, patients can grant or revoke access to their medical information, ensuring full ownership and control.
- Interoperable Health Records: These facilitate seamless data exchange among healthcare facilities, pharmacies, insurers, and research institutions.
- Data-Sharing Agreements: Automated via smart contracts for consent management, ensuring that records are accessible only under authorized conditions.

The platform can be deployed globally, supported by a Regulatory Compliance Layer that ensures adherence to HIPAA, GDPR, and other healthcare regulations. By enabling secure data exchange and validation, EHRData reduces medical errors, administrative inefficiencies, and duplicate records. Built on a Hyperledger Fabric-based private permissioned blockchain, EHRData offers several advantages:

- Endorsement-Based Consensus: Transactions must be validated by designated endorsing peers to ensure data confidentiality and consistency.

- Privacy-Enhancing Features: Channel-based data partitioning restricts access to authorized participants only.
- Tamper-Proof Audit Logs: All transactions related to access and modification of patient data are permanently recorded, providing full traceability.

5.8. Guardtime

Guardtime is a blockchain-based cybersecurity company specializing in secure data exchange, digital identity verification, and data integrity [62]. The firm utilizes its proprietary Keyless Signature Infrastructure (KSI), a scalable, high-performance, and quantum-resistant blockchain technology designed to provide tamper-proof digital records. Unlike conventional blockchain platforms relying on distributed ledgers and smart contracts, Guardtime's KSI was originally developed for corporate and national security applications but has since been extended to government, healthcare, and supply chain sectors, offering real-time audits, data integrity assurance, and regulatory compliance. In the healthcare domain, Guardtime's technology mitigates risks such as fraud, data breaches, and counterfeit drugs by ensuring secure patient data management, clinical trial integrity, and pharmaceutical traceability. Instead of employing PoW or PoS, the KSI blockchain uses a hash-tree (Merkle tree) aggregation structure, providing efficient, immutable, and mathematically verifiable data integrity proofs. Guardtime's architecture is highly scalable, capable of processing billions of transactions per second. Key components of Guardtime's system include:

- Keyless Signature Infrastructure (KSI): A hash-based, quantum-secure cryptographic signature that guarantees data integrity without requiring a centralized trust authority.
- Tamper-Proof Data Verification: Ensures that prescriptions, clinical trials, and medical data cannot be altered surreptitiously.
- Interoperability and Compliance: Supports cross-border healthcare data sharing while ensuring adherence to FDA, GDPR, and HIPAA regulations.
- Quantum Resistance: KSI is resistant to quantum computing threats, providing future-proof security compared to conventional public-key cryptography.

Guardtime's solutions enable automated regulatory audits and real-time patient data verification by integrating seamlessly with existing EHR systems, medical IoT devices, and cloud storage platforms. Using a hash-based consensus mechanism, KSI eliminates the need for miners, staking, or validators. The process operates as follows:

- Each transaction is hashed and time-stamped before being aggregated into a larger data structure.
- Mathematical proofs are stored in a global ledger, allowing any participant to verify data integrity without exposing the underlying information.
- Consensus is achieved through decentralized verification nodes, preventing fraud, data tampering, and unauthorized access.

This approach enables real-time verification without computational overhead, making Guardtime's KSI blockchain a highly efficient and scalable solution for both enterprise and healthcare applications.

5.9. MultiChain

MultiChain is a robust and configurable platform designed for the deployment and management of private, permissioned blockchain networks [63]. Originating from a fork of the Bitcoin Core client, MultiChain maintains cross-platform compatibility with major operating systems, including Windows, Linux, and macOS [64]. Its architecture prioritizes restricted blockchain visibility and controlled access, employing integrated permission systems to enforce secure governance and regulate transaction execution. A key innovation of MultiChain is the replacement of traditional PoW consensus with a lightweight mining

protocol (Proof of Authority, PoA), which guarantees network security while significantly reducing computational overhead. This approach enables deterministic transaction validation exclusively among authorized participants. MultiChain supports the addition of new nodes through an authenticated handshake process, ensuring compliance with established configuration standards. Its ability to host multiple distinct blockchain instances on a single server highlights the platform's scalability and adaptability. Owing to its emphasis on security, configurability, and operational control, MultiChain is particularly suitable for enterprise and institutional applications that require private, efficient, and customizable blockchain solutions. MultiChain is a permissioned and private blockchain platform that restricts participation to authorized entities, making it suitable for healthcare environments where controlled access and data governance are essential. As an open-source solution, it represents a cost-effective alternative to proprietary platforms, with expenses primarily associated with infrastructure deployment and maintenance. Unlike Ethereum, MultiChain employs a lightweight permissioned consensus model based on authorized validator nodes, reducing computational overhead and improving operational efficiency. Although MultiChain does not natively support advanced smart-contract environments such as Solidity or the EVM, it provides streams for managing structured data, which are well suited to healthcare applications involving electronic health records, audit logs, and access-control information. Furthermore, MultiChain offers low latency, high throughput (typically up to 1000+ TPS), and support for parallel blockchains, enabling scalable deployments. As a general-purpose blockchain framework, it can be integrated with external applications and healthcare standards such as HL7 and FHIR to implement functionalities including patient consent management, privacy preservation, and secure health data exchange.

5.10. Quorum

Quorum is an enterprise-grade blockchain platform originally developed by J.P. Morgan as an extension of the Ethereum protocol, specifically designed to address the privacy, scalability, and performance requirements of business-critical applications [65]. Unlike public Ethereum networks, Quorum operates as a permissioned blockchain in which participation is restricted to authorized entities. This characteristic makes it particularly suitable for healthcare environments, where sensitive patient information must be shared among trusted stakeholders, including hospitals, laboratories, insurance companies, and regulatory agencies. By maintaining compatibility with the Ethereum Virtual Machine (EVM), Quorum enables the deployment of smart contracts written in Solidity while providing enhanced confidentiality and transaction throughput. A key feature of Quorum is its support for private transactions, which allow sensitive information to be shared exclusively among authorized participants while maintaining the integrity of the distributed ledger. This capability is achieved through dedicated privacy managers that separate confidential transaction payloads from publicly visible metadata. In addition, Quorum replaces Ethereum's resource-intensive consensus mechanisms with more efficient protocols such as Istanbul Byzantine Fault Tolerance (IBFT) and Raft, significantly reducing transaction latency and increasing throughput. These characteristics make Quorum particularly attractive for healthcare applications involving secure medical record sharing, claims processing, consent management, and inter-organizational data exchange, where both privacy preservation and operational efficiency are essential requirements.

5.11. Corda

Corda is a permissioned distributed ledger platform developed by R3 with the objective of facilitating secure transactions among regulated organizations while preserving data confidentiality [66] Unlike traditional blockchain platforms, Corda does not broadcast

transactions to all network participants. Instead, transaction data are shared only with the parties directly involved in a given interaction, thereby minimizing information exposure and improving privacy. This design philosophy makes Corda particularly suitable for healthcare ecosystems, where medical records, insurance claims, and clinical data often contain highly sensitive information subject to strict regulatory requirements. Smart contracts in Corda are implemented using Java or Kotlin and are integrated into a transaction model based on states, contracts, and flows. States represent shared facts recorded on the ledger, contracts define the rules governing state transitions, and flows coordinate communication among participating nodes. Transaction validity is ensured through a notary service that prevents double-spending and validates transaction uniqueness using consensus protocols such as Raft or Byzantine Fault Tolerant mechanisms. The selective data-sharing model adopted by Corda, combined with strong identity management, cryptographic security, and regulatory compliance capabilities, makes the platform well suited for healthcare applications including electronic health record management, patient consent tracking, medical supply chain monitoring, and healthcare billing processes.

5.12. Comparison

The blockchain platforms described for the eHealth sector exhibit distinct characteristics. Table 3 provides a comparative analysis of these platforms, evaluating key attributes such as access type, licensing model, consensus mechanism, smart contract support, latency, and throughput, measured in transactions per second (TPS). Regarding access type, Ethereum, MedRec, and Solve.Care are public blockchains, accessible to all participants, whereas Hyperledger Fabric, IBM Blockchain, EHRData, BurstIQ, and MultiChain are private, permissioned networks, allowing only authorized users to participate. Quorum and Corda can be deployed either as private or consortium-based distributed ledger platforms, enabling multiple trusted organizations to securely share data while maintaining controlled access. Private and consortium blockchains generally provide enhanced security, governance, and privacy compared with public networks due to their restricted participation model. Guardtime, in contrast, employs a hash-based system and does not support smart contracts. Concerning the licensing model, Ethereum, Hyperledger Fabric, Quorum, Corda, MultiChain, and MedRec are available as open-source platforms, permitting unrestricted usage and customization. Many healthcare-oriented solutions, including IBM Blockchain, BurstIQ, EHRData, and Solve.Care, require proprietary licenses or commercial subscriptions. Although Hyperledger Fabric, Quorum, Corda, and MultiChain are open-source, their deployment and maintenance may still involve significant infrastructure and operational costs depending on the scale of the healthcare environment. With respect to consensus mechanisms, Ethereum and Solve.Care implement PoS, a more energy-efficient protocol compared to the PoW mechanism originally employed by MedRec on Ethereum. Although PoW provides strong security guarantees, it is computationally intensive. Hyperledger Fabric and IBM Blockchain adopt Practical Byzantine Fault Tolerance (PBFT), a faster but more centralized approach. This modular, permissioned architecture enables high scalability and supports parallel transaction processing via private channels. Quorum utilizes enterprise-oriented consensus protocols such as Istanbul Byzantine Fault Tolerance (IBFT) and Raft, providing low latency and high throughput while preserving fault tolerance. Similarly, Corda employs a notary-based validation mechanism using either Raft or Byzantine Fault Tolerant protocols to guarantee transaction uniqueness and integrity without requiring global transaction broadcasting. BurstIQ employs a proprietary PoS-based consensus, while Guardtime relies on its hash-based KSI blockchain. Regarding smart contract support, all platforms except Guardtime are compatible with smart contracts. Ethereum, MedRec, Solve.Care, and Quorum support smart contracts written in

Solidity through compatibility with the Ethereum Virtual Machine (EVM). Hyperledger Fabric employs Chaincode, a dedicated framework supporting multiple programming languages such as Go, Java, and JavaScript. Corda implements smart contracts using Java and Kotlin, enabling the development of enterprise-grade applications while leveraging existing software engineering tools and practices. MultiChain provides limited smart-contract functionality through custom scripting and permission management mechanisms. Finally, in terms of latency and throughput, permissioned blockchains such as Hyperledger Fabric, IBM Blockchain, Quorum, Corda, and MultiChain exhibit low latency, typically measured in milliseconds, and throughput often exceeding 1000 TPS. Public blockchains, including Ethereum and MedRec, show higher latencies and lower throughput (approximately 15–100 TPS), primarily due to their public access model and PoW/PoS consensus protocols. Both BurstIQ and Guardtime demonstrate throughput above 1000 TPS. In particular, Guardtime achieves exceptionally high scalability by employing a KSI infrastructure instead of conventional consensus protocols, enabling large-scale data validation with minimal computational overhead. In contrast, Ethereum and Ethereum-based platforms, such as Solve.Care and MedRec, exhibit moderate to low scalability due to the inherent limitations of public networks, including higher latency and limited throughput. These challenges are only partially mitigated by ongoing Layer-2 solutions and scaling initiatives. MultiChain is consistent with prior platforms such as MedRec and Guardtime in storing sensitive healthcare information off-chain within EHR repositories, thereby improving privacy protection and regulatory compliance. To safeguard patient information, the platform employs Role-Based Access Control (RBAC) and encryption mechanisms. Quorum adopts a similar approach, while additionally supporting private transactions that allow confidential data to be exchanged exclusively among authorized participants. Corda further strengthens privacy through its data-isolation model, in which transaction information is shared only among directly involved parties rather than being replicated across the entire network. These privacy-preserving mechanisms facilitate compliance with healthcare regulations such as GDPR and HIPAA while enabling secure data sharing among healthcare providers, insurers, and other stakeholders.

Table 3. Comparison of Healthcare Blockchain Platforms.

Blockchain	Type	Free/Paid	Consensus	Smart Contract	Latency	TPS	Scal.	Privacy	Compliance	Cost	Ref.
Hyperledger Fabric	Priv.	OS	PBFT	Yes	Low	1k	High	RBAC, Enc.	HIPAA	Mod.-High	[47–49]
Ethereum	Publ.	OS	PoS	Yes	High	15–100	Mod.	Pseudo-nymity, Enc.	HIPAA *, GDPR *	High	[50–56]
MedRec	Publ.	OS (MIT)	PoW	Yes	High	15	Low	RBAC, Enc.	HIPAA	High	[57]
BurstIQ	Priv.	Paid	Prop. PoS	Yes	Med.	1k+	High	RBAC, Enc.	HIPAA	High	[58]
IBM Blockchain	Priv.	Paid	PBFT	Yes	Low	1k	High	RBAC, Enc.	HIPAA	High	[59]
Solve.Care	Publ.	Paid	PoS	Yes	High	100	Mod.	RBAC, Enc.	HIPAA	High	[60]
EHRData	Priv.	Paid	PBFT	Yes	Low	1k	High	RBAC, Enc.	HIPAA	High	[61]
Guardtime	Priv.	Paid	KSI	No	Low	1k+	V. High	ZKP, Diff. Priv.	GDPR, HIPAA	High	[62]
MultiChain	Priv.	OS	PoA	Limited	Low	1k+	High	RBAC, Enc.	GDPR, HIPAA	Mod.	[63,64]
Quorum	Priv./Cons.	OS	IBFT/Raft	Yes	Low	1k+	High	RBAC, Enc., Private Tx	HIPAA, GDPR	Mod.	[65]
Corda	Priv./Cons.	OS	Notary (Raft/BFT)	Yes	Low	1k+	High	RBAC, Enc., Data Isolation	HIPAA, GDPR	Mod.-High	[66]

* Compliance with HIPAA/GDPR on public Ethereum depends on the application architecture, encryption mechanisms, off-chain storage, and governance policies.

6. Proposed Solution: IBEH

In this Section we describe the proposed IBEH solution for the eHealth sector and we compare it with the other available solutions. The proposed IBEH architecture adopts a client-server and distributed ledger approach. The application logic is separated from the blockchain infrastructure. The technical and scientific description of the IBEH architecture is depicted in Figure 3.

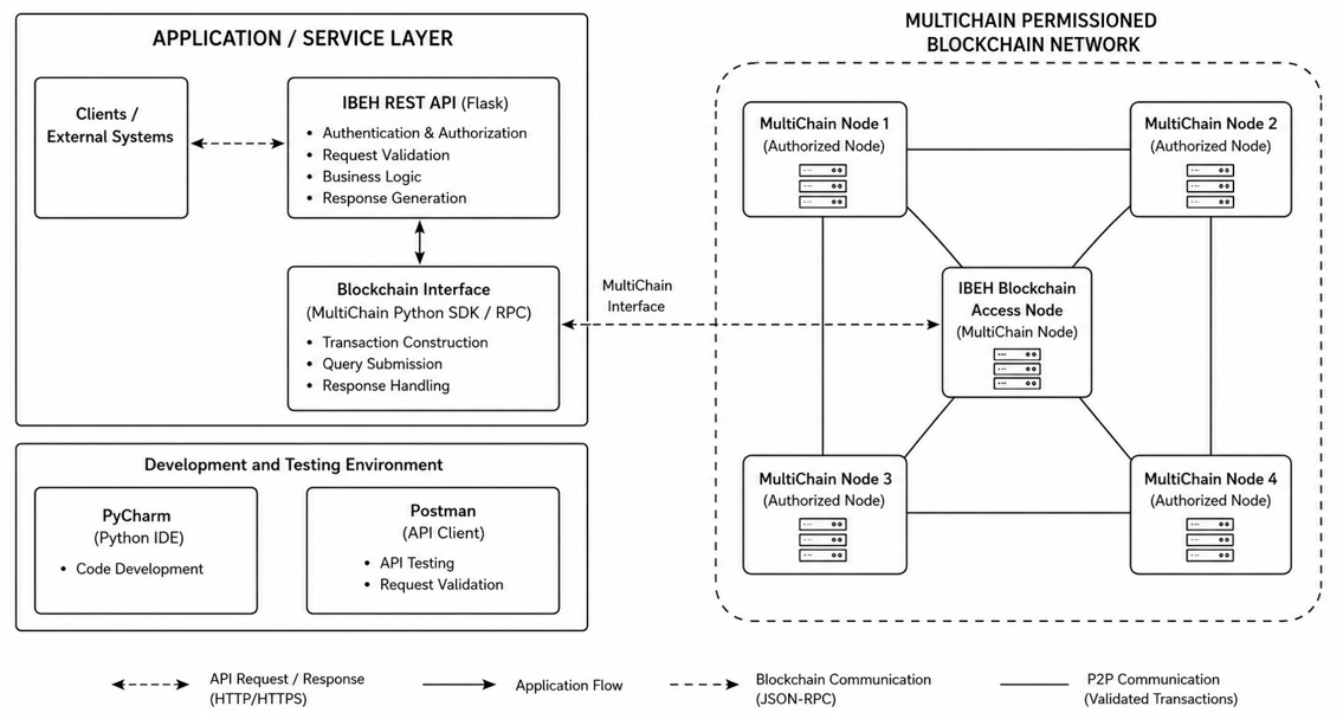


Figure 3. System Architecture of the IBEH solution.

The architecture of IBEH offers both a fully managed and verifiable environment for managing healthcare information and a blockchain layer for decentralization, integrity, and traceability, with access and read/write permissions. From an application perspective, IBEH features a backend implemented in Python 3.11. The Flask framework provides the REST API layer through which external applications and authorized users can interact with the blockchain infrastructure. Postman version 12.3.0 and PyCharm version 2022.2.5 are also included. The former is used to generate and validate API requests during the development and testing phases, while the latter is used as an integrated development environment in Python for implementing the application logic and smart contracts. Therefore, these components constitute the software development and service access environment. The blockchain layer is implemented using the MultiChain private blockchain framework, subject to specific access permissions. The blockchain is composed of several blockchain nodes interconnected with a peer-to-peer topology. Unlike public blockchains, nodes in the IBEH network can only access the network with the authorization of the full node(s). Based on the blockchain's properties, each IBEH node maintains a synchronized copy of the distributed ledger locally and participates in validating transactions according to MultiChain's consensus and authorization mechanisms (Proof of Authority). Communication between the application and the blockchain layer occurs through a MultiChain full node that serves as an access point to the IBEH blockchain, represented by the node highlighted in Figure 3. The Flask-based application translates user-requested operations via REST APIs and communicates with this node through the MultiChain blockchain interface. The MultiChain full node then validates transactions and stores them in the distributed ledger,

where they are recorded untampered. Conversely, the application can request and send requests to the blockchain to operate on the data through the same interface. This way, the IBEH backend can retrieve the information contained in the distributed ledger and return the response associated with the specific operation via the API layer. Functionally, the architecture implements the following workflow for write operations. For read operations, the workflow is followed in the opposite way. The separation between the application and blockchain layers allows for a modular architecture, since logically, the healthcare services and API logic are separated from the underlying distributed ledger. The IBEH solution was developed using Python 3.11, which is the development environment for the Flask-based application and its interaction with the MultiChain network. The experimental implementation was deployed and tested in a real-world environment with multiple multichain nodes. This allowed us to test and functionally validate the following elements:

- API calls
- blockchain transactions
- communication between the application layer and the distributed ledger before deployment in a larger, multi-organizational healthcare environment.

6.1. Multichain and Its Innovative Features for the IBEH

The IBEH solution is built on MultiChain, providing a flexible, open-source, and high-performance alternative to existing blockchain platforms. Unlike vertical applications such as MedRec or Solve.Care, which are tightly coupled with Ethereum's public infrastructure, MultiChain offers a configurable permissioned environment with lower latency and higher throughput, making it particularly suitable for large-scale healthcare data management. The novelty of IBEH lies in leveraging MultiChain's stream functionality to structure healthcare records, consent tracking, and auditing processes in a secure and organized manner, while simultaneously enabling interoperability with smart contracts through sidechain or EVM integration [67]. This dual capability addresses a critical gap in the current landscape: the lack of cost-efficient, customizable blockchain frameworks that can be tailored to healthcare-specific requirements without being constrained by the scalability and latency limitations of public blockchains or the rigidity of commercial permissioned platforms. By positioning itself between general-purpose permissioned frameworks (e.g., Hyperledger Fabric) and healthcare-specific blockchain applications (e.g., MedRec), MultiChain occupies a strategic niche, combining adaptability, performance, and cost-effectiveness to support privacy-preserving and interoperable eHealth systems. The primary challenge addressed by IBEH in the healthcare context is the absence of a private, high-performance, and cost-efficient blockchain platform capable of ensuring security, interoperability, and scalability for healthcare applications. Public blockchains, including Ethereum-based platforms such as MedRec and Solve.Care, suffer from high latency (6–12 s) and limited throughput (15–100 TPS), making them unsuitable for real-time management of large clinical datasets. They also incur high transaction costs and provide limited control over access to sensitive information. Commercial private solutions, such as IBM Blockchain and BurstIQ, while robust, are expensive, require proprietary infrastructure, and offer limited flexibility for integrating smart contracts or existing healthcare systems. In contrast, MultiChain delivers a permissioned environment with low latency and high throughput (500–1000+ TPS), is open source, and enables fine-grained customization through features such as streams and smart contract support via sidechains or EVM integration.

6.2. Data Privacy and Regulatory Compliance

Table 4 summarizes the contributions of a MultiChain-based blockchain for eHealth to critical aspects of data privacy, regulatory compliance, and ethical governance, high-

lighting its role in enabling secure, auditable, and patient-centric management of sensitive health information.

Table 4. Contribution of the proposed MultiChain-based eHealth blockchain (IBEH) to privacy, compliance, and ethics.

Area	Contribution of IBEH
Data Privacy	Supports GDPR and HIPAA requirements through a permissioned architecture, granular access control, encryption mechanisms, off-chain storage of sensitive information, and patient consent management.
Regulatory Compliance	Provides immutable audit trails, transaction traceability through blockchain streams, automated compliance enforcement via smart contracts, and support for regulatory monitoring and reporting.
Ethical Implications	Implements privacy-by-design principles, restricts access to authorized healthcare entities, improves transparency regarding data usage, and promotes accountability and trust.

Regarding data privacy, the permissioned architecture and fine-grained access controls ensure that sensitive patient information is accessible only to authorized entities, while off-chain storage and cryptographic mechanisms further safeguard confidentiality. In terms of regulatory compliance, the immutable ledger and structured streams provide transparent audit trails, facilitating monitoring, reporting, and enforcement of automated compliance policies, thereby supporting adherence to GDPR and HIPAA requirements. With respect to ethical considerations, the system embodies principles of privacy by design, responsible data stewardship, and patient autonomy by enabling explicit consent management and full traceability of data access. Collectively, these features demonstrate that IBEH effectively addresses the challenges of decentralized healthcare data management while ensuring security, regulatory compliance, and ethical governance.

6.3. Discussion and Limitations

The proposed IBEH solution was designed as an eHealth-oriented architectural framework, built on a permissioned multichain infrastructure. IBEH does not introduce a new blockchain protocol, a new consensus mechanism, or a new distributed ledger technology. Its contribution primarily concerns the design and implementation of a private blockchain that can be accessed by eHealth data management applications, including the possibility of using smart contracts. IBEH defines how healthcare applications interact with the blockchain network (in this case, a private blockchain) and introduces a processing flow structured according to a specific and precise workflow (described in Figure 4).

The innovation of IBEH must therefore be interpreted at the architectural and integration levels. This approach also ensures a clear functional separation of the various tasks. The IBEH layer manages healthcare applications and interactions with healthcare services. MultiChain, on the other hand, is responsible for managing the typical blockchain and distributed ledger operations. The intrinsic characteristics of MultiChain, such as the participation of nodes with specific authorization from the full node and the storage of transactions within the ledger with the possibility of maintaining a local copy on each node, provide the IBEH solution with a robust and tamper-proof layer. The choice of MultiChain as the underlying blockchain platform is based on the architectural and functional requirements of the IBEH implementation under consideration, rather than on the assumption that MultiChain is intrinsically superior to other blockchain technologies. with permission. Solutions such as Hyperledger Fabric, Quorum, or other private blockchains compatible with Ethereum or Corda are also well-established and robust distributed ledger frameworks. As cloud-based platforms, they can offer high processing speed and access control, but they can also suffer from excessive latency or limited customization. Multichain, on the other hand, being completely open source, is a blockchain that can be installed on

common COTS devices and can offer reduced latency and greater customization options. MultiChain was chosen because it offers a relatively lightweight approach to creating a private blockchain network and provides typical blockchain functionality through APIs and RPC commands that can be used to develop smart contracts and execute them within the Flask application environment adopted by the IBEH solution. Other blockchain types were also considered. Hyperledger Fabric is certainly a valid alternative thanks to its high computational performance and broad support for enterprise and multi-organizational environments. Fabric offers a complete architecture with particularly powerful features. These are valuable when organizations and companies require different organizational rules and a more complex implementation model. The IBEH architecture considered does not require the entire organizational model envisioned by Fabric, as its basic elements include access to the ledger by authorized nodes only, and the writing and retrieval of transaction information and its verifiability. Solutions like Quorum and those compatible with Ethereum are also valid alternatives to IBEH. Their most important advantage is the execution environment they offer, which is fully compatible with Ethereum and smart contracts. This feature is very useful when applications require very demanding on-chain processing. However, the IBEH architecture requires that typical blockchain functionality be left to MultiChain, and that the application logic and processing of healthcare services be managed at the application level. Therefore, a complete environment for the execution of complex smart contracts and compatible with EVM was not an essential functional requirement for IBEH, as it would have introduced additional unnecessary complexity. Corda is certainly another valid alternative, especially for environments complex and structured. Its architecture includes the definition of the blockchain state, smart contracts, workflows and notarization applications, etc. The IBEH architecture, on the other hand, adopts a distributed ledger interaction model that is much simpler because it does not require the full, workflow-based transactional model offered by Corda. Even in the case of Corda, its additional features were therefore not essential requirements. Consequently, the choice of MultiChain represents a good compromise between the features required by IBEH and the applications The proposed healthcare services and the complexity of the distributed ledger's underlying infrastructure. Despite the advantages offered by MultiChain, the proposed architecture also presents several limitations and necessary tradeoffs. A first limitation is IBEH's dependence on MultiChain. Although the application layer and blockchain layer are logically separate, IBEH relies on MultiChain for its entire blockchain component. It is dependent on MultiChain, and its possible replacement with another blockchain would require additional integration and adaptation of the interface, consensus mechanisms, application interfacing, etc. IBEH's modular structure reduces this dependence, but only at the application level; it cannot completely eliminate the dependence on the specific platform. A second limitation concerns the blockchain's programmability. MultiChain is certainly a valid solution for managing the healthcare data and transactions considered in IBEH, but it provides a development and execution environment that is less general in terms of programmability than platforms specifically designed for smart contracts or programmable chaincode. This situation represents a necessary compromise between a simple architecture and on-chain programmability. Scalability is also a crucial element for the blockchain under consideration. The growing number of nodes increases both the number of participants and the communication, synchronization, writing, and data validation capabilities. Furthermore, the local copy of the ledger among all nodes introduces an intrinsic overhead compared to centralized solutions, increasing resource consumption. When considering the different healthcare information systems, unavailability is a crucial element. IBEH decouples the healthcare application layer from the blockchain infrastructure layer, but it does not intrinsically solve all the interoperability

issues associated with different healthcare information systems, which may use different data formats, semantic structures and metadata, communication protocols, and consensus mechanisms. For example, interoperability with healthcare standards such as HL7 FHIR requires additional features that are not provided for in either IBEH or MultiChain. Finally, the comparison between MultiChain, Hyperledger Fabric, Quorum, and Corda presented in this paper should be interpreted as an architectural and technological comparison, rather than as evidence of the superiority of one platform over the other. The various solutions mentioned above differ not only in terms of performance, but also in terms of functionality and intrinsic features, and MultiChain represents the best compromise as a design solution to be considered in this paper.

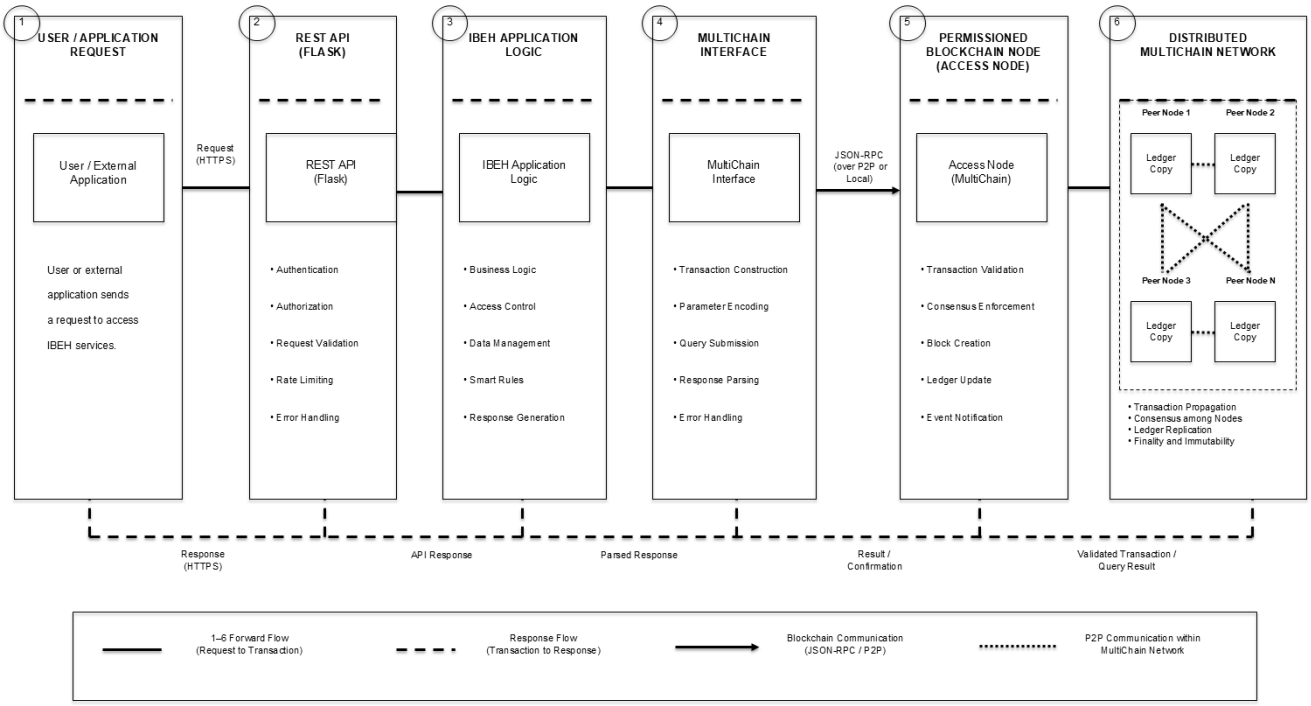


Figure 4. Functional Workflow the IBEH solution.

7. Development of IBEH Blockchain for eHealth

7.1. Overall System Architecture and Experimental Tests

Starting from the previous discussions concerning the application of multichain to the ehealth sector, in this Section we describe our implementation of a IBEH multichain-based system for the ehealth sector.

The considered Multichain-based system architecture consists mainly of four components, as shown in Figure 5. MultiChain is the private blockchain network, Pycharm is the Python editor, POSTMAN is the tool considered for the development of application programming interfaces (APIs) and Flask is used to implement the developed APIs. For the implementation of our IBEH solution the Python version 3.11 was selected [68]. This version supports the rapid development of small-scale web applications and enables testing in a local environment (localhost), providing a flexible and efficient development platform for blockchain-based eHealth solutions.

As mentioned, the IBEH solution is based on a distributed ledger architecture specifically designed and developed to reproduce the interaction between the application layer and the permissioned blockchain network, as summarized in Table 5.

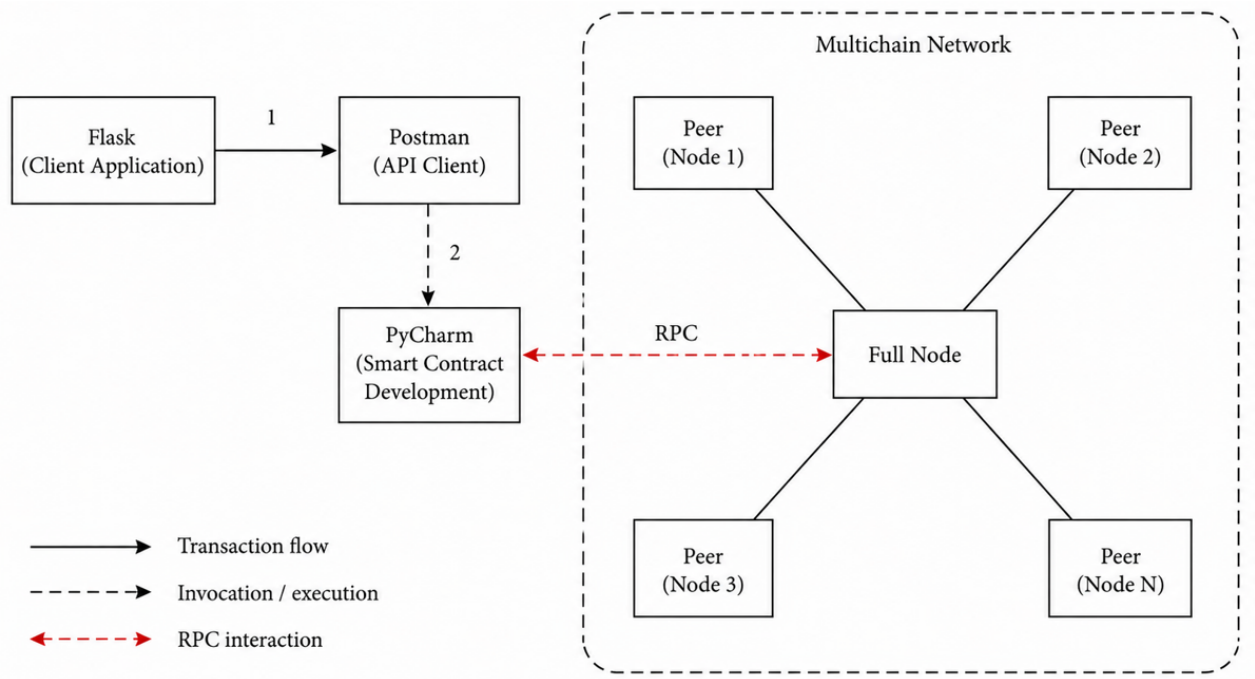


Figure 5. IBEH System components.

Table 5. Components and roles of the IBEH experimental test architecture.

Domain/Component	No.	Main Function
Client/Test Domain—API Client	2	Generates and submits healthcare application requests to the IBEH REST API.
IBEH Application Domain—Healthcare Application Host	4	Provides the IBEH application and healthcare-oriented processing logic.
IBEH Application Domain—Flask Web Server / REST API	4	Provides the entry point to IBEH and the REST endpoints; it manages the HTTP requests/responses.
IBEH Application Domain—IBEH Application Logic	4	Provides the analysis and processing of healthcare requests, and generates the required blockchain operation.
IBEH Application Domain—MultiChain Interface	4	It represents the link between the IBEH application logic to the MultiChain blockchain infrastructure.
Blockchain Domain—MultiChain Access Full Node	1	Provides the access point between the IBEH application domain and the permissioned MultiChain network and peers.
Blockchain Domain—Authorized MultiChain Peer Nodes	49 (emulated by 5 PC)	Join the IBEH blockchain and related functionalities.
Blockchain Domain—Permissioned MultiChain Network	50	Provides the distributed ledger infrastructure; it represents the link between the access full node and the authorized peer nodes.

The test architecture consists of four healthcare application hosts, two clients for generating API requests, a MultiChain access full node, and 50 MultiChain peer nodes that form the private blockchain network. The application hosts run the Python 3.11 implementation of IBEH along with the Flask web server, which contains the REST endpoints and acts as the entry point to the application. The application hosts also execute the processing logic and communicate with the MultiChain full node. Two clients are used to generate requests sent to the IBEH REST API. During the experimental tests, Postman was used to manage the external healthcare application and generate HTTP requests to the Flask web server. Each request generated by clients is received by the Flask web server and processed by the IBEH application, and then the next blockchain transaction is sent to MultiChain. The

blockchain infrastructure consists of a permissioned MultiChain network, composed of 50 blockchain nodes emulated by 5 PC. One node acts as an access point between the IBEH application and the blockchain network. The remaining nodes operate as authorized peers of the MultiChain full node. The access node receives blockchain transactions generated by IBEH and participates in the MultiChain peer-to-peer network along with the other authorized peers. Valid blockchain transactions are then checked and validated within the distributed ledger through MultiChain's mechanisms. From an implementation standpoint, the IBEH architecture used in the tests is therefore organized into three logical domains:

- the client/test domain, for generating application requests;
- the IBEH application domain, which contains the Flask web server's REST API, the application logic, and the MultiChain interface;
- the Blockchain domain, consisting of the MultiChain access full node and other authorized peers.

During the tests, the interaction between these components follows a precise sequence of steps: The client sends an HTTP request to one of the REST endpoints exposed by the Flask web server, which receives the request and forwards it to the corresponding function of the IBEH application. The IBEH generates the appropriate operation on the MultiChain blockchain and sends it via the blockchain interface to the access full node, which processes the operation. A blockchain transaction is generated and sent to all other authorized MultiChain peers. If the operation requires a response, the result is returned by the MultiChain full node to the IBEH application and then, subsequently, via the relevant REST API to the requesting client.

7.2. Smart Contract

IBEH implements and executes smart contracts at an application level specifically designed to interact with the MultiChain platform. Unlike other solutions that offer a native environment for smart contract development and execution, the conditional logic is developed in Python and executed within IBEH's application level. The Python modules were developed and tested using PyCharm and interact with MultiChain through Remote Procedure Calls (RPCs) and related APIs. Therefore, in the case of IBEH, smart contracts are built with programmable Python modules that implement the conditional logic controlling healthcare applications and provide the corresponding results in terms of operations on the MultiChain blockchain's distributed ledger. Thanks to the functional separation between the conditional logic at the application level and transaction management at the blockchain level, IBEH does not modify the native MultiChain protocol or introduce additional functionality. The smart contract modules were designed according to a modular approach that allows different functional categories to be associated with specific healthcare applications, as shown in the following Table 6.

The main functional categories include identity and authorization management for peers participating in the IBEH blockchain, medical records management, data processing consent and access management, and auditing and traceability. Identity and authorization functions allow a specific entity or peer to be authorized to perform a specific operation. Medical records management functions enable the creation, updating, retrieval, and verification of data associated with healthcare information and stored in the blockchain ledger. Consent and access management functions provide the rules to be applied for accessing patient data. Finally, auditing and traceability functions provide the methodology needed to store and verify operations performed on IBEH.

The healthcare application, if authorized, makes a request via the IBEH REST API. The Python smart contract module involved processes the request and verifies whether the required conditions are met. If so, the module generates the corresponding Multi-

Chain RPC command and sends it to the enabled MultiChain node, which processes the requested operation according to the established mechanisms. When a transaction is created, this node ensures it is propagated within the authorized peer-to-peer network. If the required conditions are not met, the operation is rejected by the contract logic at the application level.

Table 6. Design characteristics of the IBEH smart-contract modules.

Parameter	IBEH Design
Implementation language	Python 3.11
Development environment	PyCharm
Execution location	IBEH application layer
Blockchain interaction	MultiChain RPC interface
Underlying ledger	MultiChain private permissioned blockchain
Maximum module size	Approximately 600 LOC
Contract architecture	Modular
Main functions	Identity/authorization, healthcare record management, consent/access management, audit/traceability
Input processing	Validation before blockchain invocation
Authorization	Checked before the corresponding RPC operation
Blockchain state modification	Performed through MultiChain RPC operations
Main design principles	Modularity, Separation of Concerns, Access Control, Guard Conditions, Check-before-Execution, Controlled State Transition
Contract execution model	Off-chain/application-layer execution
Ledger execution model	Transaction processing and recording by MultiChain
Native on-chain smart contract	No
Transaction propagation	MultiChain permissioned P2P network

The implemented smart contract modules have a maximum size of approximately 600 lines of code (LOC), which indicates the implementation complexity. This upper limit was chosen to keep the software modules manageable, facilitating the readability and execution of the software code, and the related testing and verification operations. Therefore, the complexity of smart contract execution is moved to the IBEH application level. The contract modules verify whether the required conditions are met and manage the transactions before sending the MultiChain RPC operations.

In software engineering, recent approaches have emphasized the importance of implementing reusable mechanisms for processing and verifying transactions managed by smart contracts. Specifically, the AdapT solution provides a reusable package for building smart contracts that process transactions of congruent data types [69]. Additionally, CongruenT provides a methodology for reusing objects and classes for the conditional statements required for smart contract verification [70]. Both approaches emphasize modularity, separation between different transaction verification/processing functionalities, and the reuse of common conditional statements. IBEH is fully compatible with both approaches because it implements these reusability principles at the application level within its smart contract creation and management layer. Operations involving the distributed ledger remain delegated to MultiChain via RPC calls.

7.3. Proof of Authority (PoA)

MultiChain employs the PoA consensus mechanism, which relies on trusted entities, known as validators, to create new blocks based on reputation rather than computational power or stake. Originally proposed by Gavin Wood in 2017, PoA is commonly adopted in private blockchains, test networks, and development environments. Validators are pre-selected, and their identities are verified through mechanisms such as Know Your Customer (KYC), ensuring accountability and mitigating the risk of malicious activity. In PoA, validation is deterministic, with validators granted the exclusive right to propose new blocks according to a predefined schedule, thereby eliminating the need for computationally intensive processes characteristic of PoW. A primary advantage of PoA is its scalability and high performance. With a limited number of validators, PoA achieves high transaction throughput and rapid block confirmation—often under one second—making it suitable for applications requiring low-latency transaction processing. Furthermore, PoA is more energy-efficient than PoW, as it avoids the substantial computational demands associated with mining. However, PoA introduces a degree of centralization, as control is concentrated among a small set of validators. Network security is highly dependent on the integrity of these validators, and any compromise could potentially jeopardize the entire system [71]. Despite these considerations, PoA is particularly well-suited for private blockchain deployments and applications where trust among a limited set of participants is high, and where performance and energy efficiency are critical. A careful assessment of trade-offs between throughput, centralization, and security is essential when adopting PoA for blockchain networks.

7.4. Flask

In the development of this blockchain prototype, the Python microframework Flask [72] is employed to implement the APIs that enable communication between the blockchain and external clients or services. These APIs expose endpoints allowing users or systems to interact with the blockchain, including submitting new transactions, querying the ledger state, retrieving blocks, or initiating consensus procedures. Flask operates according to the Web Server Gateway Interface (WSGI) [73], a standard protocol that manages interactions between the web server and the Python application, ensuring HTTP requests are properly routed to the corresponding Python functions. Each API endpoint is defined as a Flask route, mapping a specific URL path and HTTP method (e.g., GET, POST) to a Python function that executes the relevant blockchain logic. For instance, a POST request to /transactions/new may trigger a function that validates a new transaction, appends it to the pending transaction pool, and returns a confirmation response. Similarly, a GET request to /chain could return the entire blockchain in JSON format. The Jinja2 template engine [74] is used primarily for rendering HTML responses when a web interface is required, although most APIs return structured JSON for programmatic access. Figure 6 illustrates the configuration of a basic Flask web server for a blockchain node, demonstrating how routes are registered, requests are processed, and responses are returned, thereby enabling seamless interaction between the blockchain, clients, other nodes, and external applications.

7.5. Postman

As this work focuses on a blockchain prototype, no graphical user interface has been developed. Interaction with the blockchain is achieved through Postman [75,76], a comprehensive API development environment. Postman enables developers to construct and send API requests, inspect both request and response data, and debug endpoints directly connected to the blockchain. Moreover, Postman supports multiple response formats,

including JSON, XML, HTML, and plain text, and offers integrated testing capabilities through JavaScript, allowing verification of API functionality and automated response validation. This approach provides a practical and efficient means of interaction with the blockchain prototype during development and testing. The /chain method is a "GET" type method for creating a JSON file with a copy of the Blockchain, as shown in Figure 7. It allows users to retrieve the full blockchain from the node.

```
#server creation
app=Flask
name
# generates a unique number for the current node node_identifier = str(uuid4())).replace('-', '")
#blockchain instantiation
blockchain = Blockchain
if name == "_main_':':
from argparse import ArgumentParser
parser = ArgumentParser()
parser.add_argument('-p','--port', default=8000, type=int, help='listening port')
args = parser.parse_args()
port = args.port
app.run(host='0.0.0.0', port-port)
```

Figure 6. Using Flask in Blockchain Prototype.

```
@app.route('/chain', methods=['GET']) def full_chain(): 2 response =
}
{
'chain': blockchain.chain, 'length': len (blockchain.chain),
(response), 200
```

Figure 7. Method /chain.

The /transactions/new method is a "POST" type method enabling the single node to receive new transactions from the actors interacting with the node itself and add them to a new block. The Figure 8 defines a Flask endpoint that allows clients to submit a new transaction to the blockchain node.

```
@app.routes ('/transactions / ne ne def neans:
values = request.get_son
# check for necessary parameters
required = ['id', 'channel', 'data','timestamp']
if not all (k in values for k in required):
msg = 'Missing parameters:
missing = list(set(required) - set (values)) for m in missing:
msg += m
if m != missing[-1]: msg += ',
return msg, 400
# creating a new transaction
Inde blockchainnew_transaction(values['id'], values['channel']. values ['data'].values ['timestamp'])
response = {'response' :f' Transaction will be added to block {inde} (response), 201
```

Figure 8. Method /transactions/new.

The transaction endpoint first validates the presence of all required attributes in the incoming request. These attributes include the actor's unique identifier, the channel through which the data was received, the data payload itself, and a timestamp indicating when the operation occurred. Only if all fields are provided does the system proceed to create a new transaction. Upon successful addition of the transaction to the node's pending transaction list, the client receives a confirmation message specifying the index of the block in which the transaction will eventually be recorded. For instance, using Postman, a transaction can be submitted via the HTTP request to an IP public address, e.g., 203.0.113.25, port 8000 and the "transactions" method. As an example, a sensor node measuring a patient's body temperature sends this information to the blockchain node. The node responds by indicating that the transaction will be included in block 3. Since block 3 has not yet been

mined, it is not yet appended to the blockchain. The subsequent mining operation is required to finalize the inclusion of the transaction within the blockchain.

The /mine endopint is a "GET" type method that allows the miner to mine the block and permanently add it to the Blockchain. The Figure 9 shows how to create a a Flask endpoint enabling the node to mine a new block in the blockchain.

```
@app.route ('/mine', methods=['GET'])
def mine():
# "proof" calculation by the algorithm of 'Proof of W ultimo_block = blockchain.last_block
proof = blockchain.proof_of_or
# creation of the transaction that identifies the node that mined
# the block, for this the channel and data fields are empty
blockchain. new_transaction(
id=node_identifier,
channel="" data=""
timestamp=time()
# add the new block to the chain
hash_previous = blockchain. hash (last_block) block = blockchain.new_block (proof, hash_previous)
# returns the block
response = {
}
'answer": "New block created".
'index': block ('index'].
'transactions': block ['transactions'],
'proof': block ['proof'],
'hash_previous": block ['hash_previous'],
(response), 200
```

Figure 9. Method /mine.

According to the hash of the previous block, when the block is mined ensuring, it will be added to the blockchain. The similar HTTP call via Postman is performed to the same IP public address, e.g., 203.0.113.25, port 8000 and the "mine" method.

The /nodes/register method is a "POST" type method, which allows the user to register a new node on the Blockchain and have it communicate with an existing node, as shown in Figure 10.

The node registration endpoint enables a new blockchain node to join the existing network and obtain the list of all currently connected nodes. For example, a third node running on port 8090 can register with an existing node on port 8000 through the /nodes/register endpoint. Upon successful registration, the new node retrieves a copy of the blockchain using the /chain endpoint and ensures synchronization with the network by invoking the /nodes/resolve method. This method applies the consensus algorithm to validate and reconcile blocks, guaranteeing that the node maintains the most up-to-date and consistent version of the blockchain across the distributed network. The /nodes/resolve method is a "GET" type method, which allows the node to update the blockchain with the nearby nodes. The Figure 11 defines a Flask endpoint that allows a blockchain node to resolve conflicts and achieve consensus with its neighbors.

The /nodes/resolve endpoint is responsible for conflict resolution by comparing the blockchain lengths of neighboring nodes. When a longer valid chain is identified, the node replaces its current chain with the longer one; if no longer chain exists, the existing chain is maintained as valid. For instance, node 8090 can invoke this endpoint to the same IP public address, e.g., 203.0.113.25 and the "resolve" method to update its local blockchain with the most recent valid chain from its peers, thereby ensuring consistency and integrity across the distributed network.

```
@app.route ("/nodes / register', methods=['POST'])
def register_nodes():
    values = request.get_son
    node = values.get('node')
if node is None:
    return "(err) provide a valid node address", 400
blockchain. registration_new_node (node)
nodes = list (blockchain.node)
nodi_tx
for node in blockchain.node:
    nodi_t +
    if node != nodes[-1]:
        nodi_t +
    response = {
        'answer': 'Added node',
        'nodes': nodi_tx
    }
(response), 201
```

Figure 10. Method /register.

```
@app.routes ('/nodes/resolve', methods=[GET])
def consensus):
replaced = blockchain. algorithm_for_consense()

if replaced:
response = {
'answer': 'Replaced current chain with that of neighbors',
'new chain': blockchain.chain
}
Elsa:
response = {
'answer: Valid current chain',
'chain': blockchain.chain
}
(response), 200
```

Figure 11. Method /resolve.

7.6. Blocks, Transactions and Consensus Mechanism

7.6.1. Blocks

MultiChain provides PoA as a consensus mechanism. It is based on a limited number of validators responsible for block creation. The Figure 12 is responsible for creating and adding a new block to the blockchain.

```
def new_block(self, proof, hash_previous):

block = {
    'Indexchain) + 1,
    'timestamp': time(),
    'transactions': self.new_transactions,
    'proof: proof,
    'hash_previous': hash_previous or self.hash(self.chain [-1]),
}

# cleaning transaction list to store
self.new_transactions = []
self.chaln.append (block)
return block
```

Figure 12. New Block functions.

7.6.2. Transactions

This function allows the user to create a new transaction once the data has been received and returns the index of the block that will contain this information, as shown in Figure 13.

```
def new_transaction(self, id, channel, data, timestamp):

self.new_transactions.append({
"id': id,
'channel': channel,
'data': data,
'timestamp': timestamp)
})

return self.last_block ['index']
```

Figure 13. New Transaction functions.

A transaction is composed of the following fields:

- id: This is the identifier of the person who generated the message
- channel: This is the channel used to send the message, it can be for example wifi, ethernet or Bluetooth.
- data: This is the actual information
- timestamp: This is the date and time of the operation

7.6.3. Consensus Mechanism

As mentioned in the other Sections, the PoA Consensus Mechanism allows to confirm and validate transactions to produce new blocks of the chain, as shown in Figure 14.

```
def proof_of_authority(self, last_block, node_id=None):
    proof = node_id or self.node_identifier  # the node issuing the block
    return proof
```

Figure 14. Proof of Authority.

The proof_of_authority function determines the "proof" of a block in the PoA consensus mechanism. Unlike PoW, PoA does not involve solving a computational challenge; instead, the proof corresponds to the identifier of the authorized node generating the block. The node identifier, represented as node_id or self.node_identifier, serves as the proof, although a digital signature could alternatively be employed to authenticate the block. By returning this proof, the block is validated within the network, as only authorized nodes are permitted to generate blocks. This approach replaces the PoW mechanism, allowing blocks to be created efficiently and securely based on node authorization.

7.6.4. Hash

The hash function uses the hashlib library. It takes a block and transforms it into a string using the JSON library. This string is then encoded using the SHA-256 algorithm to obtain a unique key, as shown in Figure 15.

```python
def hash(block):
    # Dictionary sorting
    block_string = json.dumps(block, sort_keys=True).encode()
    return hashlib.sha256(block_string).hexdigest()
```

Figure 15. The Hash Function.

The main scope of the implemented Smart Contract is to insert data into an EHR on a MultiChain blockchain using RPC, as described in the following Figure 16.

```python
import requests
import json
# --- RPC CONFIGURATION FOR PEER WITH PUBLIC IP ---
rpc_user = "multichainrpc"        # RPC username configured on the remote node
rpc_password = "password"          # RPC password configured on the remote node
rpc_host = "203.0.113.25"          # Public IP of the peer node
rpc_port = "8332"                  # RPC port configured on the remote node
url = f"http://{rpc_host}:{rpc_port}"
# Generic function for RPC calls
def multichain_rpc(method, params=None):
    headers = {'Content-Type': 'application/json'}
    payload = json.dumps({
        "method": method,
        "params": params or [],
        "id": 1
    })
    response = requests.post(url, headers=headers, data=payload, auth=(rpc_user, rpc_password))
    return response.json()
# Function to verify EHR data insertion
def verify_ehr_data_insertion(stream_name, key, expected_value):
    response = multichain_rpc("liststreamkeyitems", [stream_name, key])
    if 'result' in response and response['result']:
        latest_data = response['result'][-1]['data']['hex']
        decoded_data = bytes.fromhex(latest_data).decode('utf-8')
        print(f"Latest data for {key}: {decoded_data}")
        if decoded_data == expected_value:
            print("Data insertion verified successfully.")
        else:
            print("Data verification failed: Expected value does not match.")
    else:
        print("No data found for the specified key.")
```

Figure 16. Smart Contract for EHR insertion.

The provided Python script illustrates interaction with a MultiChain node via Remote Procedure Call (RPC), enabling secure and controlled access to a permissioned blockchain. The MultiChain_rpc function constructs JSON payloads for method calls with associated parameters and sends them to the remote node, returning responses in JSON format. This modular design allows for consistent invocation of any RPC method, supporting extensibility and maintaining a clear separation of concerns. The verify_ehr_data_insertion function implements the core verification logic by retrieving the most recent value associated with a specified key from a designated stream. Hexadecimal data retrieved from the stream is decoded into a human-readable format and compared against the expected value to verify correct insertion. The script leverages MultiChain streams and unique keys for each patient record to enable efficient retrieval of the latest data while maintaining fine-grained access control over sensitive health information. Security is enforced through RPC authentication, restricting access to authorized nodes. Additionally, encoded storage combined with SHA-256 hashing ensures data integrity and immutability, mitigating the risk of inadvertent disclosure. Validation processes are performed externally in Python, enabling real-time verification of retrieved values against expected results, thus ensuring both the integrity and traceability of patient records.

8. Performance Analysis

In this Section we evaluate the performance of the considered implementation of the IBEH solution both from the network and blockchain point of views. Table 7 summarizes the performance characteristics of four representative blockchain platforms for eHealth (Ethereum, Hyperledger Fabric, and Quorum IBFT) compared with IBEH, in terms of transaction throughput (TPS) and transaction latency. The reported values were collected from scientific literature and studies.

Table 7. Comparison of transaction throughput and latency for representative blockchain platforms.

Blockchain	TPS Min	TPS Max	Lat. Min	Lat. Max	Nodes Considered	Reference
Ethereum	15	30	12 s	15 min	Public mainnet (thousands of nodes)	[77,78]
Hyperledger Fabric	140	2250	∼1 s	∼4 s	15 cloud servers	[79]
Quorum IBFT	∼300	∼900	0.4 s	1.0 s	4 validator nodes	[80]
Corda	∼26	∼1000	0.3 s	6.0 s	Permissioned network	[81,82]
IBEH	157	495	0.20 s	2.15 s	50 nodes, 500 smart contracts	Proposed solution (Multichain) [83]

The performance of the IBEH solution was experimentally evaluated using the test architecture described in Section 7.1. Transaction latency and throughput in terms of Transactions per Second (TPS), selected as the main KPIs of the blockchain infrastructure, were primarily evaluated. Transaction latency indicates the time required to finalize a transaction on the ledger, considering the time elapsed between its submission by the IBEH application and its confirmation on the ledger by the MultiChain blockchain. Throughput indicates the number of transactions successfully executed and confirmed per unit of time. These KPIs were measured by varying the number of blockchain nodes, the number of node logins, and the complexity of the application-level smart contract modules in terms of LoC. To provide a more comprehensive comparative evaluation, the experimental results obtained by the IBEH solution were compared with the performance metrics of other blockchains, such as Ethereum, Hyperledger Fabric, Quorum IBFT, and Corda, which are known in the literature and summarized in Table 7. These platforms were not implemented in the IBEH test environment. The latency and throughput values were obtained from the experimental studies and technical sources cited in Table 7. To better compare performance, the performance ranges that each platform can achieve in terms of minimum and maximum values have been reported. The comparison in Table 7 therefore provides a comparative evaluation containing some known references in the literature of established blockchain platforms to which the performance achievable by the IBEH solution refers.

The minimum and maximum TPS values represent the range of transaction processing capabilities observed under the experimental conditions described in the corresponding references. Similarly, the minimum and maximum latency values indicate the time required to validate and commit transactions, ranging from the initial transaction inclusion to final confirmation depending on the blockchain architecture and consensus protocol.

It is important to note that the reported results were obtained under different network configurations and workloads. Ethereum measurements refer to the public mainnet environment, which consists of thousands of geographically distributed nodes and therefore reflects real-world operating conditions. In contrast, Hyperledger Fabric and Quorum IBFT were evaluated in controlled experimental environments with a limited number of nodes, specifically 15 cloud servers and 4 Quorum validator nodes, respectively. Consequently, the

numerical values reported in Table 7 should be interpreted as representative performance indicators rather than directly comparable measurements. The results obtained from the simulation campaign are generally consistent with the performance ranges reported in Table 7. In particular, the latency values generated for Ethereum range from approximately 12 s to more than 350 s, which closely matches the transaction confirmation and finality intervals reported in the literature. This behavior reflects the characteristics of public blockchain networks, where a high degree of decentralization and a large number of participating nodes lead to relatively high transaction confirmation times. Similarly, the simulated Hyperledger Fabric latency values fall within the interval of 1–4 s reported by Thakkar et al., confirming the ability of permissioned blockchain architectures to provide significantly lower transaction delays than public blockchain platforms. The obtained results therefore validate the adopted simulation model and demonstrate its consistency with experimentally observed performance data. For Quorum IBFT, the simulated latency values are concentrated between approximately 0.2 s and 0.4 s. Although these values are slightly lower than the upper bound reported in the benchmark study by Baliga et al. (0.4–1.0 s), they remain of the same order of magnitude and confirm the low-latency behavior of Byzantine Fault Tolerant permissioned blockchain systems. The discrepancy can be attributed to the simplified assumptions adopted in the simulation environment and the absence of additional communication overheads typically observed in real deployments. The proposed IBEH solution shows latency values ranging from approximately 0.2 s to 2.15 s and transaction throughput values between 157 TPS and 495 TPS. These results place the proposed architecture in an intermediate position between highly scalable enterprise blockchain platforms and public blockchain networks. Compared with Ethereum, the proposed solution achieves substantially lower transaction latency while maintaining significantly higher throughput. Furthermore, although Hyperledger Fabric and Quorum IBFT may achieve higher peak throughput under specific benchmark conditions, the proposed IBEH architecture provides a favorable trade-off between performance, decentralization, and operational flexibility. It should be emphasized that the values reported in Table 7 were obtained from different experimental studies employing heterogeneous network configurations and workloads. In contrast, the results presented in this work were generated under a common simulation environment consisting of 50 nodes and 500 smart contracts. Each node can run at maximum 100 smart contracts. Therefore, the comparison should not be interpreted as a direct replication of the original benchmark experiments, but rather as a comparative evaluation under a unified and controlled operating scenario. Despite these differences, the observed trends are consistent with the performance characteristics reported in the literature and provide evidence of the effectiveness of the proposed IBEH architecture.

8.1. Blockchain Performance Analysis

This section provides the analysis of the smart contracts' performance as a function of the Number of Nodes (NoN) and Lines of Code (LoC), offering insights into the scalability and efficiency of blockchain-based systems.

The experimental analyzes were conducted on a MultiChain network consisting of 50 nodes, serving as a decentralized overlay for the management and exchange of healthcare data. Each node executed approximately 100 smart contracts, with lengths ranging from 10 to 600 LoC. These smart contracts were designed to manage access to EHRs stored within hospital medical records and healthcare information systems. Integration of these records was facilitated through a federated cloud architecture, ensuring distributed data storage across multiple cloud environments while maintaining security, privacy, and interoperability standards. The blockchain network, implemented as a distributed overlay,

enables seamless and secure interaction among heterogeneous healthcare systems and stakeholders. The distributed ledger guaranties data integrity and traceability, supporting dynamic sharing of healthcare information among authorized participants. Access to the network is granted to a range of users, including patients, physicians, laboratory technicians, and other healthcare professionals, allowing secure and efficient retrieval and management of EHRs. Using this blockchain-based infrastructure, the system improves data accessibility, promotes collaborative healthcare management, and ensures compliance with regulatory requirements regarding data privacy and security. The considered metrics are:

- Transaction Latency
- TPS

The transaction latency in a private blockchain with PoA can be defined as:

$$\mathrm{L} = T_c - T_s \tag{1}$$

where:

- T_s is the timestamp when the transaction is sent,
- T_c is the timestamp when the transaction is confirmed.

Since PoA blockchains have pre-authorized validators and do not rely on computationally expensive proof-of-work mechanisms, the average transaction latency can be approximated as:

$$\mathrm{L} \approx T_p + T_b \tag{2}$$

where:

- T_p is the transaction propagation time over the network,
- T_b is the average block generation time.

As T_b is predefined and centrally regulated in PoA, the transaction latency is much more predictable compared to PoW-based blockchains. In a PoA blockchain TPS is generally higher compared to PoW because the consensus mechanism is faster and block time is deterministic. The TPS can be defined as:

$$\mathrm{TPS} = \frac{N_t}{T} \tag{3}$$

where:

- N_t is the total number of confirmed transactions within a given time window T,
- T is the measurement duration (in seconds).

Since blocks in PoA are generated at fixed intervals T_b, we can estimate TPS as:

$$\mathrm{TPS} = \frac{B}{T_b} \tag{4}$$

where:

- B is the maximum number of transactions per block,
- T_b is the average block generation time.

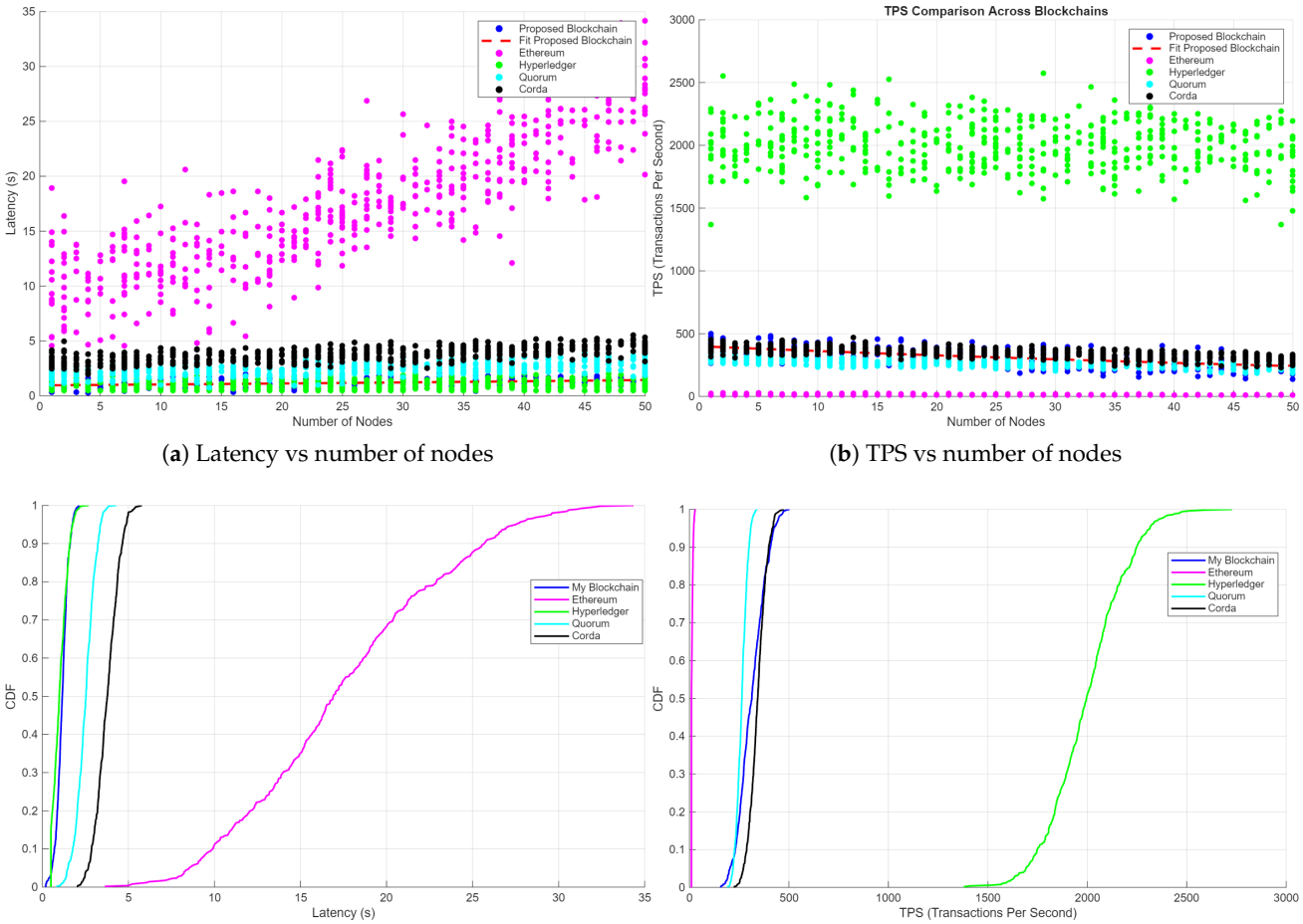
If we consider a system with N_v validators processing transactions in parallel, we obtain a refined estimate of effective TPS:

$$\mathrm{TPS} = \frac{N_v \cdot B}{T_b} \tag{5}$$

where:

- N_v is the number of active validators.

Figure 17 illustrates the IBEH performance compared between IBEH and the other considered blockchains for eHealth in terms of latency and TPS when NoN is increased.



(**a**) Latency vs number of nodes

(**b**) TPS vs number of nodes

(**c**) Latency CDF comparison

(**d**) TPS CDF comparison

Figure 17. Performance comparison of the proposed IBEH solution with Ethereum, Hyperledger, Quorum, and Corda blockchains. (**a**) Latency as a function of the number of nodes. (**b**) Throughput (TPS) as a function of the number of nodes. (**c**) Cumulative distribution function (CDF) of latency. (**d**) CDF of throughput (TPS).

Figure 17a shows the impact of NoN increase on the blockchain performance. Ethereum consistently demonstrates the highest latency across networks of varying sizes. Even for relatively small networks comprising one to ten nodes, observed delays average approximately 5.3 s, while larger networks with fifty nodes can experience latencies exceeding 34 s, reflecting the inherent overhead introduced by its PoW and associated consensus mechanisms. In contrast, Hyperledger exhibits consistently low and stable latency, typically ranging from 0.5 to 2 s, largely independent of NoN. This performance underscores the efficiency of its permissioned architecture and consensus protocols, such as PBFT. Quorum presents intermediate latency characteristics, generally between 1 and 3 s, with only moderate increases as the network scales, indicating suitability for enterprise applications requiring privacy alongside moderate throughput. Corda demonstrates moderate latency, usually spanning 2 to 5 s, with sensitivity to network size positioned between Ethereum and Hyperledger, reflecting its partially peer-to-peer consensus model that balances transaction finality with network reliability. Collectively, these observations indicate that latency escalates with increasing NoN for

Ethereum and, to a lesser extent, Quorum and Corda, whereas Hyperledger remains largely unaffected, highlighting its superior scalability and operational efficiency in permissioned blockchain environments. Figure 17b illustrates TPS measured across various blockchain platforms and differing network sizes. The results reveal substantial variations in throughput depending on both the platform architecture and the number of nodes. Ethereum consistently demonstrates the lowest TPS values, typically ranging from 10 to 20 TPS, reflecting the limitations imposed by its consensus mechanisms and public network constraints. In contrast, Hyperledger achieves markedly higher throughput, frequently surpassing 2000 TPS across multiple network configurations, highlighting the efficiency of its permissioned architecture and parallelizable consensus protocols. Quorum and Corda exhibit intermediate performance, with TPS generally between 200 and 400, reflecting their trade-offs between privacy, consensus overhead, and network scalability. These observations underscore the pronounced differences in throughput capabilities between public and permissioned blockchain platforms and illustrate the impact of network size and consensus design on overall transaction performance. Across varying network sizes, the observed TPS exhibit non-linear trends, indicating that an increase in the NoN does not necessarily correspond to improved performance. In certain cases, TPS may even decline, particularly for public platforms such as Ethereum and for Corda, which is sensitive to network expansion. By contrast, Hyperledger demonstrates robust scalability, consistently maintaining high TPS across different NoN configurations, reflecting the efficiency of its permissioned architecture and endorsement-based consensus. Corda shows occasional performance peaks at very low node counts, yet its throughput generally remains below that of Hyperledger. Quorum exhibits relatively stable performance, though typically achieving lower TPS than Hyperledger and occasionally lower than Corda. Collectively, these results underscore the pronounced disparities in scalability and throughput between permissioned blockchain platforms, such as Hyperledger, Quorum, and Corda, and public platforms like Ethereum, highlighting the critical influence of consensus mechanisms and network topology on overall performance. A good fitting model for IBEH is represented by the following equation:

$$y = a \cdot \exp(b \cdot x) \tag{6}$$

with:

$$a = 0.84, \quad b = 0.025 \tag{7}$$

The model provides a Coefficient of Determination is

$$R^2 = 0.90 \tag{8}$$

This observed trend suggests that, within distributed networks, increasing the NoN beyond a critical threshold substantially amplifies communication and synchronization overhead. The consequent quadratic growth in latency arises from the increasing complexity associated with achieving consensus and maintaining data consistency across the network. Although a larger node count enhances decentralization and network security, it concurrently introduces performance bottlenecks. These findings underscore the imperative for more efficient consensus mechanisms and optimized network protocols to sustain scalability and maintain high transaction throughput in large-scale blockchain deployments.

According to Figure 17b the TPS performance exhibits significant variability depending on both the underlying blockchain platform and the network size. Specifically, Ethereum consistently demonstrates lower TPS values relative to other platforms, typically ranging from 10 to 20 TPS. In contrast, Hyperledger achieves substantially higher throughput, frequently surpassing 2000 TPS across multiple network configurations. Quorum and Corda display intermediate performance, with TPS generally falling between 200 and 400,

reflecting their trade-offs between privacy, consensus overhead, and scalability. Across varying network sizes, the observed TPS trends are not strictly linear. An increase in NoN does not necessarily translate into improved performance and, in certain cases, may even result in reduced TPS, particularly for platforms such as Ethereum and Corda. Conversely, Hyperledger demonstrates greater resilience to variations in network size, consistently maintaining high TPS across different configurations. Corda exhibits occasional performance spikes at very low NoN, yet its TPS generally remains below that of Hyperledger. Quorum maintains relatively stable performance but typically achieves lower TPS than Hyperledger and, in some instances, lower than Corda. Overall, these observations underscore the substantial differences in scalability and throughput between permissioned platforms, such as Hyperledger, Quorum, and Corda, and public platforms like Ethereum. Also, in this case, a good fitting model for TPS is represented by an exponential equation:

$$y = a \cdot \exp(b \cdot x) \tag{9}$$

with:

$$a = 269.21, \quad b = -0.019 \tag{10}$$

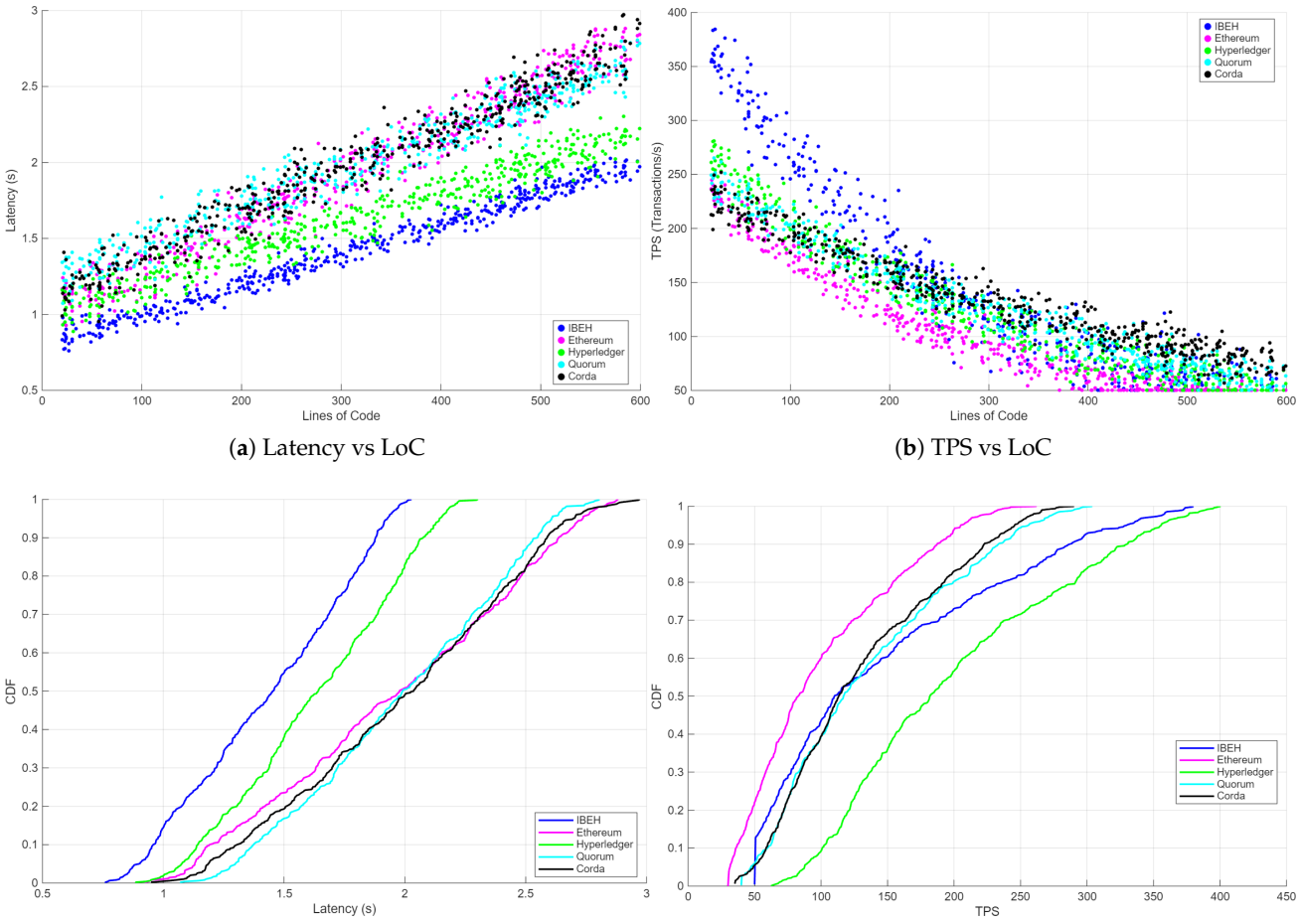
The model provides a Coefficient of Determination is

$$R^2 = 0.89 \tag{11}$$

Beyond a threshold, the coordination overhead among multiple nodes outweighs the benefits of parallel processing, resulting in a decline in TPS. This phenomenon reflects a well-documented scalability limitation in blockchain systems, where an increasing number of nodes can induce network congestion, higher computational demands, and slower transaction validation. Although adding nodes improves resilience and decentralization, it does not inherently improve performance. To address this reduction in TPS, approaches such as sharding, Layer-2 scaling solutions, and adaptive consensus algorithms are recommended. Figure 17c,d show the CDF of latency and TPS vs. NoN, respectively. Most transactions exhibit latencies below two seconds; however, a long-tail distribution indicates occasional high delays attributable to factors such as network congestion, computational load, and consensus overhead. Similarly, although a substantial fraction of transactions achieve high throughput, the presence of a long tail of lower TPS values suggests performance degradation under suboptimal conditions. These observations underscore the importance of dynamic resource allocation, congestion control, and optimizations such as transaction batching, enhanced block propagation, and parallel processing to ensure consistent performance in blockchain networks. As shown in Figure 17d, the CDF of TPS shows that a considerable portion of transactions achieve high throughput, yet there is a long-tail distribution with lower TPS values. Figure 18 shows the Smart Contract performance comparison of IBEH and the other considered blockchains for eHealth in terms of latency and TPS depending of number of LoC.

Figure 18c presents the CDF of transaction latency across different platforms in terms of LoC. The IBEH curve is left-shifted, indicating faster execution, with 90% of smart contracts completing within 1.2 s. Ethereum exhibits the highest latency, with 90% of contracts requiring up to 2.3 s. Hyperledger demonstrates consistently low latency, approximately 1.1 s at the 90th percentile, whereas Quorum and Corda occupy intermediate positions, around 1.3–1.4 s. These results underscore IBEH's superior efficiency and predictability in executing smart contracts. Figure 18d illustrates the CDF of TPS across different blockchain platforms in terms of LoC. IBEH outperforms all other platforms, achieving a median TPS of approximately 380 and a 90th percentile of roughly 420. Ethereum exhibits the lowest

throughput, with a median around 120 TPS, while Hyperledger maintains consistently high throughput, with a median near 400 TPS. Quorum and Corda demonstrate moderate performance, with median TPS values ranging between 220 and 250. The CDF highlights IBEH's capacity to sustain high transaction rates, even for increasingly complex smart contracts.



(**a**) Latency vs LoC (**b**) TPS vs LoC

(**c**) Latency CDF vs LoC (**d**) TPS CDF vs LoC

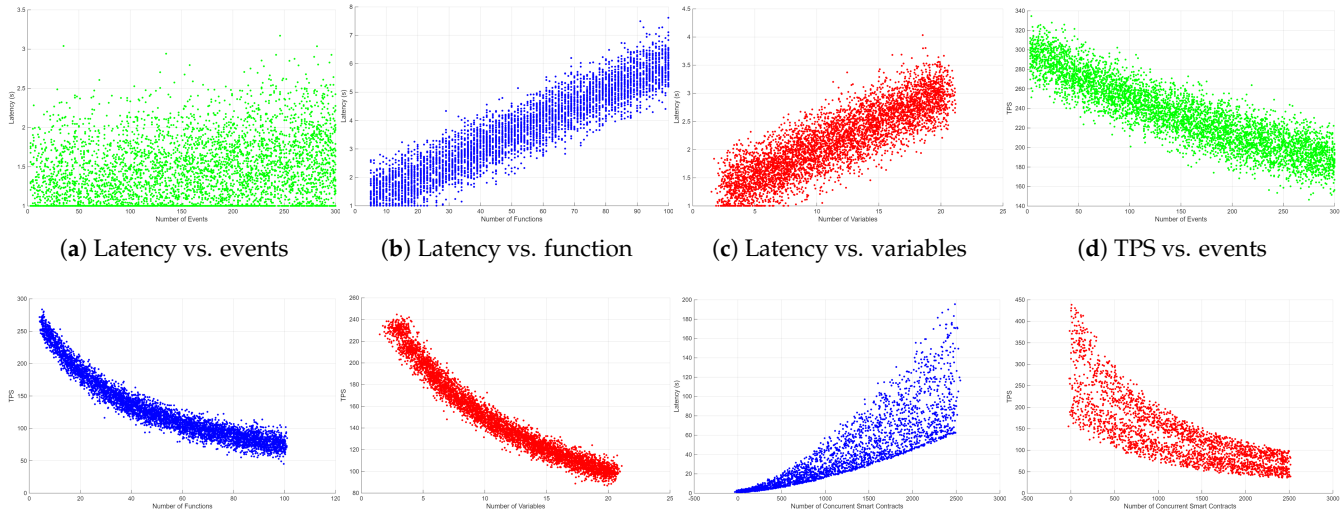
Figure 18. Smart contract performance in terms of latency and throughput (TPS) as a function of the number of LoCs. (**a**) Latency vs. LoC. (**b**) TPS vs. LoC. (**c**) Latency Cumulative distribution function (CDF) vs. LoC. (**d**) TPS CDF vs. LoC.

8.2. Smart Contract Performance Comparison

The following Figure 19 show the performance of smart contracts in terms of latency and TPS varying their complexity (number of events, variables, and functions) and the concurrent access.

As shown in Figure 19a–c, regarding latency, it is evident that the number of functions exerts the most significant impact, with a mean latency of 3.56 seconds, a maximum of 6.42 s, and a 90th percentile of 4.80 s. In contrast, the number of variables and events has a comparatively smaller effect on latency, with mean values of 2.45 s and 2.03 s, respectively, and correspondingly lower maxima and 90th percentiles. These results indicate more stable execution times when the number of variables or events increases. Notably, all latency metrics exhibit a minimum of 1 s, representing the baseline processing time for smart contracts under minimal computational load. As shown in Figure 19d–f, Throughput trends exhibit an inverse relationship with contract complexity. Execution of contracts with a higher number of events results in the highest mean TPS (150.8), a maximum of 220.1, and a 90th percentile of 190.2, indicating relatively efficient handling of event-heavy contracts. Conversely, an increased number of func-

tions reduces throughput, with a mean of 140.2 TPS and a maximum of 210.3, reflecting the processing overhead associated with more complex functional logic. The number of variables exerts the most pronounced impact on reducing throughput, with a mean of 85.3 TPS and a maximum of 130.6, highlighting that contracts with numerous variables can significantly constrain system performance. Figure 19g–h give a graphical representation of blockchain performance under concurrent smart contract execution. In the "Concurrent Access" scenario, the observed latency exhibits a minimum of 2.01 s, a mean of 89.03 s, a maximum of 455.34 s, and a 90th percentile of 205.21 s. These results indicate a pronounced increase in transaction confirmation time as the number of simultaneous smart contract executions grows. Correspondingly, the throughput (TPS) demonstrates a minimum of 14.09 TPS, a mean of 62.04 TPS, a maximum of 237.23 TPS, and a 10th percentile of 111.56 TPS, reflecting a clear degradation in transaction processing capacity under high concurrency. Collectively, these findings quantitatively underscore the impact of concurrent access on blockchain performance, highlighting substantial variability in both latency and throughput, and revealing potential bottlenecks when multiple smart contracts are executed simultaneously.



(**a**) Latency vs. events (**b**) Latency vs. function (**c**) Latency vs. variables (**d**) TPS vs. events

(**e**) TPS vs. functions (**f**) TPS vs. variables (**g**) Latency vs. access number (**h**) TPS vs. access number

Figure 19. Smart contract performance in terms of latency and throughput (TPS) under varying complexity parameters.

9. Conclusions

The IBEH solution allows for the separation of the application layer from the blockchain layer. This architectural approach also enables the development of a private blockchain infrastructure that is completely flexible, customizable, and easily deployable. Interaction between healthcare applications (e.g., smart contracts) and the blockchain network is enabled by REST APIs and RPC calls. The smart contracts implement the conditional logic for granting, denying, or revoking authorizations, along with the application features and policies required for managing and sharing healthcare data. Thanks to its modular architecture, the IBEH solution integrates blockchain functionality into healthcare applications without strictly tying the application logic to the underlying blockchain infrastructure. From a performance point of view, the analysis of the six performance graphs provides critical insights into the trade-offs between scalability, efficiency, and complexity in blockchain networks. The results demonstrate that increasing NoN enhances decentralization and security; however, this improvement comes at the expense of higher latency and reduced transaction throughput due to increased communication overhead

and consensus delays. Additionally, the complexity of smart contracts, quantified in terms of LoC, significantly affects network performance. More complex contracts incur greater computational overhead, resulting in longer execution times and decreased TPS, thereby highlighting the importance of optimized contract design. To sustain high performance, blockchain networks should incorporate advanced scalability solutions, including sharding, layer-2 rollups, and optimized consensus mechanisms, which can mitigate the observed efficiency losses. Furthermore, efficient smart contract development practices—such as minimizing redundant computations and leveraging off-chain processing—are crucial for maintaining throughput while preserving security and functionality. These findings carry substantial implications for domains reliant on blockchain technology, particularly the eHealth sector. Given the sensitivity and high volume of medical data transactions, achieving low latency and high TPS is essential for applications such as real-time patient monitoring, secure EHR sharing, and telemedicine services. By optimizing blockchain scalability and smart contract efficiency, eHealth systems can enable faster data processing, enhanced security, and improved interoperability among healthcare providers, thereby fostering a secure and efficient digital healthcare ecosystem that supports improved patient outcomes and operational performance. Future work on the proposed MultiChain-based permissioned blockchain for eHealth can be extended along several complementary directions. Advanced access control mechanisms, including attribute-based or role-based policies, could provide finer-grained authorization for diverse healthcare stakeholders. Smart contracts can be enhanced to automate complex workflows, such as consent management, data-sharing agreements, and insurance claim verification. Integrating data provenance and auditing tools would further enhance transparency, ensuring traceability of all modifications and access events to patient records. As medical data volumes continue to expand, scalability remains a critical concern. Layered approaches, such as sidechains or off-chain storage, can accommodate large datasets while maintaining on-chain hash integrity, whereas sharding or partitioning the blockchain by institution or data type can increase throughput and reduce latency. Consensus optimizations— e.g., lightweight PBFT variants tailored to permissioned networks—can further improve performance without compromising security. Beyond these enhancements, the blockchain can serve as a secure infrastructure for AI-enabled healthcare applications. Federated learning frameworks may be integrated to enable distributed model training across multiple hospitals without exposing sensitive patient data, with the blockchain ensuring auditability of model updates. AI-driven diagnostic systems can leverage verified historical data stored on-chain to improve predictive accuracy, support personalized medicine, and detect anomalies in real time. Moreover, integration with IoT health sensors can provide continuous, reliable data feeds for AI models, enhancing remote patient monitoring and enabling more proactive care. Collectively, these advancements position the proposed blockchain framework as a robust, scalable, and future-ready platform for secure, AI-assisted healthcare systems.

Author Contributions: Conceptualization, A.V. and F.M.; methodology, A.V. and F.M.; software, A.V.; validation, F.M.; formal analysis, A.V. and F.M.; investigation, A.V. and F.M.; data curation, A.V.; writing—original draft preparation, A.V. and F.M.; writing—review and editing, A.V. and F.M.; visualization, F.M.; supervision, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Department for Digital Transformation under the Italian Government—Presidency of the Council of Ministers within the Edge Cloud Computing (ECC) Field Trials Initiative—E-CLONET research project (CUP: E82E26000030001).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

dApp	Decentralized Application
EHRs	Electronic Health Records
HMS	Health Management Systems
TEEs	Trusted Execution Environments
DLTs	Distributed Ledger Technologies
EEA	Ethereum Enterprise Alliance
ID	Digital Identity
MSP	Membership Service Provider
MVCC	Multi-Version Concurrency Control
ICOs	Initial Coin Offerings
NFTs	Non-Fungible Tokens
HIPAA	Health Insurance Portability and Accountability Act
IoT	Internet of Things
GDPR	General Data Protection Regulation
BFT	Byzantine Fault Tolerance
CFT	Crash Fault Tolerance
KSI	Keyless Signature Infrastructure
PoW	Proof of Work
PoS	Proof of Stake
PoA	Proof of Authority
TPS	Transactions per Second
PBFT	Practical Byzantine Fault Tolerance
RBAC	Role-Based Access Control
ZKP	Zero-Knowledge Proof
Diff Privacy	Differential Privacy
EVM	Ethereum Virtual Machine
APIs	Application Programming Interfaces
KYC	Know Your Customer
WSGI	Web Server Gateway Interface
HTTP	HyperText Transfer Protocol
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
HTML	HyperText Markup Language
SHA	Secure Hash Algorithm
RPC	Remote Procedure Call
NoN	Number of Nodes
LoC	Lines of Code
CDF	Cumulative Distribution Function

References

1. Seo, J. The Future of Digital Authentication: Blockchain-Driven Decentralized Authentication in Web 3.0. *J. Web Eng.* **2024**, *23*, 611–636. [CrossRef]
2. Li, X.H. Blockchain-based Cross-border E-business Payment Model. In Proceedings of the 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT), Hangzhou, China, 5–7 March 2021; pp. 67–73. [CrossRef]
3. Vizzarri, A.; Mazzenga, F. 6G Use Cases and Scenarios: A Comparison Analysis Between ITU and Other Initiatives. *Future Internet* **2024**, *16*, 404. [CrossRef]
4. Villarreal, E.R.D.; García-Alonso, J.; Moguel, E.; Alegría, J.A.H. Blockchain for Healthcare Management Systems: A Survey on Interoperability and Security. *IEEE Access* **2023**, *11*, 5629–5652. [CrossRef]

5. Haleem, A.; Javaid, M.; Singh, R.P.; Suman, R.; Rab, S. Blockchain technology applications in healthcare: An overview. *Int. J. Intell. Netw.* **2021**, *2*, 130–139. [CrossRef]
6. Abbas, Q.E.; Sung-Bong, J. A Survey of Blockchain and Its Applications. In Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Okinawa, Japan, 11–13 February 2019; pp. 1–3. [CrossRef]
7. Matera, F.; Settembre, M.; Vizzarri, A.; Mazzenga, F.; Llorca, J.; Tulino, A. Opportunities and Challenges for the Integration of AI in Network Evolution Toward 6G. In Proceedings of the 2024 AEIT International Annual Conference (AEIT), Trento, Italy, 25–27 September 2024; pp. 1–6. [CrossRef]
8. Bashir, I. *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained*, 4th ed.; Packt Publishing: Birmingham, UK, 2023.
9. Kim, A.; Essaid, M.; Ju, H. Survey on Blockchain P2P Network. In Proceedings of the 2023 24st Asia-Pacific Network Operations and Management Symposium (APNOMS), Sejong, Republic of Korea, 6–8 September 2023; pp. 413–416.
10. Saini, R.; Saini, N.; Aleem, A.; Singla, A. P2P Communication System Using Blockchain. In Proceedings of the 2025 3rd International Conference on Communication, Security, and Artificial Intelligence (ICCSAI), Greater Noida, India, 4–6 April 2025; pp. 1262–1266. [CrossRef]
11. Dayong, Z.; Wahab, N.H.A.; Kadir, K.A.; Aldhaqm, A.; Nasir, H.M.; Wong, K.Y. Research on Blockchain: Privacy Protection of Cryptography Blockchain-Based Applications. In Proceedings of the 2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Taiz, Yemen, 10–11 October 2023; pp. 1–6. [CrossRef]
12. Shwetha, N.; Sujatha, S.R. A Comprehensive Study on Blockchain-Enabled Cryptographic Frameworks for Electronic Health Records. In Proceedings of the 2026 5th International Conference on Computer Networks, Big Data and IoT (ICCBI), Dubai, United Arab Emirates, 18–20 June 2026; pp. 248–255. [CrossRef]
13. Bhalerao, G. Decentralized Cryptographic Governance for Cloud-Based Medical Record Infrastructure: A Distributed Ledger Framework for Cyber-Resilient Healthcare Data Sovereignty. In Proceedings of the 2026 2nd International Conference on Emerging and Intelligent Technologies and Systems (EITS), Valencia, Spain, 9–12 June 2026; pp. 136–143. [CrossRef]
14. De Almeida, S.C.V.C.; Rodrigues, C.K.D.S.; Giozza, W.F. Operational requirement ranking for consensus algorithm selection in Blockchain health management systems. In Proceedings of the 2023 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, Portugal, 20–23 June 2023; pp. 1–6. [CrossRef]
15. Bessani, A.N. Byzantine State Machine Replication in the Age of Blockchains: Fundamentals and Recent Results (Tutorial). In Proceedings of the 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks—Supplemental Volume (DSN-S), Porto, Portugal, 27–30 June 2023; p. 1. [CrossRef]
16. Jain, T.K. Protection of Clinical Records using Blockchain with Proof of Stake as Consensus Algorithm. In *Proceedings of the 7th International Conference on Information Management & Machine Intelligence (ICIMMI '25)*; Association for Computing Machinery: New York, NY, USA, 2026; Volume 57, pp. 1–13. [CrossRef]
17. Zhu, S.; Sun, C.; Zhang, H.; Wang, L. IIN-Health: A Dual-Chain Federated Learning Framework with Adaptive BFT Consensus for Auditable Medical Data Sharing. *Intell. Converg. Netw.* **2026**, *7*, 146–165. [CrossRef]
18. Fang, Y.; Zhou, Z.; Dai, S.; Yang, J.; Zhang, H.; Lu, Y. PaVM: A Parallel Virtual Machine for Smart Contract Execution and Validation. *IEEE Trans. Parallel Distrib. Syst.* **2024**, *35*, 186–202. [CrossRef]
19. Bărbută, D.-E.; Butincu, C.N.; Alexandrescu, A. SPARK-IT Green: A Decentralized Blockchain-Based Platform for Sustainable Innovation, Mentorship and DAO Governance. In Proceedings of the 2025 7th International Conference on Blockchain Computing and Applications (BCCA), Durbovnic, Croatia, 14–17 October 2025; pp. 451–458. [CrossRef]
20. Kasyapa, M.S.B.; Vanmathi, C. Blockchain integration in healthcare: A comprehensive investigation of use cases, performance issues, and mitigation strategies. *Front. Digit. Health* **2024**, *6*, 1359858. [CrossRef] [PubMed] [PubMed Central]
21. Stamatellis, C.; Papadopoulos, P.; Pitropakis, N.; Katsikas, S.; Buchanan, W.J. A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric. *Sensors* **2020**, *20*, 6587. [CrossRef]
22. Zaabar, B.; Cheikhrouhou, O.; Jamil, F.; Ammi, M.; Abid, M. HealthBlock: A secure blockchain-based healthcare data management system. *Comput. Netw.* **2021**, *200*, 108500. [CrossRef]
23. Deepa, K.; Parmar, A.S.; Goyal, H.S.; Mishra, K.; Panda, S. HealthRec-Chain: Patient-centric blockchain enabled IPFS for privacy preserving scalable health data. *Comput. Netw.* **2024**, *241*, 110223. [CrossRef]
24. Tran, T.D.; Duc, M.T.; Thuy, T.L.T.; Duy, P.T.; Cam, N.T.; Pham, V.H. Medichain: A Multi-chain Based System for Managing Prescription Medications Using Zero Knowledge Proofs. In *Proceedings of the International Conference on Intelligent Systems and Networks (ICISN 2024)*; Lecture Notes in Networks and Systems; Nguyen, T.D.L., Dawson, M., Ngoc, L.A., Lam, K.Y., Eds.; Springer: Singapore, 2024; Volume 1077. [CrossRef]
25. Liu, J.; Fan, Y.; Sun, R.; Liu, L.; Wu, C.; Mumtaz, S. Blockchain-Aided Privacy-Preserving Medical Data Sharing Scheme for E-Healthcare System. *IEEE Internet Things J.* **2023**, *10*, 21377–21388. [CrossRef]

26. Fernandes, A.; Rocha, V.; Conceição, A.F.D.; Horita, F. Scalable Architecture for sharing EHR using the Hyperledger Blockchain. In Proceedings of the 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), Salvador, Brazil, 16–20 March 2020; pp. 130–138. [CrossRef]
27. Nagori, M.; Patil, A.; Deshmukh, S.; Vaidya, G.; Rahangdale, M.; Kulkarni, C.; Kshirsagar, V. MultiChain Enabled EHR Management System and Predictive Analytics. In *Smart Trends in Computing and Communications*; Zhang, Y.-D., Mandal, J.K., So-In, C., Thakur, N.V., Eds.; Springer: Singapore, 2020; Volume 165, pp. 179–187.
28. Rai, B.K.; Fatima, S.; Satyarth, K. Patient-Centric MultiChain Healthcare Record. *Int. J. E-Health Med. Commun.* **2022**, *13*, 1–14. [CrossRef]
29. Aziz Torongo, A.; Toorani, M. Blockchain-based Decentralized Identity Management for Healthcare Systems. *arXiv* **2023**, arXiv:2307.16239.
30. Psarra, E.; Apostolou, D.; Verginadis, Y.; Patiniotakis, I.; Mentzas, G. Permissioned Blockchain Network for Proactive Access Control to Electronic Health Records. *BMC Med. Inform. Decis. Mak.* **2024**, *24*, 303. [CrossRef]
31. Nedaković, A.; Hasselgren, A.; Kralevska, K.; Gligoroski, D. Hyperledger fabric platform for healthcare trust relations—Proof-of-Concept. *Blockchain Res. Appl.* **2023**, *4*, 100156. [CrossRef]
32. Glicksberg, B.S.; Burns, S.; Currie, R.; Griffin, A.; Wang, Z.J.; Haussler, D.; Goldstein, T.; Collisson, E. Blockchain-Authenticated Sharing of Genomic and Clinical Outcomes Data of Patients with Cancer: A Prospective Cohort Study. *J. Med. Internet Res.* **2020**, *22*, e16810. [CrossRef]
33. Zhang, R.; Xue, R.; Liu, L. Security and Privacy for Healthcare Blockchains. *IEEE Trans. Ind. Inform.* **2021**, *15*, 3668–3686. [CrossRef]
34. Cheng, A.H.; Luo, J.W.; Chang, M.T. EHealth Integrity Model Based on Permissioned Blockchain. *Future Internet* **2019**, *11*, 76. [CrossRef]
35. Martin, G.C.; Lee, M.D.; Trujillo, T.L.; Gray, A.P. TrustHealth: Enhancing eHealth Security with Blockchain and Trusted Execution Environments. *Electronics* **2024**, *13*, 2425. [CrossRef]
36. Kuo, T.-T.; Ohno-Machado, L. ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks. *arXiv* **2018**, arXiv:1802.01746.
37. Passerat-Palmbach, J.; Farnan, T.; Miller, R.; Gross, M.S.; Flannery, H.L.; Gleim, B. A Blockchain-Orchestrated Federated Learning Architecture for Healthcare Consortia. *arXiv* **2019**, arXiv:1910.12603.
38. Zhang, D.; Wang, S.; Zhang, Y.; Zhang, Q.; Zhang, Y. A Secure and Privacy-Preserving Medical Data Sharing via Consortium Blockchain. *Secur. Commun. Netw.* **2022**, *2022*, 2759787. [CrossRef]
39. Roberts, A.C.; Smith, L.M.; Perez, R.A. Blockchain Technology in Healthcare: A Systematic Review. *Healthcare* **2019**, *7*, 56. [CrossRef]
40. Singh, A.B.; Kumar, J.P.; Sharma, N.M. Chapter 1: Blockchain Technology and Its Applications in Health Data Privacy: A Survey. In *Futuristic Trends in Block Chain Applications IIP Series*; IIP: Chikkamagaluru, India, 2023; Volume 3, ISSN 3070-7617 (Online), E-ISBN 978-93-6252-632-8. Available online: https://iipseries.org/assets/docupload/rsl2024131DF2E467CAA3B.pdf (accessed on 16 August 2026).
41. Xi, P.; Zhang, X.; Wang, L.; Liu, W.; Peng, S. A Review of Blockchain-Based Secure Sharing of Healthcare Data. *Appl. Sci.* **2022**, *12*, 7912. [CrossRef]
42. Singh, A.K.; Garg, A.; Nayyar, A. Blockchain for Security and Privacy in Healthcare Informatics. Available online: https://digital-library.theiet.org/doi/abs/10.1049/PBHE041E_ch5 (accessed on 16 August 2026).
43. Main, H.; Velhal, D.; Nair, A.; Hule, A.; Dalvi, H. Blockchain-Assisted Agentic AI Framework for Intelligent Digitization of Indian Medical Prescriptions. In Proceedings of the 2026 IEEE Global Symposium on Emerging and Communication Technologies (GSEACT), Hyderabad, India, 12–13 June 2026; pp. 1–6. [CrossRef]
44. Bhardwaj, C.; Kushwaha, V.K.; Gupta, S.; Sharan, B.; Singh, M.K. A Secure Blockchain–IPFS Framework for Decentralized Healthcare Image Storage with ECIES-Based Access Control. In Proceedings of the 2026 IEEE Global Symposium on Emerging and Communication Technologies (GSEACT), Hyderabad, India, 12–13 June 2026; pp. 1–6. [CrossRef]
45. Aparna, P.; Pramod, S.; Bhagyashree, K.J.; P, R.K. A Secure and Scalable Digital Healthcare System Based on Blockchain Telemedicine Architecture. In Proceedings of the 2026 IEEE Global Symposium on Emerging and Communication Technologies (GSEACT), Hyderabad, India, 12–13 June 2026; pp. 1–6. [CrossRef]
46. Rahman, A.; Walee, N.A.; Mahir, S.H.; Tashrif, M.T.A.; Rana, M.S. BIoMT-VFL: Ensuring Blockchain-based Security and Privacy in Internet of Medical Things through Vertical Federated Learning. In Proceedings of the 2026 7th International Conference on Artificial Intelligence, Robotics, and Control (AIRC), Savannah, GA, USA, 8–10 April 2026; pp. 428–433. [CrossRef]
47. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; Caro, A.D.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the EuroSys 2018 Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.

48. EEA: The Leading Nonprofit Driving Enterprise Ethereum Adoption. Available online: https://entethalliance.org/ (accessed on 16 August 2026).
49. Li, Y.; Li, D.; Yang, Q. A Blockchain-Based Semi-Centralized Business System for Multi-Enterprise. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023; pp. 8859–8864. [CrossRef]
50. Buterin, V. Ethereum White Paper. A Next-Generation Smart Contract and Decentralized Application Platform. 2014. Available online: https://ethereum.org/whitepaper (accessed on 16 August 2026).
51. Nakai, T.; Sakurai, A.; Hironaka, S.; Shudo, K. A Formulation of the Trilemma in Proof of Work Blockchain. *IEEE Access* **2024**, *12*, 80559–80578. [CrossRef]
52. Cavicchioli, M. The Merge and NFT: What will happen with Ethereum PoS. Cryptonomist Blog. 2022. Available online: https://en.cryptonomist.ch/2022/10/31/merge-and-nfts/ (accessed on 16 August 2026).
53. Wood, G. Ethereum: A Secure Decentralized Generalised Transaction Ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
54. Solidity. Solidity 0.8.19 Release Announcement. 22 February 2023. Available online: https://www.soliditylang.org/blog/2023/02/22/solidity-0.8.19-release-announcement/ (accessed on 16 August 2026).
55. Ethereum Foundation. ERC-20 Token Standard. Available online: https://ethereum.org/developers/docs/standards/tokens/erc-20 (accessed on 16 August 2026).
56. Ethereum Foundation. ERC-721 Non-Fungible Token Standard. Available online: https://ethereum.org/developers/docs/standards/tokens/erc-721/ (accessed on 16 August 2026).
57. Nchinda, N.; Cameron, A.; Retzepi, K.; Lippman, A. MedRec: A Network for Personal Information Distribution. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; pp. 637–641. [CrossRef]
58. Wu, Y.; Hu, Y.; Chen, M.; Yesha, Y.; Debbah, M. Blockchains for Internet of Things: Fundamentals, Applications, and Challenges. *IEEE Netw.* **2024**, *38*, 443–450. [CrossRef]
59. Rizzardi, A.; Sicari, S.; Jesus F. Cevallos, M.; Coen-Porisini, A. IoT-driven blockchain to manage the healthcare supply chain and protect medical records. *Future Gener. Comput. Syst.* **2024**, *161*, 415–431. [CrossRef]
60. Solve.care Ltd. *SC Whitepaper*; Solve.care Ltd.: Wollongong, Australia, 2024.
61. Han, Y.; Zhang, Y.; Vermund, S.H. Blockchain Technology for Electronic Health Records. *Int. J. Environ. Res. Public Health* **2022**, *19*, 15577. [CrossRef] [PubMed] [PubMed Central]
62. Semenzin, S.; Rozas, D.; Hassan, S. Blockchain-based application at a governmental level: Disruption or illusion? The case of Estonia. *Policy Soc.* **2022**, *41*, 386–401. [CrossRef]
63. Greenspan, G. MultiChain Private Blockchain—White Paper. MultiChain. 2015. Available online: https://www.multichain.com/download/MultiChain-White-Paper.pdf (accessed on 16 August 2026).
64. Polge, J.; Robert, J.; Le Traon, Y. Permissioned blockchain frameworks in the industry: A comparison. *ICT Express* **2021**, *7*, 229–233. [CrossRef]
65. Ahmed, I.; Turki, M.; Baklouti, M.; Dammak, B.; Alshahrani, A. Towards an Optimized Blockchain-Based Secure Medical Prescription-Management System. *Future Internet* **2024**, *16*, 243. [CrossRef]
66. Sumo, R.; Jong, S. Use of Blockchain Technology to Accelerate Digital Health Transformation Programs. *Blockchain Healthc. Today* **2025**, *8*. [CrossRef] [PubMed] [PubMed Central]
67. Coin Sciences Ltd. MultiChain Developer Documentation. Available online: https://www.multichain.com/developers (accessed on 16 August 2026).
68. Python Software Foundation. *Python 3.11.0 Release Notes*; Python Software Foundation: Beaverton, OR, USA, 2022.
69. Górski, T. AdapT: A reusable package for implementing smart contracts that process transactions of congruous types. *Softw. Impacts* **2024**, *21*, 100694. [CrossRef]
70. Górski, T. CongruenT: A package for smart contract verification functions that ensures the reuse of condition objects and classes. *SoftwareX* **2026**, *35*, 102796. [CrossRef]
71. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access* **2019**, *7*, 22328–22370. [CrossRef]
72. Mainland, G.; Morrisett, G.; Welsh, M. Flask: Staged Functional Programming for Sensor Networks. In Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming (ICFP '08), Victoria, BC, Canada, 22–24 September 2008; pp. 335–346.
73. Ziyodov, A. Python Web Applications and WSGI. *Int. Bull. Appl. Sci. Technol.* **2023**, *3*, 402–405. [CrossRef]
74. Mhatre, S.M.; Mishra, B.K.; Raja, I.K.; Goswami, B.K.; Agrawal, S.; Upadhyay, P. Cryptosight: Decrypting Future Trends with LSTM, PyCaret and Jinja 2.0. In Proceedings of the 2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE), Rourkela, India, 19–21 July 2024; pp. 1–4.
75. Westerveld, D. *API Testing and Development with Postman: A Practical Guide to Creating, Testing, and Managing APIs for Automated Software Testing*; Packt Publishing: Birmingham, UK, 2021.

76. Thooriqoh, H.A.; Mulyo, B.M.; Rakhmadi, A. Advanced RESTful API Testing: Leveraging Newman's Command-Line Capabilities with Postman Collections. In Proceedings of the 2024 IEEE 10th Information Technology International Seminar (ITIS), Surabaya, Indonesia, 6–8 November 2024; pp. 188–193. [CrossRef]
77. Bez, M.; Fornari, G.; Vardanega, T. The Scalability Challenge of Ethereum: An Initial Quantitative Analysis. In Proceedings of the IEEE International Conference on Service-Oriented System Engineering (SOSE), San Francisco, CA, USA, 4–9 April 2019; pp. 167–176.. [CrossRef]
78. Ethereum Foundation. Single Slot Finality. Ethereum.org. 2024. Available online: https://ethereum.org/roadmap/single-slot-finality/ (accessed on 16 August 2026).
79. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018; pp. 264–276. [CrossRef]
80. Baliga, A.; Subhod, I.; Kamat, P.; Chatterjee, S. Performance Evaluation of the Quorum Blockchain Platform. *arXiv* **2018**, arXiv:1809.03421. [CrossRef]
81. Minango, J.; Carvajal Mora, H.; Zambrano, M.; Orozco Garzón, N.; Pérez, F. Distributed Ledger Technology in Healthcare: Enhancing Governance and Performance in a Decentralized Ecosystem. *Technologies* **2025**, *13*, 58. [CrossRef]
82. R3. Performance Benchmarking Results. R3 Technical Documentation. Available online: https://docs.r3.com/en/platform/corda/4.8/enterprise/performance-testing/performance-results.html (accessed on 16 August 2026).
83. Liu, J. Key Technologies of a Service Platform Based on Blockchain Technology. In Proceedings of the 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 29–30 April 2023; pp. 1–5. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.