

Blockchain as a Service for Electronic Health Records

Suraj Gupta^{†*}, Frédéric Lang^{*}, Umar Ozeer[†], and Gwen Salaün^{*}

^{*}Université Grenoble Alpes, CNRS, Grenoble INP, Inria, LIG, 38000 Grenoble France

[†]EURIS Cloud Santé, Boulogne-Billancourt, France

Email: *{frederic.lang, gwen.salaun}@inria.fr, †{suraj.gupta, umar.ozeer}@euris.com

Abstract—EHRs or Electronic Health Records are digital medical records regarding the health of patients. They are sensitive in nature and their storage and management needs to adhere to strict guidelines. Multiple applications depend on this data for efficiently managing their healthcare requirements. This work proposes FaSTr, a solution based on blockchain that enables Fast, Secure and Transparent communication and collaboration between applications, by providing services. These services are accessible via defined protocols, such as APIs. FaSTr exposes secure API endpoints to provide functionality to manage and share EHRs. This is achieved with the help of several novel ideas: A compatibility layer that improves interoperability, an API server that exposes important endpoints for services and a private network of nodes for each service resulting in high availability and robustness. Finally, performance evaluations demonstrate the real-world usability and scalability of FaSTr.

Index Terms—EHR, Blockchain, Attribute-Based Encryption, Services

I. INTRODUCTION

Electronic Health Records (EHRs) serve as the foundation for modern healthcare technology. Clinical Decision Support Systems (CDSS) provide real-time insights for enhanced diagnostics, while telehealth platforms enable comprehensive remote consultations. Population health management tools identify trends for preventive care, and patient portals empower individuals to access their medical records directly. Research platforms utilize de-identified datasets to drive medical innovation, from drug development to personalized medicine. These diverse use cases demonstrate the critical role of EHRs in streamlining healthcare processes and supporting data-driven care.

Successfully leveraging this ecosystem demands robust interoperability among EHR providers. Healthcare organizations face significant challenges with inconsistent data formats, varying access protocols, and incompatible authentication mechanisms across different systems [1]. Without unified standards, delivering comprehensive healthcare solutions becomes increasingly complex. A system promoting interoperability must establish common protocols for data exchange while maintaining flexibility for integration across diverse storage infrastructures, rather than creating yet another isolated storage solution.

Security is another major concern in handling EHRs. EHR management must comply with stringent regulations including HIPAA and GDPR [2]. These frameworks mandate comprehensive protection measures: robust authentication protocols,

granular access controls, and immutable audit logs. In the European Union, National Data Protection Authorities conduct GDPR audits and ensure compliance. Maintaining a tamper-proof record of all access permissions and interactions is critical for assisting regulatory compliance, by having complete trust on the audit logs, as they are immutable in nature. Finally, a centralized EHR system may offer standardized protocols but creates vulnerability through single points of failure. Conversely, decentralized approaches mitigate this risk while maintaining enforceable access policies, though they introduce integration complexities.

Efficiency and real-time data access are vital for healthcare applications. Complex multi-layered authentication protocols can introduce significant delays in EHR retrieval [3]. For CDSS that need information in near real-time, this can be detrimental. However, quick retrieval must not compromise security. A well-designed system must ensure high availability and responsiveness, handling large volumes of EHR requests while maintaining strict access control policies.

The need for immutability immediately suggests a solution based on blockchain, which is a decentralized, distributed ledger technology that securely records information in a way that prevents tampering or alteration. However, the use of blockchain for EHR management presents several challenges. First, a blockchain network should never store any sensitive data directly, as we do not aim to provide a HIPAA/GDPR-compliant storage solution. Instead, to facilitate greater collaboration, we require a provider-agnostic system that integrates with multiple EHR providers while ensuring data security and regulatory compliance. Second, data inside the blockchain is visible to all members of the blockchain network, which poses risks to privacy and access control. Hence, mechanisms for effective segregation of data are necessary to maintain compliance and security. For example, it would not make sense to give an auditor access to data used for access management.

In this paper, we propose FaSTr, a blockchain-based solution that aims to make the process of interacting with EHRs fast, secure, and transparent by providing services. As shown in Figure 1, FaSTr integrates with an EHR storage system and provides services using secure API endpoints that authorized applications can interact with to access the EHR data they need. The salient features of FaSTr are as follows:

- Handle multiple requests from applications for EHRs in

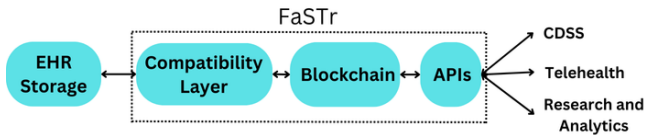


Fig. 1. System architecture demonstrating the integration of FaSTR with existing EHR storage systems. The compatibility layer bridges traditional EHR storage with the blockchain network, while applications (CDSS, Telehealth, Research) interact through standardized API endpoints.

near-real time while maintaining high availability and robustness.

- Enable strict access control policies for accessing EHRs.
- Integrate with EHR storage systems and standardize the way applications make their requests.
- Provide immutable logging of changes in access policies and all interactions with EHRs.

To maintain high availability and robustness, FaSTR employs a private network of compute units called nodes. These nodes handle the requests made by applications. Strict access control policies are enforced using Attribute-Based Encryption (ABE), which also prevents the direct storage of sensitive information on-chain. ABE was first introduced by Sahai and Waters as a cryptographic technique enabling fine-grained access control over encrypted data [4]. A compatibility layer facilitates interoperability by enabling integration with any EHR storage solution. An API server provides standardized interaction methods, invoking smart contracts on the blockchain for authorization. Certificate Authorities manage authentication for the blockchain network. Finally, the immutability aspect of the blockchain technology ensures a trustworthy log of all requests, enhancing auditability and security.

This paper is divided into the following sections. Section II introduces the basics of a blockchain network and ABE. Thereafter, Sections III & IV introduce FaSTR, its working and discuss a threat analysis. Section V discusses the experiments and deployment considerations for FaSTR. Finally, Sections VI & VII talk about the related works in the field and conclude this paper while discussing the next steps for this work.

II. PREREQUISITES

This section discusses the technologies used and their basic concepts. The following topics will be discussed: components of a blockchain network and an introduction to ABE.

A. Blockchains

Blockchains were originally conceptualised in 2008 by an anonymous entity known as Satoshi Nakamoto [5], and it gained prominence as the underlying technology for Bitcoin, the first cryptocurrency. Blockchain technology is a decentralized and distributed digital ledger system that allows for secure, transparent, and tamper-resistant recording of transactions across multiple computers. Any action that is processed or recorded by the blockchain is called a transaction. This could be the creation of new assets or the change in ownership of an asset. An asset is a piece of digital information, that

is stored on a blockchain network. In our context, an asset could mean an Electronic Health Records or a batch of records. Transactions make up the contents of a block, and the hash of this block is stored in the next block, hence, resulting in an immutable chain of blocks. This immutable record of events makes this technology an extremely good candidate for a number of industries, such as finance and healthcare.

Blockchain technology can be divided in two broad types, permissioned and permissionless. As their names suggest, a permissionless network can be accessed by anyone, but a permissioned network would require explicit authentication. Following are the core concepts of a blockchain network:

1) *General flow*: A user sends a transaction to the network. This request is considered by the cluster of consensus nodes which are executing a consensus algorithm. If the consensus algorithm gives a positive result, a new block containing the user-sent transaction is added to each ledger of each peer, otherwise the block is rejected.

2) *Blocks*: A block in a network comprises of a list of transactions and the hash of the previous block, forming an immutable chain of blocks.

3) *Consensus Algorithm*: Each block that is added to the network must be governed by some logic/rules. This set of rules governing the addition of blocks on the network is called the consensus algorithm. Various consensus algorithms exist in distributed systems. Some, like Proof of Work (PoW), primarily focus on rewarding consensus nodes for their participation [6]. Others, such as Practical Byzantine Fault Tolerance (PBFT), concentrate mainly on error resistance [7].

4) *Smart Contracts*: In the context of blockchain networks, smart contracts are self-executing programs that automatically enforce the terms of an agreement without requiring intermediaries. These contracts are deployed on decentralized blockchains and operate based on predefined rules written in code. When predetermined conditions are met, the smart contract executes the corresponding actions, such as transferring digital assets, verifying identities, or triggering external processes. In FaSTR, we deploy smart contracts to automate various on-chain tasks, such as authorization and updating access policies for Electronic Health Records. By leveraging cryptographic security and decentralized consensus mechanisms, smart contracts ensure transparency, immutability, and trustworthiness in digital transactions.

B. Attribute-Based Encryption

ABE is a public-key encryption technique, where a user's secret key and the ciphertext depend on attributes (for example occupation or floor number). Access to data is granted if user attributes match the access policy of the encrypted data, offering flexible access control. ABE has two types: Key Policy ABE (KP-ABE) and Ciphertext Policy ABE (CP-ABE). In KP-ABE, the access policy is linked to the user's secret key, while attributes describe the data. In CP-ABE, the access policy is in the ciphertext, and attributes are tied to the user's secret key. We use CP-ABE as it allows us to define a role-based authorization system, where roles can be defined by the

attributes of a user. For example, we can define an access policy stating that an EHR can only be accessed by the role 'Doctor'. Only the secret keys that have the role 'Doctor' embedded in them will be able to access that EHR.

III. SYSTEM DESIGN AND ARCHITECTURE

This section discusses the system design, its components, and our approach to addressing the requirements and challenges mentioned in the introduction.

FaSTr prioritizes high availability through networks of nodes within the blockchain that eliminate single points of failure. The system continues to function as long as the majority of nodes remain operational. Unlike resource-intensive consensus mechanisms used in traditional blockchains like Bitcoin and Ethereum, FaSTr employs a fault-tolerant RAFT algorithm which prioritizes performance and scalability over token-based incentives.

In order to maximise compatibility with multiple EHR storage solutions, a compatibility layer adapts to various systems by encrypting important meta-data with ABE. Healthcare applications require distinct access patterns: telehealth platforms need read capabilities, auditors examine logs, and research platforms require both read/write permissions. FaSTr addresses this by storing data on service-specific private ledgers across dedicated node networks.

Blockchain's immutable transaction history enables tamper-proof audit logs of all access events and permission changes. Standardized APIs invoke smart contracts upon authentication, eliminating integration complexity while maintaining security protocols. Figure 2 gives an overview of the system.

A. Compatibility Layer

This subsection will discuss how the compatibility layer interacts with an EHR database. The compatibility layer is responsible for adapting to the EHR storage and providing the blockchain with necessary metadata.

As illustrated by Figure 4, it first sends a request to the EHR storage to connect. Once accepted, metadata necessary for managing and accessing the sensitive EHRs is requested. The compatibility layer is also responsible for encrypting all the data before storing it on the blockchain. This encryption of metadata helps us achieve two goals. First, we avoid storing any information directly from the EHR storage. Second, we can use this encrypted metadata for authorization.

For each asset to be managed, its unique identifier in the EHR storage, its hash and its access policy are requested. An access policy defines the permissions required to access that asset. The compatibility layer encrypts the hash of the asset with ABE. The attributes required to decrypt this information are determined by the access policy.

Finally, this component uploads the encrypted data, the hash of the asset and the access policy to the blockchain, using the same unique identifier. This way, the data in the EHR storage and its metadata in the blockchain have the same identifier.

While the compatibility layer promotes interoperability, it differs significantly from standards specifically designed to

promote interoperability, such as HL7's Fast Healthcare Interoperability Resources (FHIR) [8]. While FHIR standardizes data structures and RESTful interfaces for direct system communication, FaSTr's compatibility layer serves as an integration and security layer that bridges EHR storage systems with blockchain infrastructure. Unlike FHIR's focus on data exchange formats, FaSTr abstracts underlying storage while adding immutable access control through ABE encryption and blockchain-based audit logging. FaSTr can work alongside FHIR-compliant systems by treating FHIR endpoints as EHR storage solutions, thereby adding blockchain security and transparency to existing implementations. This approach provides advantages FHIR alone cannot offer—tamper-proof audit trails, granular attribute-based access control, and decentralized authorization—while maintaining flexibility to adapt to any EHR storage format, whether FHIR-compliant or proprietary legacy systems.

B. Requesting ABE Keys

All keys are generated with the help of a key generation center (KGC), which uses a master secret key and a set of parameters to generate the required keys with the embedded attributes. Figure 3 gives an overview of the process for requesting ABE keys:

- 1) The application requests for signing up for the EHR storage. In this sign-up request, they define their necessities and the data they would need.
- 2) The EHR storage understands this request and defines a certain set of attributes, also called credentials (L) for the application that best fits their needs.
- 3) The application can take their credentials (L) and request for their key from the KGC.
- 4) The KGC uses the credentials (L) and the master secret key to generate the required secret key (SK_L) and return it to the application.
- 5) The application obtains their secret key (SK_L) and can use it to obtain the data they need.

C. Access Management

This subsection discusses data access management. To standardise the way applications communicate with FaSTr, we design an API server (as shown in Figure 2). This server is responsible for hosting all the APIs that can be used by the applications to interact with EHRs. Requests made with these APIs will only be considered if they are accompanied with the right cryptographic material. On successful authentication, these APIs can invoke smart contracts on the blockchain network that can perform authorization and give access to the sensitive EHRs.

Any application that wishes to interact with the EHRs must first have itself registered with the blockchain network. Registering with the blockchain network allows us to use a certificate authority to manage certificates for authentication with the blockchain. A set of private keys with embedded user attributes is also generated as discussed in the previous subsection.

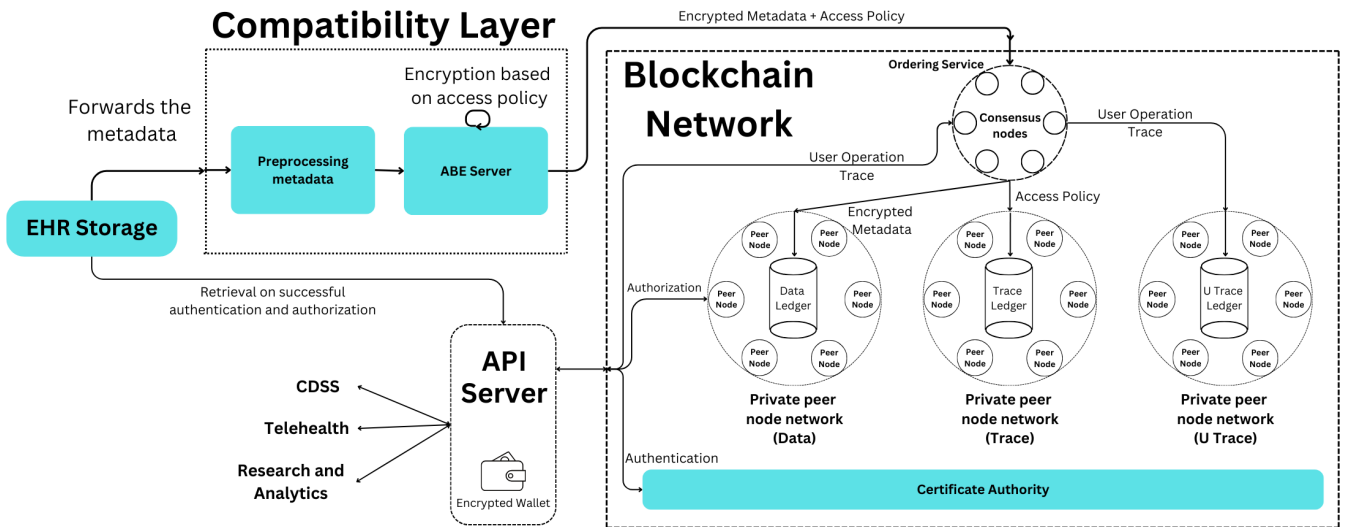


Fig. 2. Detailed FaSTr implementation architecture showing the interaction between core components.

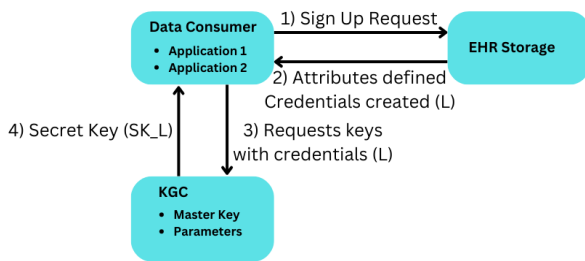


Fig. 3. Step-by-step process for attribute-based encryption key generation.

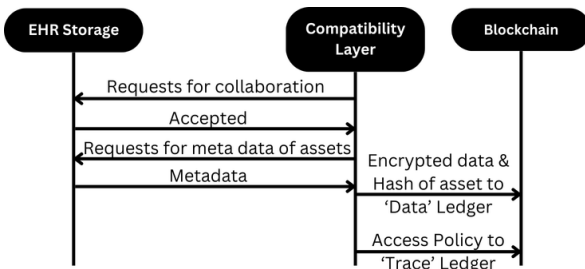


Fig. 4. Workflow diagram illustrating the compatibility layer's role in bridging EHR storage systems with the blockchain.

Now, this application can use the exposed APIs in the API server to make a request to read some EHRs. This request will first be analysed to see if it has proper cryptographic material to make requests to the blockchain network. On successful authentication by a smart contract, the application can make a request for an asset using its identifier.

The blockchain will then present encrypted data corresponding to the identifier of the asset. The application must successfully decrypt this data to have access to the requested asset. This decryption can be done with the secret key obtained during the generation of the ABE keys. If the attributes of the secret

key satisfy the access policy of the encrypted data, it will be successfully decrypted. For example, if the access policy was "TeleHealth or Research" and the secret key has the attribute "TeleHealth", it will be able to decrypt successfully. If the decrypted result matches the hash of the asset stored on the blockchain, authorization will be granted to access that asset. The API server will request the EHR storage for the asset using its identifier. Finally, the resulting EHR will be transferred to the application.

To enhance ease of use, an encrypted wallet is established at the API server. This wallet can securely store the private keys for ABE. These keys can be used on successful authentication with the blockchain. For example, an application might store their private key on this encrypted wallet. Next time the application authenticates itself on the blockchain to request for EHRs, its private key will automatically be used from the wallet for authorization.

If the access policy of any asset needs to be updated, the EHR storage must only send the new metadata for that asset to the compatibility layer. The compatibility layer will notice that some metadata for an existing identifier has arrived. It will encrypt the hash of the data according to the new access policy and issue an update request to the blockchain. The blockchain will thereafter update the access policy and the encrypted data for the asset pointed by the identifier.

Access revocation is achieved with the help of the Certificate Authority. We used a CA to manage the access of applications to the blockchain network. Access can be revoked and terminated from the blockchain by invalidating the certificates applications use to access the blockchain network.

D. Service Management

The following services are provided:

- Maintaining access control for EHRs
- Immutable logging of change in access permissions

- Immutable logging of all requests for EHRs

These services provide the means to have granular access control and transparent access to sensitive data for virtually any EHR storage system.

Providing such services would require a convenient way to manage data on the blockchain. However, a classic blockchain network has only one shared ledger that holds all the data. This introduces a problem where segregation of data becomes a challenge. For example, it would be impractical to store the metadata that helps in permission management in the same ledger as the one used to maintain immutable event logs. Moreover, there will be a huge volume of requests made by multiple different applications.

Since FaSTr aims to handle all these requests and provide services, it needs to be highly available. A single point of failure must be avoided at all costs. We overcome these problems with the help of a private network of nodes for each of the 3 services provided, as illustrated in Figure 2.

These private networks use their own private ledgers and do not remain public for all members of the blockchain. They remain visible only to the members of that private network. This allows us to solve both challenges. Firstly, the private network greatly increases availability of the system as more nodes handle requests made by applications. Moreover, a single point of failure is avoided. Secondly, we can give selective access to the blockchain network and efficiently segregate data stored on the blockchain network, based on the service it provides. For example, an auditor can be given access to the ledger holding immutable logs to conduct an audit, but they will not be allowed to access the ledger used for requesting EHRs.

The private ledger 'Data' keeps all the metadata required for authorization, 'Trace' maintains an immutable record of change in access policies and 'UserTrace' maintains an immutable record of all events when data was accessed from the EHR storage.

E. Interaction Flow

Figure 5 shows the interaction between an application requesting for EHRs and the blockchain network. In this scenario, we assume that the application has been registered and it has the necessary keys and certificates to access the blockchain network. The application uses an API to make a request for EHRs. This request must be accompanied by the right cryptographic material. Once the authenticity of the request is confirmed, the blockchain network sends some encrypted data associated to the requested EHR data. At this point, the application can use an encrypted wallet on the API server or their own personal wallet to decrypt the data using their private key. This private key is allotted when attributes for an application are defined. The decrypted result is sent to the blockchain. If it matches with the same hash value stored in the blockchain, then access is granted. The API server requests for the EHRs from the storage and provides the results to the application. A log of this entire interaction is generated and stored in an immutable manner in the blockchain.

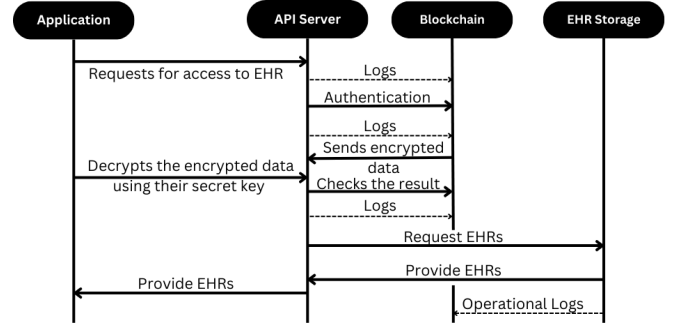


Fig. 5. Complete interaction flow when applications request EHR access through FaSTr.

IV. SECURITY ANALYSIS AND THREAT MODEL

FaSTr employs a structured threat modelling approach to systematically identify and address security vulnerabilities across the system. This comprehensive analysis reveals three critical trust boundaries that require protection: the interface between external applications and FaSTr infrastructure (API server boundary), the connection between FaSTr and existing EHR systems (EHR Storage boundary), and the integration point between FaSTr and its compatibility layer. Within these boundaries, several protected assets require specific security considerations based on their characteristics. Patient health records demand the highest level of confidentiality, access control policies require unwavering integrity, and audit logs necessitate both availability and integrity to fulfil their security functions.

The system faces several potential attack vectors that must be addressed through appropriate mitigation strategies. Adversaries may attempt Man-in-the-Middle (MITM) attacks by intercepting communications between applications and the FaSTr API server, potentially capturing sensitive authentication credentials or manipulating data in transit. To counter this threat, FaSTr implements TLS 1.3 [9] for all API communications. Additionally, mutual TLS authentication requires both client and server to authenticate using X.509 certificates [10], establishing a secure communication channel.

Access privilege escalation presents another significant risk, where malicious actors might attempt to gain unauthorized access to EHRs by manipulating ABE attributes or exploiting smart contract vulnerabilities. FaSTr addresses this through a principle of least privilege approach, providing users only the minimum necessary attributes. A hierarchical Role-Based Access Control system with explicit grant and revoke operations further provides granular permission control while limiting potential attack surfaces.

Cryptographic attacks constitute the third threat category, where adversaries might attempt to compromise the ABE encryption scheme or access private keys. FaSTr's defense includes Waters' CP-ABE implementation [11] with 256-bit security level for robust protection. The system stores private keys in tamper-resistant Hardware Security Modules and enforces automatic key rotation every 90 days, limiting

vulnerability windows.

Through this comprehensive security model, FaSTr provides a robust defence approach that protects EHR data across all system boundaries while maintaining the performance characteristics necessary for healthcare environments.

V. EXPERIMENTATION

This section discusses the practical implementation of the PoC (proof of concept), the tools and the assumptions used to test how this system performs. These tests were carried out to measure the impact on latency (in seconds) and throughput (in transactions per second (TPS)) with respect to an increasing ledger size. We benchmark these metrics as we want to find out how severely the throughput and latency are affected as the blockchain network grows.

A. Proof of Concept

The PoC is constructed by implementing all the components of Figure 2. The database of EHRs is implemented using PostgreSQL. The compatibility layer is a component that processes and encrypts data. For this purpose, it is developed using Python as it excels at handling data. The ABE functionality is set up using OpenABE. The Hyperledger Fabric framework is used to establish a permissioned blockchain network, which is discussed in the following section. A Node.js web application framework called Express, is used to develop the API server. The APIs are developed with the help of the Fabric Gateway. Finally, the testing environment is a virtual machine running on 4 cores of an i5 intel vPRO processor with 4GB of RAM.

B. Hyperledger Fabric

Hyperledger Fabric is an open-source blockchain framework developed by the Linux Foundation as part of the Hyperledger project. It is designed to be a modular and versatile platform for developing enterprise-grade blockchain applications. This framework is used in multiple production deployments in the supply chain, logistics and healthcare industries. A few examples include IBM Food Trust and MediLedger. The blockchain network in the PoC is developed using this framework. The components of Hyperledger Fabric used to build the PoC are as follows:

1) *Fabric CA*: Due to the permissioned nature of Fabric, a Certificate Authority (CA) is set up with Fabric CA for the access management of the network.

2) *Channels*: Fabric provides for private communication between the participating nodes of the blockchain network with the help of channels. A separate private ledger is maintained for each channel, and only the nodes that are a part of that channel are permitted to access that private ledger.

3) *Fabric Gateway*: The Fabric Gateway is a core component of a Hyperledger Fabric blockchain network, and coordinates the actions required to submit transactions and query ledger state on behalf of applications. By using the Gateway, applications only need to connect to a single endpoint in a Hyperledger Fabric blockchain network.

4) *Ordering Service*: Hyperledger Fabric features a node called an orderer that executes the consensus algorithm, which along with other orderer nodes forms an ordering service.

C. Hyperledger Caliper

Hyperledger Caliper is a blockchain performance benchmark framework and is a part of the Hyperledger foundation. It allows us to execute predetermined workloads to determine the performance of the network. The workloads used in these tests were taken directly from the PoC. For example, for the workload of the encrypted text, hash data of assets from the PostgreSQL database (serving as the EHR storage) was taken and encrypted with ABE. Moreover, for the workload of logs being stored, PostgreSQL Audit Extension (pgaudit) is used.

D. OpenABE

The OpenABE library is used to set up ABE. It is built on top of the abstract Zeuro Math library which supplies all of the elliptic-curve operations [12]. It is instantiated with the state-of-the-art Barreto-Naehrig (BN) curves with the embedding degree $k = 12$ (or commonly referred to as BN-254). This particular asymmetric curve is known to yield a very efficient pairing implementation and a security level equivalent to AES-128. Moreover, we benchmarked the efficiency of this library. We took 100 attributes and embedded them in a secret key. 100 attributes would result in $100!$ ($9.332622e+157$) number of possible combinations of attributes for access policies. This is enough for any use case and can provide for extremely granular access control. For our tests, we consider the worst case scenario and generate a key with all 100 attributes embedded. Then we perform encryption and decryption operations with this secret key. We perform these operations 10 times and obtain the following results as the average of the 10 iterations :

- Average time to generate key: 47 ms
- Average time for encryption: 202 ms
- Average time for decryption: 369 ms

These results show that we can have AES-128 level of security with near real time encryption and decryption, with a massive set of attributes.

E. Testing Parameters

The goal of our testing is to find out how the network performs in terms of latency and throughput, as the size of the blockchain network increases. Latency and throughput are important metrics as they express how responsive the network is and give an insight on the real-world usability of our proposition.

We use Hyperledger Caliper to send requests to the blockchain network and observe how these requests are handled. To determine maximum throughput, we employ rate controllers that let us control the request send rate. Starting with very low rates, we increase the rate if the network successfully processes requests without creating a backlog, allowing us to identify the maximum number of requests the network can handle.

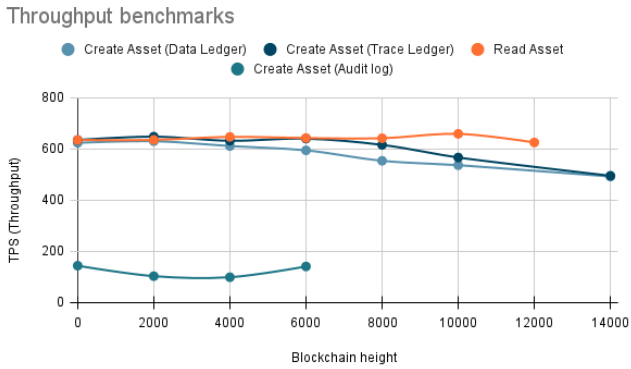


Fig. 6. Throughput benchmarking of various operations vs the blockchain height

The results in subsequent subsections illustrate throughput or latency versus blockchain height (network size). Blockchain height means the size of the blockchain. Throughput values represent the maximum number of transactions the network can process per second at specific blockchain heights, demonstrating optimal network performance as it grows. All network components are deployed on the same local network for controlled testing.

We test the three private networks (data, trace, and user trace ledgers) based on read/write operations performed by the compatibility layer and applications. Additionally, we evaluate ordering service scalability and performance impact as we increase the number of orderer nodes.

F. Creating an Asset

The compatibility layer adds blocks to the blockchain network containing transactions for metadata addition to the data ledger and permission tracking to the trace ledger. Application interaction traces are stored in the user trace ledger (tested separately due to different data volumes).

Figure 6 presents test results showing maximum possible throughput for data and trace ledgers at varying heights. Both networks demonstrate similar performance, with marginal differences explained by data volume: the data ledger stores 256-bit hashes plus encrypted information, while the trace ledger contains only access policies. Consequently, the trace ledger performs slightly better due to its less voluminous data blocks. Figure 7 illustrates latency evolution, averaging results from both networks for overall representation. The blockchain network maintains latency below 0.05 seconds until approximately 7000 blocks, then increases slightly above 0.05 seconds through 14000 blocks. When managing 100 assets per block, this translates to 1.4 million records at 14000 blocks.

The results demonstrate minimal performance degradation: throughput decline of only 100 TPS with average latency under 0.01 seconds while managing 1.4 million records. This shows that performance remains stable as the blockchain network scales.

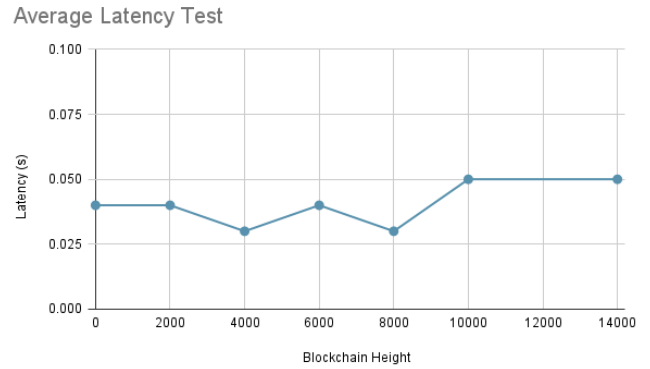


Fig. 7. Latency benchmarking of creating an asset on-chain vs the blockchain height

G. Reading an Asset

Applications interacting with FaSTr frequently request data reads from the data ledger, which stores encrypted data requiring decryption to verify access permissions. We focus testing on this ledger since it receives the majority of read requests. Figure 6 shows throughput plotted against blockchain height during read request execution. Remarkably, throughput remains practically unchanged as blockchain height increases. This stability occurs because read operations do not invoke consensus algorithms or require ordering service participation. The blockchain can maintain consistent performance regardless of ledger size.

Average latency remains at 0.01 seconds across all blockchain heights. Combined with 0.35 seconds average decryption time (from Section IV.D), the total time between requesting and obtaining an EHR record stays under one second. This confirms that applications can expect near real-time responses even as the system scales.

H. User Trace Metrics

The user trace ledger contains comprehensive logs including all application requests, network handling details, and complete records of operations performed on anonymized EHRs during research activities. This extensive logging ensures regulatory compliance and prevents malpractice, but results in extremely voluminous data storage.

Figure 6 demonstrates throughput versus blockchain height for log data addition. Throughput is significantly lower compared to other networks due to the large volume of log data. Assuming each asset includes 12 hours of system operation logs, 4000 assets would store approximately 5.4 years of complete system logs.

Despite lower throughput, the graph shows minimal decline as ledger height increases. The separate private network architecture ensures this ledger's lower performance does not impact efficiency of application-facing networks. The data ledger maintains high responsiveness for application requests while comprehensive audit logs are maintained independently.

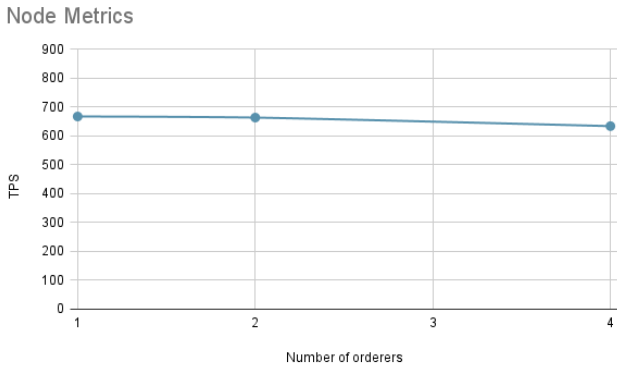


Fig. 8. Throughput benchmarking of the number of orderer nodes in the blockchain network

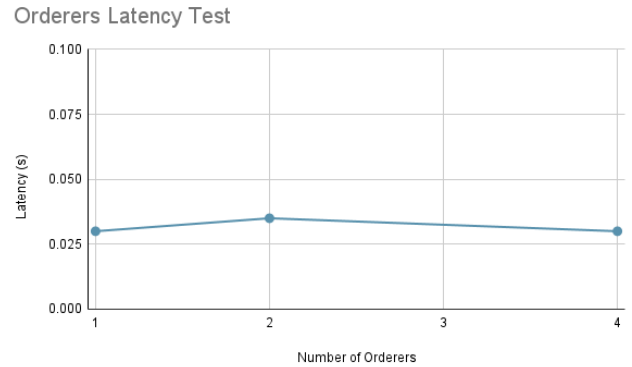


Fig. 9. Latency benchmarking of the number of orderer nodes in the blockchain network

I. Ordering Service Nodes

The ordering service validates blockchain transactions through consensus algorithms. We implement the RAFT consensus algorithm for its focus on fault tolerance in permissioned environments. While PBFT offers stronger Byzantine fault tolerance, FaSTr operates in trusted institutional contexts where such protection is less critical. Moreover, PBFT requires fixed node participation and cannot dynamically accommodate node joins and exits. RAFT’s Joint Consensus mechanism enables seamless addition and removal of nodes from the network, thereby enhancing system robustness by allowing the consensus network to adapt to changing conditions and recover from node failures without disrupting overall operation.

For scalability, we conduct tests to evaluate performance impact as we scale from 1 to 4 orderer nodes. Figures 8 and 9 present results for simple write requests on an empty blockchain, providing clear insights into ordering service influence on network performance.

Results show only marginal changes in both throughput and latency as orderer nodes increase. The ordering service can accommodate additional nodes for improved robustness without sacrificing network performance, enabling system resilience through redundancy.

J. Deployment Considerations in Distributed Networks

Implementing FaSTr across multiple distributed networks presents unique challenges beyond controlled laboratory environments. Real-world deployment scenarios must address several critical factors to ensure system reliability and regulatory compliance.

Network latency and geographic distribution significantly impact blockchain performance in healthcare environments. Network throughput depends heavily on the latency and bandwidth between distributed nodes, with consensus algorithms being particularly sensitive to delays [13]. This consideration is critical for real-time applications such as CDSS, where timely access to patient information directly impacts care outcome. To mitigate these challenges, FaSTr can benefit from regional node clustering strategies and dedicated network

infrastructure between participating institutions to maintain the performance necessary for clinical workflows.

Scalability presents another challenge as healthcare networks grow in scope and complexity. Our testing demonstrates that the blockchain network avoids significant latency issues as network size expands, but considerations must still be made for data volume management and cross-institutional query performance. Healthcare systems generate substantial data that require efficient storage strategies to maintain responsiveness. Complex queries spanning multiple institutions may encounter latency challenges that can be addressed through sharding strategies that partition data across institutional boundaries and caching mechanisms for frequently accessed cross-institutional information.

Regulatory compliance across jurisdictions presents perhaps the most complex challenge. Healthcare regulations vary significantly across regions, with HIPAA requirements in the United States differing substantially from GDPR provisions in Europe [14]. This creates challenges related to cross-border data transfer and potential conflicts between regulatory authorities. FaSTr can address these through jurisdiction-specific smart contracts that enforce localized regulations, configurable privacy settings based on regional requirements, and separate ledgers for different regulatory zones while still maintaining interoperability advantages.

VI. RELATED WORK

The storage and management of sensitive data have always posed challenges. Blockchain networks provide a secure, immutable, and robust way to address these challenges. The following works discuss how blockchain technology has been leveraged to enhance the storage, management, and sharing of EHRs.

A. Patient-Centric Access to EHRs

Electronic health records originate from and belong to patients, but they often face challenges in accessing them when needed. ActionEHR [15] proposes a solution based on a permissioned blockchain network that allows patients to

TABLE I
TABLE OF RELATED RESEARCH WORK

Paper	Technology used	Consensus	Access Control		Features				
			Authentification	Authorization	Permissions Tracking	Log management	Access revocation	Compatibility Layer	Scalability
ActionEHR [15]	HLF	PBFT	FabricCA	ChainCode		•	•		
MedRec [16]	Ethereum	PoW	PKC	Smart Contract	•			•	
FHIRChain [17]	Ethereum	PoW	PKC	Smart Contract	•		•		
Secure Cloud-Based EHR System Using Attribute-Based Cryptosystem and Blockchain [18]	HLF	–	ABE	ChainCode			•		
Blockchain-Based Approach for e-Health Data Access Management with Privacy Protection [19]	HLF	50% approval	ABE	ABE			•		
Ancile [20]	Ethereum	PoV	PKC	Smart Contract	•		•		
FaSTr	HLF	RAFT	FabricCA	ChainCode & ABE	•	•	•	•	•

manage their records across multiple hospitals. This facilitates timely and secure data sharing between healthcare providers while giving patients absolute control over EHR permissions. ActionEHR uses Hyperledger Fabric, similarly to FaSTr, but implements a PBFT model for transaction validation. However, Salem et al. [21] highlight that PBFT’s performance degrades as the number of orderers increases due to communication complexity and CPU overhead.

FaSTr overcomes communication overhead, allowing it to scale with the number of orderer nodes, as demonstrated in Section IV. This scalability ensures high reliability and availability, allowing the system to continue functioning even if a minority of orderer nodes fail. Further performance improvements can be achieved using high-performance hardware with a large orderer and peer node network, mitigating CPU and communication bottlenecks.

B. Blockchain for Aggregated EHR Management

While ActionEHR focuses on giving patients control over their records, MedRec [16] aggregates all medical records of a patient. As individuals visit multiple hospitals and pharmacies throughout their lives, their medical records become fragmented across different infrastructures. MedRec addresses this challenge using a permissionless Ethereum blockchain network. It implements a trustless model with the Proof of Work consensus algorithm, where nodes (miners) solve computationally intensive tasks before appending blocks. Hospitals and EHR storage providers are incentivized to participate as miners by granting them access to aggregated, anonymized data as mining rewards.

Similarly, FHIRChain [17] adapts to multiple EHR storage systems using the Ethereum blockchain network. It incorporates the HL7 Fast Healthcare Interoperability Resources

(FHIR) standard for shared clinical data. The use of a public Ethereum blockchain enables secure access to EHRs across multiple storage systems.

C. Performance Considerations in Blockchain Networks

Zhang et al. [22] evaluate the latency of Ethereum networks, demonstrating that the Ethereum currency value (gas price) significantly impacts network latency. Their study shows that average latency can range between 5 and 45 seconds, with lower latency achieved at a high gas price. While increasing gas fees reduces latency, it discourages participants from sending transactions, leading to fewer transactions and potential backlog issues.

In contrast, FaSTr employs a permissioned system and a consensus algorithm focused on error and fault tolerance, achieving an average latency of just 0.05 seconds. This low latency is critical, as multiple applications depend on FaSTr for quick and secure EHR access. Unlike Ethereum-based solutions, FaSTr does not rely on a currency for consensus, eliminating transaction validation costs and improving network throughput.

D. Comparison of Blockchain-Based EHR Solutions

The KSI (Keyless Signature Infrastructure) blockchain, developed by Guardtime and implemented in Estonia’s eHealth system, represents a unique approach to healthcare data integrity using quantum-resistant keyless signatures and distributed timestamping [23]. While KSI focuses primarily on data integrity verification without storing actual health records on-chain, FaSTr extends functionality beyond integrity to provide comprehensive access control and service management through ABE encryption and private networks. Unlike KSI’s centralized signing authorities, FaSTr’s decentralized RAFT consensus and permissioned node architecture offers greater

transparency while maintaining similar regulatory compliance capabilities.

The blockchain-based approach for e-Health data access management with privacy protection [19] and Secure cloud-based EHR system using attribute-based cryptosystem and blockchain [18] both utilize Hyperledger Fabric with ABE. However, they lack certain crucial functionalities, such as, tracking the change in access permissions and scalability. Our work builds on these efforts by integrating these missing features, enabling authorized applications to securely access EHRs in a transparent and near real-time manner.

Table I presents a comparative analysis of research efforts in this domain, listing key features of each solution [20]. The first column identifies the reviewed solutions, with our proposed solution included as the last entry. The second column specifies the blockchain technology used, while the third column details the consensus mechanisms implemented, where PoV represents Proof of Validation. The authentication and authorization techniques are outlined in the access control section, with PKC referring to Public Key Cryptography. The final five columns highlight various features offered by these solutions.

Unlike solutions designed for specific sharing workflows, FaSTr functions as a provider-agnostic middleware that facilitates secure access to EHR data regardless of the requesting entity's classification. It is also noteworthy that Ethereum-based solutions excel in tracking permission and ownership changes of assets through their currency-based validation model. However, our approach achieves the same tracking capability without relying on a cryptocurrency. Neiheiser et al. [24] discuss Ethereum's scalability challenges, noting that even with optimizations such as side-chains and rollups, the main chain still experiences significant load constraints. By avoiding a currency-based consensus, our solution improves network throughput and scalability. Rojo et al. [25] proposed a permissioned blockchain for EHR operation traceability that requires separate networks per patient, introducing cryptographic identity management complexities. FaSTr instead leverages multiple private ledgers within a single blockchain network, eliminating the overhead of managing independent networks for each patient.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents FaSTr, a solution implemented on blockchain technology that promotes fast, secure and transparent access to sensitive EHRs. The system architecture and the technologies used to implement that architecture are discussed. Finally, latency and throughput testing demonstrates that the network scales well with an increasing ledger size.

Many important applications require EHRs. We presented a solution that uses a compatibility layer to collaborate with any EHR storage and provide important functionality based on sensitive EHRs. The properties of a blockchain network are used and an immutable record of audit logs is maintained. This audit log can be used to deter bad intentions. Multiple private networks are maintained for improved robustness and

segregation of data. Finally, the API server standardises the way applications make their requests.

There are certain aspects that are still lacking from this solution, which are planned to be implemented in a future version. One of these aspect is to use formal verification techniques similar to those employed by Cuci et al. [26] for their health cloud management protocol. Their model checking approach successfully verified critical security properties for health data management and could be extended to blockchain-based systems like FaSTr to provide mathematical guarantees about security compliance properties, particularly for the smart contract components that govern access control. Ledger-specific optimizations present additional opportunities, particularly for audit logs where throughput and latency improvements could be achieved through techniques such as data sharding or modified Merkle trees. Li et al. [27] propose a framework based on blockchains and modified versions of merkel trees, for efficient storage and query of logs. These are interesting propositions and need to be studied for evaluating their effectiveness for FaSTr.

REFERENCES

- [1] D. M. Walker, W. L. Tarver, P. Jonnalagadda, L. Rambom, E. W. Ford, and S. Rahrkar, "Perspectives on challenges and opportunities for interoperability: Findings from key informant interviews with stakeholders in ohio," *JMIR Medical Informatics*, vol. 11, p. e43848, 2023.
- [2] D. Brooks, A. Gkoulalas-Divanis, G. Loukides, and B. Malin, "Patient privacy in the era of big data," *Balkan Medical Journal*, vol. 35, no. 1, pp. 8–17, 2018.
- [3] D. S. W. Ting, D. Chang, I. Y.-S. Yeo, P. A. Tambyah, S. Y. Chia, and A. Castillo-Monreal, "A review of multi-factor authentication in the internet of healthcare things," *Journal of Medical Systems*, vol. 47, no. 1, p. 61, 2023.
- [4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, 2005.
- [5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 03 2009.
- [6] A. Porat, A. Pratap, P. Shah, and V. Adkar, "Blockchain consensus: An analysis of proof-of-work and its applications," 2017.
- [7] M. Castro, B. Liskov, et al., "Practical byzantine fault tolerance," vol. 99, no. 1999, pp. 173–186, 1999.
- [8] Health Level Seven International, "Fast healthcare interoperability resources (fhir) release 5." <https://www.hl7.org/fhir/>, 2023. Official standard specification.
- [9] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3." RFC 8446, Aug. 2018.
- [10] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, and D. Cooper, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile." RFC 5280, May 2008.
- [11] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography – PKC 2011*, (Berlin, Heidelberg), pp. 53–70, Springer Berlin Heidelberg, 2011.
- [12] Zeutro, "Openabe: Open attribute-based encryption." <https://github.com/zeutro/openabe>, 2021.
- [13] HIMSS Global, "Blockchain in healthcare," 2022. Healthcare Information and Management Systems Society.
- [14] European Commission, "General data protection regulation (GDPR)," 2016. Regulation (EU) 2016/679.
- [15] A. Dubovitskaya, F. Baig, Z. Xu, R. Shukla, P. S. Zambani, A. Swaminathan, M. M. Jahangir, K. Chowdhry, R. Lachhani, N. Idnani, M. Schumacher, K. Aberer, S. D. Stoller, S. Ryu, and F. Wang, "Action-ehr: Patient-centric blockchain-based electronic health record data management for cancer care," *J Med Internet Res*, vol. 22, p. e13598, Aug 2020.

- [16] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd International Conference on Open and Big Data, OBD 2016, Vienna, Austria, August 22-24, 2016* (I. Awan and M. Younas, eds.), pp. 25–30, IEEE Computer Society, 2016.
- [17] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "Fhirchain: Applying blockchain to securely and scalably share clinical data," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 267–278, 2018.
- [18] Y. S. Hao Wang, "Secure cloud-based ehr system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, 2018.
- [19] L. Hirtan, P. Krawiec, C. Dobre, and J. M. Batalla, "Blockchain-based approach for e-health data access management with privacy protection," in *24th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2019, Limassol, Cyprus, September 11-13, 2019*, pp. 1–7, IEEE, 2019.
- [20] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, 2018.
- [21] S. Alqahtani and M. Demirbas, "Bottlenecks in blockchain consensus protocols," *CoRR*, vol. abs/2103.04234, 2021.
- [22] L. Zhang, B. Lee, Y. Ye, and Y. Qiao, "Evaluation of ethereum end-to-end transaction latency," in *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, April 2021.
- [23] Government of Estonia, "Ksi blockchain: Cyber security solutions in estonia." <https://e-estonia.com/solutions/cyber-security/ksi-blockchain/>, 2023. Official government portal for Estonia's digital infrastructure.
- [24] R. Neiheiser, G. Inácio Raimundo, L. Rech, C. Montez, M. Matos, and L. Rodrigues, "Practical limitations of ethereum's layer-2," *IEEE Access*, vol. PP, pp. 1–1, 01 2023.
- [25] J. Rojo, J. Garcia-Alonso, J. Hernandez, J. M. Murillo, and S. Helal, "Personal health trajectory traceability using blockchain technology," in *2023 IEEE International Conference on Digital Health (ICDH)*, pp. 322–324, 2023.
- [26] A. Cuci, U. Ozeer, and G. Salaün, "Modelling and verification of an application for managing sensitive health data," in *From Data to Models and Back - 11th International Symposium, DataMod 2023, Eindhoven, The Netherlands, November 6-7, 2023, Revised Selected Papers* (G. Broccia and A. Cerone, eds.), vol. 14618 of *Lecture Notes in Computer Science*, pp. 127–141, Springer, 2023.
- [27] W. Li, Y. Feng, N. Liu, Y. Li, X. Fu, and Y. Yu, "A secure and efficient log storage and query framework based on blockchain," *Computer Networks*, vol. 252, p. 110683, 2024.