A Secure and Scalable Sidechain Model for Fog Computing in Healthcare Systems

Ramzi Haraty, Ali Amhaz
Department of Computer Science and Mathematics Lebanese American University Beirut, Lebanon
Email: rharaty@lau.edu, ali.amhaz02@lau.edu

Keywords: blockchain, sidechain, fog computing, cloud computing, scalability, healthcare

Received: December 24, 2023

With the enormous amount of data produced daily, cloud and fog computing presented efficient and effective models for real-time data exchange. Nevertheless, this technology came with a cost at the security level, where it became an easy target for malicious actions that could quickly spread throughout the model. Blockchain, a recent and promising technology, has shown to be a suitable solution for securing transactions in the fog environment because of the distributed ledger structure that makes it resistant to many types of attacks. Scalability, however, introduced the main drawback of a blockchain by making it inefficient in some real-world applications, especially in the medical field, which includes a lot of data exchange. This work will suggest a scalable and secure model for fog and cloud computing in healthcare systems that depend on sidechains and the clustering of the available fog nodes. The importance of the model is highlighted, and experimental results showed promising outcomes.

Povzetek: V raziskavi je predlagan varen in razširljiv model stranske verige za megleno računalništvo v zdravstvenih sistemih, ki izboljšuje varnost in učinkovitost obdelave podatkov.

1 Introduction

The technological revolution in the last century has led to significant development in the software and hardware of information systems. For example, in the old banking systems, committing a transaction needed the manual assistance of one or more employees. However, almost all banking services nowadays are automated and allow clients to do various transactions from their homes or shops using online applications and micro hardware (i.e., electronic ships).

A significant part of this enormous development was the proliferation of the Internet of Things (IoT) devices. Those devices, which support connecting to the internet network, granted a variety of data manipulation actions such as gathering, transmitting, and processing. In addition, they helped in dispensing many human-controlled actions that consume time and resources. Many fields started using IoT devices because of their low prices and the ability to perform critical tasks without human supervision. Healthcare institutions, including hospitals, started using such devices to keep track of the patient's health records (i.e., blood pressure, temperature) and add them to the primary system for later use. Moreover, modern agriculture adopted IoT sensors connected to the internet to monitor the soil state for a better harvest.

This massive development of information systems and the amount of data produced (especially by IoT devices) brought the need to invent and enhance those systems to satisfy growing demands, including storage, processing power, and availability. Cloud computing came then to solve these problems, offering various services to facilitate data manipulation. It allowed the accomplishment of many tasks using remote servers provided by several international companies (i.e., Google and Apple). For example, cloud storage offered by Google supplied users with terabytes of storage at a low cost. Moreover, it eliminated the risk of losing the physical data resulting from any emergency. In addition, cloud services facilitate the deployment of large programs that needs many computer resources and cannot be done from the user side.

Although cloud computing presented the solution to many problems, it raised others. Such a technology consists of a centralized structure that serves millions of users in the same place, thus, causing unwanted latency in some critical applications. For example, in automated car projects, response time and availability are crucial measures that can lead to life-threatening problems. Those cars need quick operations toward any action that could happen on the roads (i.e., a child crossing the street). Based on that, fog computing [1] came as a solution to give a better performance. It provided services similar to cloud computing but with better performance. The distributed and close-to-user structure helped a lot in increasing the response time with much fewer failures.

With their distributed architecture, Fog systems were more secure than cloud computing. Nevertheless, the communication between the different fog nodes represented the main vulnerability that a hacker could exploit. For instance, a malicious transaction targeting a specific fog node could quickly spread throughout the system, making it very hard to recover to its original state. Moreover, heterogeneous models' damage levels would be much higher [2]. To tackle this problem, researchers proposed two different approaches [3, 4]. The first approach depends on preventing malicious transactions

before entering the system, while the other suggests detecting and recovering the damage.

In the case of detecting and recovering malicious actions, studies focused on building efficient and effective algorithms [2, 5] that could scan the system for unusual behavior and directly start the recovery process in the case of attacks. In [6], the researchers focused on machine learning to discover any intrusion and to recover it. On the other hand, the prevention systems mainly focused on blockchain to approve any transaction before entering the fog system. For example, in [7], the authors presented a blockchain model to protect the system and facilitate the data exchange between the clients and the doctors in the hospital.

Recently, blockchain technology has become a significant target for many applications because of its high security and the ability to control the flow of transactions to any system, especially fog networks. This combination (blockchain + fog) allowed to filter the transactions of IoT devices by forcing the proof of work and validation between different nodes [8]. Blockchain effectiveness and efficiency are measured using a set of metrics to study the scalability and compatibility with the given systems [9].

1.1 Overview

This section provides an overview of the key topics addressed in this work: IoT devices, cloud computing, fog computing, and blockchain technology.

IoT devices are connected to the internet to gather, transmit, or process data. Their affordability and ability to function in challenging environments have led to widespread adoption across various fields. In healthcare systems, for instance, IoT devices have become essential for monitoring patients' vital signs and issuing instant alerts during emergencies. These devices also streamline processes like medication delivery. A notable example is an IoT innovation designed to monitor blood glucose levels and administer insulin automatically [10-11].

Cloud computing offers remote access to resources such as storage, processing power, and networks, all delivered over the internet. This technology enhances service quality and ensures quick, reliable access to resources [12-13].

Fog computing, while similar to cloud computing, operates as a decentralized system positioned between cloud services and end users. It manages, stores, and processes data closer to the client, acting as a complementary component to traditional cloud services.

Blockchain technology, introduced by Satoshi Nakamoto in 2008, marked a turning point in data security and management. It relies on a decentralized structure, eliminating the need for a single control point. Blockchain is a distributed ledger where every user holds a copy of the chain, updated in real-time. Transactions are recorded in blocks, each containing a unique "hash" to prevent unauthorized alterations, and every block links to its predecessor to maintain the chain's integrity. When a user initiates a transaction, a new block is created, requiring approval from the majority or all users. This consensus mechanism ensures transparency and prevents tampering. Once added, a block becomes immutable, preserving the chain's integrity. To address scalability issues, sidechains were developed as an innovative solution. These smaller, independent chains interact with the main blockchain to exchange assets when needed. Sidechains reduce the time required to add new blocks and offer localized privacy. They can also have different structures and mechanisms from the main chain. The interaction between sidechains and the main chain relies on protocols such as the Two-Way Peg Protocol, which ensures secure and efficient data exchange. This protocol can be symmetric, used for chains with similar structures, or asymmetric, enabling cross-chain communication between differing systems like Bitcoin and Ethereum [14-15]. Figure 1 illustrates the architecture of blockchain, emphasizing its structure and security mechanisms, including the role of hashes and the genesis block.

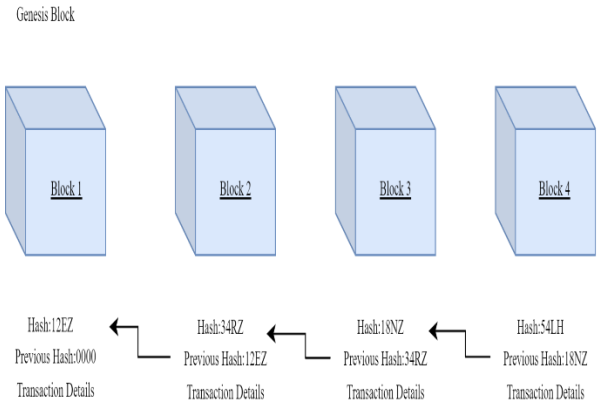

Figure 1: Blockchain architecture.

1.2 Problem statement

The fog architecture, while offering several advantages, is vulnerable to malicious transactions that can spread across the system, especially in decentralized environments like healthcare. This issue is challenging to resolve due to the interdependence of data across nodes, where a compromised node can affect others. Security solutions for fog systems generally fall into two categories: detection and recovery [2, 5, 8], which focus on identifying and mitigating malicious transactions, and prevention mechanisms aimed at blocking such transactions before they occur. Blockchain technology has been proposed as a solution to enhance security by validating transactions and providing a detailed, tamper-proof ledger [19-21]. However, as the number of fog nodes grows, scalability becomes a major concern, particularly in high-transaction environments like healthcare. This paper proposes a scalable blockchain approach for fog computing in healthcare, utilizing multiple sidechains to reduce transaction processing time and improve overall system performance.

1.3 Innovations in our work

Despite the superiority of fog computing over the cloud, it still needs a set of security restrictions to curb the vulnerabilities that take time to handle. Those vulnerabilities came from the decentralized architecture of

fog that makes it easy to spread malicious transactions between the correlated nodes. Moreover, the separated structure imposes many challenges in tracking the attacks and recovering their effects.

Blockchain, a recent and promising technology, protects the environments that include committing transactions and exchanging data. Accordingly, this work will suggest a blockchain model that secures the fog nodes using a sidechain approach. The model will cluster the nodes based on their frequency of communication to form a sidechain and ensure local privacy between them. Within the same cluster, when a node needs to commit a transaction, it is approved first by most nodes, then a new block is created on the ledger. On the other hand, when there is communication between nodes from different clusters, the transactions are committed through a main chain formed from the unclustered nodes.

Our research offers several key innovations that address limitations in existing studies:

1. Integration of Blockchain for Secure Data Sharing: We utilize blockchain technology to protect data exchanged between fog nodes, ensuring secure, immutable transactions across the system.
2. Scalability Through Sidechains: To address the scalability issues commonly associated with blockchain, we propose the formation of sidechains within clusters of fog nodes. This approach reduces the validation time for each transaction by localizing processing within smaller, manageable groups.
3. Elimination of Centralized Communication: Unlike many existing models, our approach eliminates the need for centralized communication between fog nodes in different clusters, enhancing system efficiency and reducing bottlenecks.
4. Privacy Assurance Within Clusters: By confining sensitive data and operations within individual clusters, our model ensures robust privacy for intra-cluster communication.

1.4 Organization of the paper

In section 2, the paper will review the literature review related to the topic. Then, section 3 will suggest a new model that provides a secure and scalable blockchain solution for a healthcare system. Section 4 will show and analyze the obtained results after the simulations. Finally, the conclusion will be presented in section 5.

2 Related work

This section reviews studies on cloud and fog technologies in information systems, focusing on their integration and security measures, especially blockchain and sidechain models, for improved data exchange. Security solutions are categorized into detection and recovery methods and prevention methods. Detection and recovery approaches assess attack damage and implement recovery algorithms, while prevention relies on blockchain to ensure trustworthy transactions.

2.1 Cloud and fog computing in information systems

Cloud computing has been widely adopted for its computational and financial benefits.

- Smart Cities: A hierarchical cloud model was proposed in [22], with horizontal layers for interface management and vertical layers for security and data actions, improving data availability and user interaction.
- Healthcare: In [23], cloud services were examined for strengths (accessibility) and limitations (data management), with a focus on public and private cloud models.
- Agriculture: A cloud architecture in [24] addressed traditional system inefficiencies, enhancing flexibility and enabling better weather tracking for production.

Fog computing emerged as a solution to the limitations of cloud services, particularly for IoT and real-time applications/themes:

- Characteristics of Fog: As described in [25], fog nodes bridge IoT and cloud systems, leveraging decentralized architecture, numerous nodes to prevent single-point failures, and proximity to end devices for real-time communication.
- Real-Time Applications: The authors in [26] introduced a fog model with "Third Party Memory Management" for real-time IoT requests, reducing cloud dependency.
- Autonomous Cars: In [27], fog layers integrated with machine learning (Support Vector Machine) enhanced real-time response and trajectory planning, achieving promising results in simulations.
- Detection and Recovery: Techniques like IDS-based models ([2]), node differentiation ([5]), and trust-building mechanisms ([31]) enhance database integrity and reduce attack impact.
- Prevention: Blockchain models ([35], [19]) improve secure transactions, while sidechains ([8], [17]) address scalability and performance issues.

Table 1 presents a consolidated perspective that demonstrates the evolution of cloud and fog systems, their applications, and advanced security mechanisms.

Table 1: Related works – cloud and fog computing.

Reference	Focus Area	Key Contribution	Advantages
[22]	Cloud Computing in Smart Cities	Proposed a hierarchical model for integrating cloud computing into smart cities.	- Horizontal layer for interface establishment. - Vertical layer for security and data management.
[23]	Cloud Computing in Healthcare	Studied strengths, weaknesses, and models of cloud computing in healthcare applications.	- Highlighted public vs. private clouds. - Addressed accessibility and service management differences.
[24]	Cloud in Agriculture	Introduced a cloud architecture to modernize agricultural information systems and processes.	- Improved flexibility and weather tracking. - Enhanced production chain and data storage.
[25]	Fog Computing for IoT	Explored the essential role of fog computing in IoT device development and its characteristics.	- Reduced cloud pressure. - Enhanced decentralization and real-time communication.
[26]	Fog for Real-Time Applications	Presented a fog-based model for real-time IoT response with a "Third Party Memory Management" unit.	- Differentiated between normal and real-time requests. - Reduced unnecessary cloud data uploads.
[27]	Fog in Autonomous Cars	Integrated fog computing with Support Vector Machines to enhance autonomous car communication.	- Decentralized structure for better trajectory planning. - Improved response time and reduced latency.

2.2 Fog computing in healthcare systems

Fog computing has been increasingly adopted in healthcare to address the limitations of traditional cloud-based models, especially for real-time, low-latency applications.

- Real-Time Notifications: A fog model proposed in [28] enables real-time patient health monitoring with low latency by dividing operations into four layers: sensors, fog for data analysis, cloud for storage, and a management layer for oversight.
- Comparison with Cloud: In [29], a hybrid fog-cloud architecture demonstrated 28% faster response times and enhanced security via decentralization, which mitigated certain attack risks.
- Enhanced Security: The model in [30] incorporated VM selection for better IoT management and a cryptographic mechanism for public and private key generation, achieving improved latency and performance in iFogSim simulations.

2.3 Security in fog computing models

Security mechanisms in fog systems are categorized into detection and recovery and prevention, focusing heavily on blockchain integration:

Detection and recovery:

- Data Integrity: Algorithms in [2] and [5] detect malicious transactions, assess damage, and initiate recovery using logs and IDS tools, albeit with limitations in simulation and reliance on IDS accuracy.
- Node Isolation: Models like COMMITMENT ([31]) and DataIDS ([33]) reduce attack intensity by isolating malicious nodes and utilizing dependency graphs for anomaly detection.

Prevention with blockchain:

- FogChain for Healthcare: A blockchain-based architecture in [35] improved transaction throughput and response time by 66% compared to cloud systems.
- Scalability with Sidechains: Studies in [8] and [17] introduced sidechains to enhance blockchain scalability, enabling independent yet coordinated sub-chains to handle increased user demands efficiently.

For more details on the discussed models and their metrics, refer to Table 2, which summarizes their scope, methodologies, and performance outcomes.

Table 2: Related works - healthcare and fog security.

Reference	Focus Area	Key Contribution	Advantages	Limitations
[2]	Detection & Recovery in Fog Healthcare	Introduced an IDS-based model for assessing and recovering from database attacks in fog nodes.	- Efficient damage assessment. - Distinguished bad transactions for future use.	- No real-world simulation. - IDS reliance might lead to inaccuracies.

Reference	Focus Area	Key Contribution	Advantages	Limitations
[5]	Smart City Fog Data Recovery	Differentiated private/public fog nodes and developed damage assessment and recovery algorithms.	- Focused on managing public and utility data. - Recovery algorithms fix attacked databases.	- No simulation performed.
[8]	Blockchain Scalability via Sidechains	Integrated fog with sidechains to improve blockchain scalability and processing power.	- Better transaction rates. - Effective use of processing power.	None mentioned.
[17]	Sidechain Efficiency in Fog Computing	Presented a fog-sidechain-root architecture for scalability and transaction validation.	- Improved throughput, latency, and efficiency. - Supported access control mechanism.	None mentioned.
[19]	Blockchain and IoT Integration in Fog	Proposed a blockchain-based fog model for secure IoT data exchange with performance-testing algorithms.	- Secure environment for IoT-fog data exchange. - Good performance metrics.	- Scalability not considered. - Performance tested using local parameters.
[20]	Smart Cities with Blockchain & Fog	Developed a fog-blockchain-cloud model for security and performance in smart cities.	- Enhanced response time and energy efficiency. - Adopted encryption and authentication mechanisms.	- Did not address scalability issues.
[21]	Blockchain-Based Fog Security	Introduced blockchain and encrypted signatures to secure IoT data at fog nodes.	- Enhanced defense via blockchain transparency. - Good response time and scalability.	None mentioned.
[28]	Fog in Healthcare	Proposed a fog-based notification system for real-time health records with four layers.	- Low latency and fast response time. - Reduced data overhead on the cloud.	None mentioned.
[29]	Fog vs. Cloud in Healthcare	Compared fog and cloud models in healthcare based on performance and security.	- 28% faster response time than cloud. - Effective defense via decentralized structure.	None mentioned.
[30]	Fog Security Mechanisms	Added security with patient authentication, VM-based fog node selection, and cryptographic key management.	- Improved latency and system performance. - Simulated on iFogSim software.	None mentioned.
[31]	Malicious Node Mitigation (COMMITMENT)	Proposed a fog model to isolate malicious nodes and reduce attack intensity with trust records.	- Reduced attack intensity by 66%. - Decreased latency by 15 seconds.	None mentioned.
[32]	Malicious Node Detection	Studied behavior between fog servers	- Effective for fog-based vehicle	None mentioned.

Reference	Focus Area	Key Contribution	Advantages	Limitations
		to detect unusual activity and issued alerts based on thresholds.	networks. - Real-time threat detection.	
[33]	DataIDS Model for Database Attack Detection	Introduced dependency graphs for detecting abnormal fog node behavior.	- Effective against noise, replay, and stuck-at attacks. - Adequate experimental response.	- Lacked recovery mechanisms for attacks.
[34]	Machine Learning for Fog Security	Surveyed ML/AI integration for intrusion detection and data security in fog systems.	- Highlighted ML algorithms like Random Forest, Naive Bayes, and PCA.	None mentioned.
[35]	Blockchain in Fog for Healthcare	Suggested a FogChain model integrating blockchain for real-time health data exchange.	- 66% better response time than cloud. - Suitable for healthcare IoT data.	None mentioned.

3 The suggested model

3.1 Overview

The proposed model begins by clustering fog nodes based on their communication frequency, using the k-means clustering algorithm. A parameter k specifies the minimum number of nodes in each cluster, promoting efficient and localized data exchange within the clusters. Before clustering, the 10 least-interacting fog nodes are excluded and assigned to the main blockchain (central ledger). This decision aligns with recommendations suggesting a minimum of seven nodes to form a secure blockchain, as they ensure over 66% agreement to tolerate up to two untrusted participants. To ensure the security and integrity of transactions, the model employs the SHA-256 hashing algorithm for generating transaction hashes and maintaining a tamper-proof ledger. SHA-256, a standard in blockchain systems, provides robust cryptographic security by converting input data into fixed-length, irreversible hash values. For encryption, the model utilizes Elliptic Curve Cryptography (ECC), a highly secure and computationally efficient public-key encryption algorithm. ECC ensures secure communication and data exchange between nodes while offering smaller key sizes compared to RSA, making it well-suited for resource-constrained fog computing environments. The clustering process works as follows:

1. Clustering nodes:
   - The k-means algorithm clusters fog nodes based on the frequency of communication links. Nodes that frequently exchange data are grouped into the same cluster, fostering local privacy and efficient transaction handling.
   - The number of clusters (N) is predefined, ensuring balanced cluster sizes and optimal system performance.
2. Main blockchain (Central ledger):
   - The 10 least-interacting nodes form the central ledger. These nodes maintain the global blockchain and facilitate inter-cluster communication.
3. Sidechains:
   - Each cluster forms a sidechain, where the fog nodes act as users of the chain. Transactions within a sidechain are monitored and validated by the nodes in the cluster. This design provides enhanced privacy for entities requiring secrecy, as data within a sidechain remains isolated from the central ledger.
   - The Plasma framework is used to create and manage sidechains, enabling efficient data processing and scalability. Transactions in sidechains are handled using Ethereum protocols, incorporating features like smart contracts, Decentralized Autonomous Organizations (DAO), and digital tokens for advanced functionality.

This approach offers significant advancements in security, scalability, and data integrity within the proposed system. By employing sidechains, the model ensures that sensitive data is confined and processed within specific clusters, thereby enhancing localized security and protecting information from unauthorized access or exposure. The localization of data minimizes the risk of breaches across the broader network, fostering a secure environment tailored to the needs of entities requiring heightened privacy. Additionally, the integration of sidechains addresses scalability challenges by alleviating the transaction load on the main blockchain. This design enables more efficient processing, as transactions within sidechains are handled independently of the central ledger. Consequently, this improves resource allocation and significantly reduces latency, ensuring that the system can accommodate increased demands without compromising performance.

Furthermore, the adoption of SHA-256 hashing and ECC provides robust mechanisms for consensus and

security. The SHA-256 hashing algorithm ensures the integrity of transactions by generating unique, immutable hash values, effectively preventing unauthorized tampering or data corruption. Simultaneously, ECC encryption secures data exchanges between nodes, offering a high level of cryptographic protection with smaller key sizes, which is particularly advantageous in resource-constrained environments. Together, these cryptographic techniques fortify the system against potential vulnerabilities, ensuring that all transactions are authenticated and secure. Figure 2 illustrates the detailed process, showcasing the steps involved in clustering, the formation of sidechains, and their seamless integration with the main blockchain. This visual representation highlights the model's ability to deliver a secure, scalable, and efficient architecture, specifically designed to meet the stringent requirements of applications such as healthcare systems.

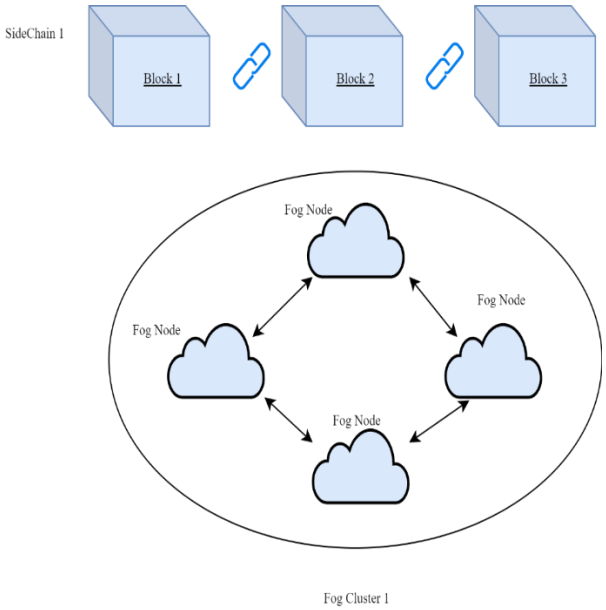

Figure 2: A Fog cluster.

Each block consists of a set of parameters that are essential for its functioning (see figure 3):

Block ID (same as Transaction ID): is a unique attribute that refers to a specific block.

Hash: As stated earlier, each block has a unique hash that ensures that the block is not altered or modified by any unauthorized users.

Previous Block Hash: This parameter creates the ledger structure by linking the blocks. When a block is modified, it will change its hash, thus making the link inconsistent. Thus, it is considered the primary defense mechanism in the blockchain.

Encrypted content: The healthcare data is encrypted and saved in this block attribute.

Signature: This attribute links the creation of the block to a specific entity without knowing the actual identity.

Timestamp: Records the date of the creation of the block.

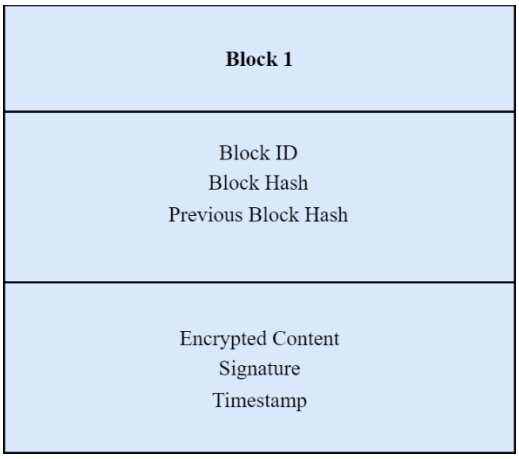

Figure 1: Block structure.

The main chain will be formed of the fog nodes in the system that do not belong to any cluster or preselected ones. This chain will be responsible for exchanging the transactions between the different sidechains. The communication between the sidechains and the main chain is performed through a protocol called a 2-way peg, which ensures the integrity of the data transmitted between them. This protocol is the most crucial component in cross-chain data exchange because it ensures the proper communication and information transfer from one chain to the other. Moreover, it obliges both chains through a digital contract to abide by the confirmed data transactions. Like side chain creation, the main chain involves the same features the plasma framework provides. The main chain is built based on the data provided for the ten selected nodes and allows the cross-chain data exchange using the 2-way-peg protocol.

Coordinator:
The coordinator presented in the model is a computer program responsible for the encryption/decryption process to ensure that the data is only accessible by authorized users. It plays a role in helping the recipient node to find the intended data after being uploaded to the main chain.

Encryption/Decryption process:
This process is done based on the private and public key concepts that can protect the data from unauthorized access. The public key will be shared between all the nodes and has the role of encrypting the data. On the other hand, the private key is given to specific nodes with the right to decrypt the data (see figure 4).

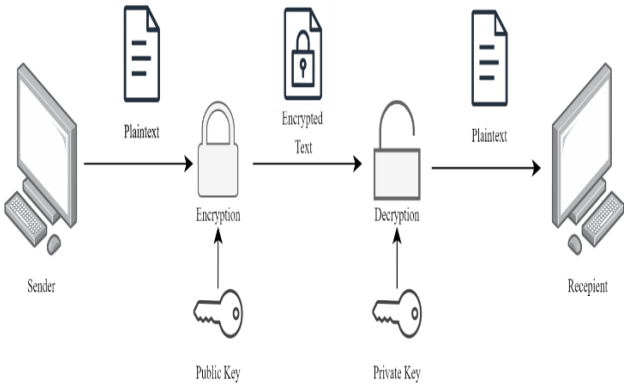

Figure 2: Public and Private Key Concept.

First, the node uploading the data will encrypt it using the public key provided to all the nodes (sent by the coordinator). In the next step, the coordinator will follow a predefined set of privileges to send the private key to the authorized recipients to ensure their right to access the data.

Implementing the different chains in the model is done through the plasma framework [37] that allows the available active nodes to be divided into chains. The framework creates the chains using an interface that permits adding the nodes to each chain (including the main chain) and setting the different attributes like proof of work, time to approve a block, and the data exchanged. Moreover, this software specifies the data exchange scheme that will select the data used in the model and the flow of data and transactions between the different entities in the simulation. The importance of this framework is in the 2-way-peg protocol, which ensures a smooth data transfer between any two chains. This protocol works by locking the data on the sending chain first, and then using a smart contract, the data will be transferred without any modification to its intended destination.

➢ **Pseudo code:**

This subsection will present the pseudo-code for the creation of the whole model (see figure 5). It will include the clustering method, the creation of the chains, and the data mapping to the chains.

1. *k* // minimum number of nodes per cluster
2. *N* // number of clusters
3. *D* //data
4. Exclude the ten least active nodes from *D*
5. Run k-means (*D*, *k*, *N*)
6. Establish SideChains + MainChain
7. Map the data to the different chains;
8. Run the transactions

The pseudocode outlines a process for clustering nodes and managing data with a focus on minimizing network load and ensuring efficient transaction handling. Initially, a minimum number of nodes per cluster (denoted by *k*) and the total number of clusters (*N*) are defined. The data set *D* is processed by first excluding the ten least active nodes, likely to enhance performance by removing underutilized or irrelevant nodes. The k-means algorithm is then applied to partition the data into *N* clusters, with k representing the minimum number of nodes per cluster. Following clustering, the system establishes both SideChains and a MainChain to handle and validate transactions securely. The data is then mapped to the appropriate chains based on the clustering results, and transactions are executed across the system, ensuring that the data is processed efficiently within the established structure.

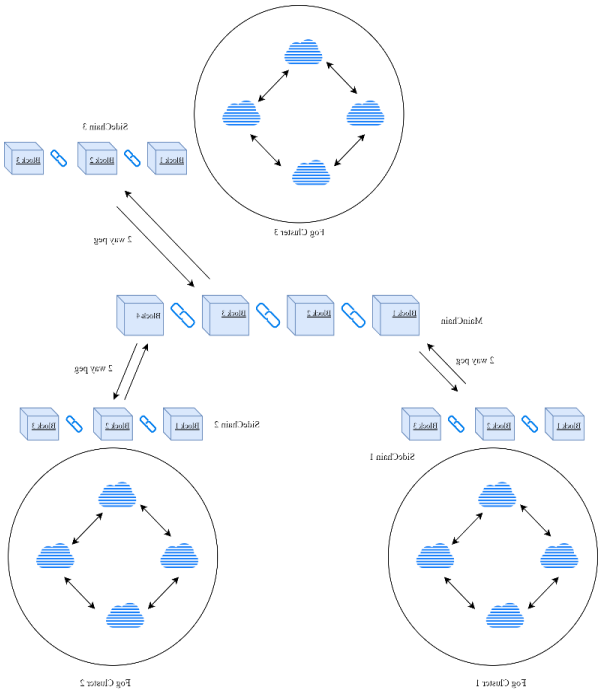

Figure 5: The complete model.

3.2 Model functionality

3.2.1 In the sidechain of a specific cluster

When a transaction is committed in a given cluster, a block is created containing the encrypted data with a specific hash pointing to the previous block's hash. This process uses the plasma framework that links the different blocks and updates the ledger at each node within the cluster. This block is then sent to the other fog nodes in the same cluster to check whether it is accepted (All/Majority of the users). If the approval is achieved, then the block is added to the sidechain of all the users (in the same cluster); else, it will be deleted.

➢ **Pseudo code:**

The pseudo-code below will present the sidechain's functionality in a specific cluster, including the approval mechanism and the node creation process.

```
1.  Di → database of fog node i
2.  Tj → Transaction j
3.  Li → SideChain ledger of node i
4.  Bj → Block of transaction j
5.  If Tj.is_committed(Di)
6.          Bj.Create();
7.  If a majority of nodes approve
8.          Bj.Add_To(Li);
9.  Else
10.         Bj.Remove();
```

The pseudocode describes the functionality of a sidechain within a specific cluster, focusing on the approval mechanism and node creation process for transactions. Each fog node *i* has its own database (*Di*) and sidechain ledger (*Li*). When a transaction *Tj* occurs, it is first checked to see if it is committed to the node's database (*Tj*.Is_committed(*Di*)). If the transaction is committed, a new block *Bj* is created to represent *Tj*. Next, the block is subject to approval by a majority of the nodes in the cluster. If the majority approves, the block is added to the node's sidechain ledger (*Li*). If the approval is not obtained, the block is removed from the process. This ensures that only validated transactions are recorded on the sidechain, maintaining the integrity of the distributed ledger within the cluster.

3.2.2 Exchanging data between clusters

When two fog nodes in two different clusters need to exchange information, the data will be sent first to the main chain through the 2-way-peg protocol. This protocol is responsible for ensuring the integrity of the data while being transferred from one chain to the other. It will lock the data in the side chain and wait until the smart contract is initiated to transfer it from one chain to the other. Then, the fog node could get the intended data from the main chain to the targeted cluster (also using the 2-way-peg). It is worth mentioning here that at every included chain, a new block will be added to the ledger.

➢ **Pseudo code:**

This part shows the functionality of the whole model when communication between the different clusters is involved. It will include how the data is transferred from one block to the other and blocks creation locations.

```
1.  Di → database of fog node i
2.  Dy → database of fog node y
3.  Tj → Transaction j //sending data to main chain
4.  Tk → Transaction k //receiving data from the
    main chain
5.  Tm → Transaction m // posting data on the main
    chain
6.  Lm → Main Ledger
7.  Li → SideChain ledger of node i
8.  Ly → SideChain ledger of node y
9.  Bj → Block of transaction j
10. Bk → Block of transaction k
11. Bm → Block of transaction m
12. If Tj.is_Commited(Di)
13.         Bj.Create();
14. If a majority of nodes approve (Cluster i)
15.         Bj.Add_To(Li);
16. Else
17.         Bj.Remove();
18.         End_Process();
19. If Tm.is_Commited(Di)
20.         Bm.Create();
21. If a majority of nodes approve (Main Chain)
22.         Bm.Add_To(Lm);
23. Else
24.         Bm_Remove();
25.         End_Process();
26. If Tk.is_Commited(Di)
27.         Bk.Create();
28. If a majority of nodes approve (Cluster y)
29.         Bk.Add_To(Ly);
30. Else
31.         Bk.Remove();
32.         End_Process();
```

This pseudocode describes the process of transferring data between different clusters, focusing on transaction creation, approval, and the movement of data across sidechains and the main ledger. First, transactions *Tj*, *Tk*, and *Tm* are identified, with *Tj* representing the transaction sending data to the main chain, *Tk* receiving data from the main chain, and *Tm* posting data on the main chain. When a transaction, such as *Tj*, is committed in the database of a fog node *i* (*Di*), a corresponding block *Bj* is created. The block is then subject to approval by a majority of nodes in the cluster *i* (lines 14–17). If approved, it is added to the sidechain ledger *Li* of node *I*; otherwise, it is removed. Similarly, when *Tm* is committed, a block *Bm* is created and added to the main ledger *Lm* after approval by the majority of nodes in the main chain (lines 20–24). For transactions like *Tk*, when data is received from the main chain, the transaction is committed in node *i*, a block *Bk* is created, and it is added to the sidechain ledger *Ly* of node *y* after receiving approval from the majority of nodes in cluster *y* (lines 26–32). This process ensures that data flows between clusters in a secure and validated manner, with transaction blocks only being added to the respective ledgers after proper approval.

3.3 An Example: Demonstrating intercluster and intracluster communication

This section provides an example to illustrate how the proposed model facilitates both intercluster and intracluster communication. By exploring real-world scenarios, we demonstrate the functionality of the system, highlighting its mechanisms for secure data exchange within a single cluster and across multiple clusters.

3.3.1 Data exchange within a cluster

To understand intracluster communication, consider the case of a patient named Jad. After certain medical procedures, the responsible department uploaded Jad's data to the sidechain, making it accessible to other authorized entities within the same cluster. Table 3 displays the patient's record.

Table 3: Patient Record.

ID	Name	Temp	Weight
1234	Jad	38	70

The process begins with the fog nodes responsible for data management creating a new block containing Jad's encrypted data and an associated hash. This block is proposed to the cluster's distributed ledger. Once a majority of the fog nodes approve the block, it is added to the ledger. The approval and voting process is orchestrated by the plasma framework, which ensures consensus among the nodes and oversees decisions regarding the block creation. Following this successful transaction, the data is now securely stored on the sidechain. Suppose a doctor operating from another fog node within the same cluster needs access to Jad's data. The doctor must request permission through the coordinator, who manages access control. The coordinator provides the necessary private decryption key, enabling the requesting node to decrypt and access the specific content on the sidechain's distributed ledger. This example illustrates the robust and secure mechanisms employed by the model to facilitate data sharing within a single cluster.

3.3.2 Data exchange between clusters

To explain intercluster communication, consider a scenario involving a patient named Sami. Sami enters the emergency department, part of fog cluster 1, and his initial record is created as shown in Table 4.

Table 4: Another patient record.

ID	Name	Temp	Weight
4321	Sami	37	75

After being transferred to the X-ray department, which belongs to fog cluster 2, the doctors in the new department request access to Sami's medical data from fog cluster 1. In this case, fog cluster 1 initiates a request to create a new block on the main chain containing the relevant data. The coordinator plays a crucial role in facilitating this operation by managing the exchange of cryptographic keys between the two clusters. Using the 2-way-peg protocol, the sidechain of fog cluster 1 transfers the data to the main chain. Subsequently, a cross-chain operation enables fog cluster 2 to retrieve the data from the main chain and integrate it into its sidechain ledger. Finally, the coordinator provides the private decryption key to authorized personnel, allowing them to access the decrypted data. This example highlights the seamless and secure data exchange between clusters, demonstrating the efficiency of the proposed model in handling intercluster communication.

4 Experimental results

This part of the paper will examine the experimental results after the simulation of the suggested model. It will first introduce the simulation software used to get the results. Then it will go over the dataset. Finally, it will show the achieved results.

4.1 Simulation software

The Plasma framework was utilized to simulate the functionality of the proposed model. Plasma supports InterLedger Protocols (ILP), which facilitate seamless data exchange between the main blockchain and sidechains. The framework allows for efficient transaction handling while enabling integration with smart contracts, critical for secure and autonomous operations. In addition, Truffle and Ganache were employed for smart contract development and testing. Truffle provided the necessary tools to compile and deploy the contracts, while Ganache simulated the blockchain environment locally, ensuring smooth testing of transaction workflows before deployment.

4.2 Dataset

For this study, a simulated dataset representing healthcare systems was used to evaluate the model. The dataset included medical reports, such as X-ray images, prescriptions, and text-based patient records. These records reflect real-world scenarios where sensitive information must be securely managed and quickly accessed.

- Block Structure: Each block in both the sidechains and mainchain could contain only one medical report.
- Block Size: The maximum block size was restricted to 2.53 KB to align with typical constraints in blockchain-based systems. This assumption allowed for testing the system's ability to manage fine-grained data distribution effectively and securely.

4.3 Hardware setup

The simulation was performed on a system with the following specifications:

- Processor: Intel Core i7, 2.30 GHz
- RAM: 8 GB
- Storage: 1 TB HDD
- Operating System: Windows 64-bit

The hardware setup ensured sufficient resources to handle the computational demands of running the Plasma framework and its associated tools.

4.4 Performance metrics and results

The performance of the proposed model was evaluated based on the following key metrics:

- Transactions Per Minute (TPM): The number of transactions processed in a given time, reflecting the system's throughput.
- Latency: The time taken for a transaction to be verified and added to a block.
- Response Time: The delay experienced by users when querying data or submitting a transaction.
- Data Exchange Size: The amount of data exchanged between sidechains and the main blockchain.

The following flowchart provides a clear visualization of the experimental process, from data preparation to performance evaluation:

[Start] --> [Dataset Preparation] --> [Simulation Environment Setup]
--> [Deploy Clustering Algorithm] --> [Form Sidechains and Mainchain]
--> [Simulate Transactions on Plasma Framework]
--> [Evaluate Metrics: TPM, Latency, Response Time, Data Exchange Size]
--> [Analyze and Compare Results with Default Blockchain Model] --> [End]

This flowchart illustrates the systematic steps taken to implement and evaluate the proposed model, ensuring a structured approach to testing its capabilities.

4.5 Performance

This subsection will focus on presenting the results achieved after the simulation of the Plasma framework. The metrics "Transaction per minute," "Latency," and "Response time" will be used to study the performance and scalability of the model in comparison with the default blockchain approach. Moreover, the data exchange size will be tracked as well. In addition, the achieved results will be discussed and analyzed.

The experimental results of the proposed model are presented in detail, focusing on a comparative analysis of performance between a single blockchain and a model incorporating five sidechains. These results are illustrated in Figure 6, which highlights the number of committed transactions as a function of time. The graph demonstrates a clear advantage in productivity for the model with five sidechains compared to the single-chain approach.

The sidechain model exhibits a significantly higher number of committed transactions over the same time period, indicating enhanced throughput and overall efficiency. This improvement can be attributed to several key factors: 1) The clustering approach used in the sidechain model reduces the number of nodes participating in each cluster. This localization simplifies the approval process for transactions, resulting in faster consensus, 2) Smaller clusters decrease network congestion, as communication is confined to a limited subset of nodes, enhancing the speed and reliability of data exchange, 3) By distributing transaction activity across five sidechains, the model alleviates the workload on individual chains. This parallelization ensures smoother data exchange, even under high transaction loads, and 4) Each sidechain operates independently but adheres to the same security protocols, maintaining data integrity while improving processing times.

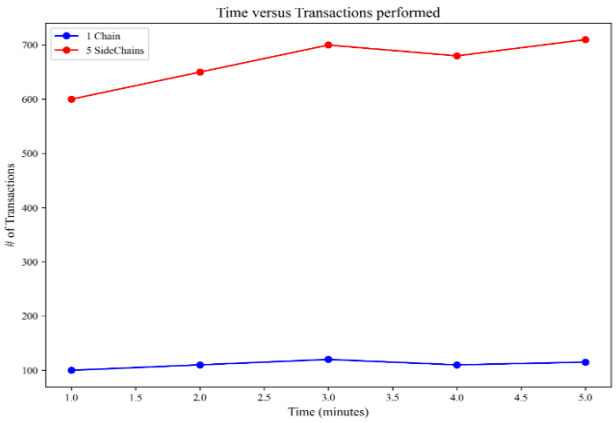

Figure 6: Time versus transactions performed.

Figure 7 examines a critical performance metric for blockchain systems: latency, defined as the time elapsed between the submission of a transaction and its final acceptance or rejection on the blockchain. This metric provides insight into the responsiveness and efficiency of the blockchain model. The results depicted in the graph highlight a noticeable reduction in latency for the proposed five-sidechain model compared to the traditional single-chain model. The decreased latency can be attributed to the introduction of sidechains significantly lowers the number of transactions each chain must handle. As a result, the time required to process and validate a transaction—whether to approve or reject it—is minimized, leading to faster transaction finalization. This is also attributed to efficient resource allocation where the smaller clusters created by the sidechain model distribute the computational workload across multiple chains, reduced the processing pressure on any single chain. This distribution allows for smoother data exchange and

quicker transaction handling, despite the maximum block size of 2.53 KB.

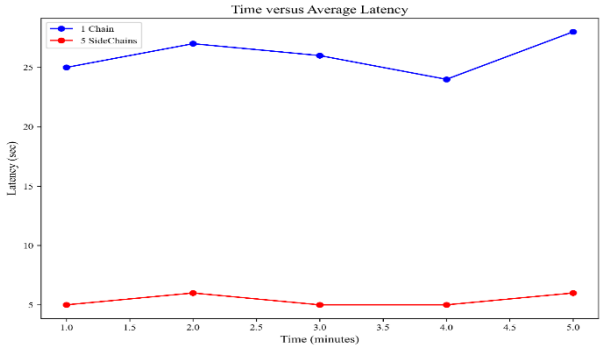

Figure 7: Time versus average latency.

The simulation also examined the data exchange size within the proposed model, which refers to the volume of data processed through the blockchain via transactions. Figure 8 illustrates the amount of data entering the two blockchain models over time. The graph clearly demonstrates that the proposed model significantly outperforms the traditional single-chain architecture, allowing a greater volume of data to be processed. The reduced transaction latency in the proposed model facilitated faster data approval, enabling higher throughput, and the enhanced architecture processed more transactions in less time, leading to greater data flow into the blockchain.

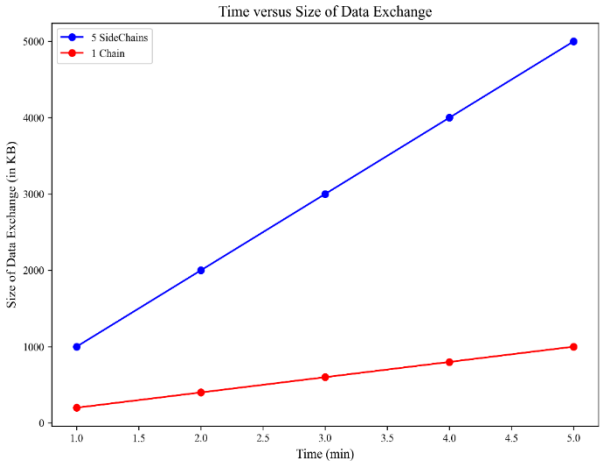

Figure 8: Time versus size of eata exchange.

Figures 9 and 10 examine the scalability of the proposed model by increasing the number of clusters to 30, assessing its effectiveness in adding blocks to the ledgers. The results clearly demonstrate that a higher number of clusters significantly enhances system performance by increasing both data exchange capacity and the number of transactions processed. This improvement highlights the model's scalability and its suitability for handling complex fog computing systems with a large number of nodes. The clustering technique eliminates the limitations of the single-chain structure, which often struggles with scalability due to the extensive validation required across numerous nodes. By leveraging sidechain architecture, the model reduces the effort involved in block creation and improves throughput. Furthermore, the sidechains operate without requiring constant connectivity to the main chain, effectively mitigating bottleneck issues and ensuring smooth data flow.

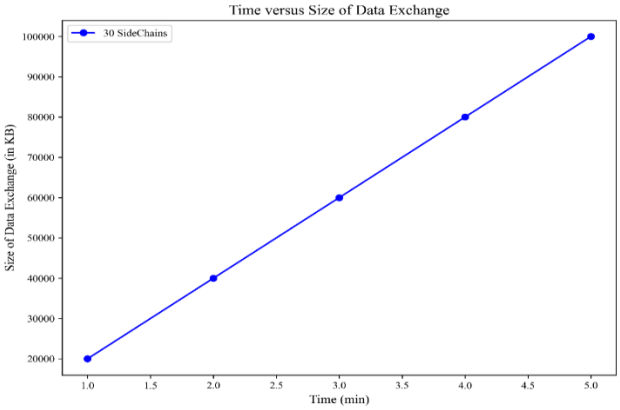

Figure 9: Time versus size of data exchange.

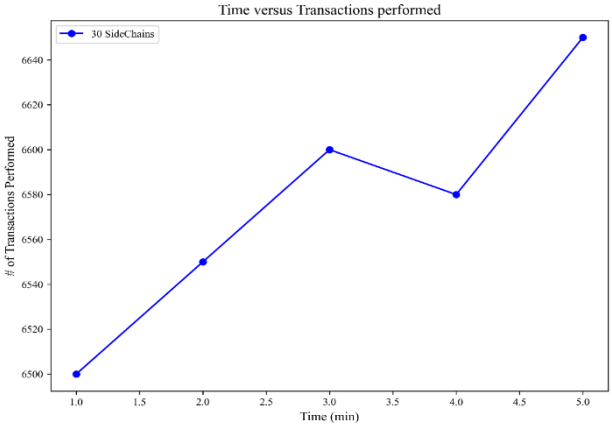

Figure 10: Time versus transactions performed.

We conducted a comparative analysis between our proposed model and the system presented in [35], which introduced a novel architecture designed to secure healthcare records in a fog computing environment. The referenced model emphasized the implementation of an additional fog layer to enhance system security, improve throughput, and deliver real-time services. To ensure a fair comparison, we simulated the same block sizes used in [35], specifically 1 KB and 0.1 KB. Figure 11 provides a visual comparison of throughput, measured in Transactions Per Second (TPS), between our model, utilizing 30 clusters, and the system from [35]. The bar graph highlights the clear superiority of our approach in handling a higher number of transactions. This performance advantage can be attributed to the clustering algorithm employed in our model, which organizes the system into sidechains. This approach significantly reduces the validation time required for each transaction by distributing the workload across multiple clusters, thereby enhancing overall throughput. The results underline the efficiency of our model in managing transactional processes within a fog computing

environment, especially when compared to traditional architectures.

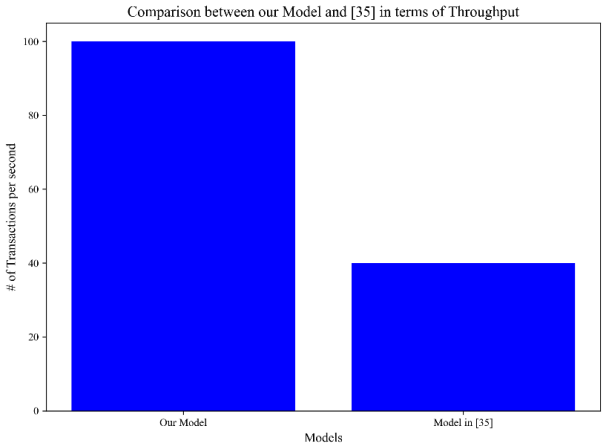

Figure 11: Comparison between our model and [35] in terms of throughput.

In addition to evaluating throughput, we also compared the latency performance of our model with the results presented in [35] as well as the latency of a traditional cloud-based model. Figure 12 clearly demonstrates the superior latency performance of our approach. The key factor contributing to this advantage is the use of sidechains, each consisting of fewer nodes compared to the single chain architecture. With a smaller number of nodes in each sidechain, the transaction validation process becomes significantly faster, as fewer participants are involved in the approval process. This reduction in the number of nodes per sidechain leads to quicker transaction commitment, thereby minimizing latency. The results highlight how our model's structure—leveraging sidechains and clustering—enables more efficient processing and faster response times, outperforming both the system in [35] and the conventional cloud model in terms of transaction validation speed.

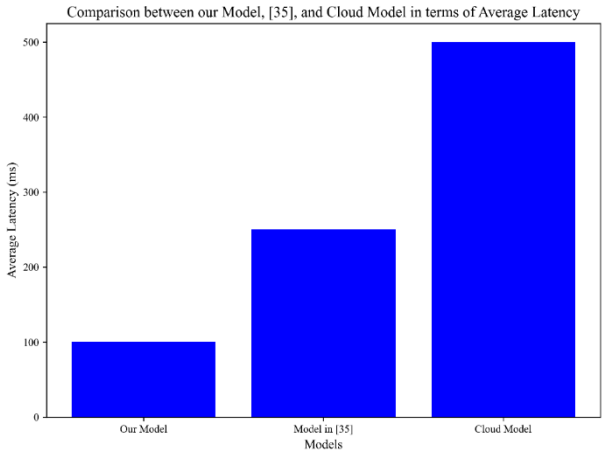

Figure 12: Comparison between our model, [35], and the cloud in terms of latency.

5 Conclusion and future work

Cloud computing has significantly enhanced information systems by providing greater processing power, flexibility, and storage. However, the increasing complexity of applications and the rise of IoT devices have outgrown the cloud's capabilities, leading to the introduction of the fog layer [38]. Fog computing offers faster response times and reduced latency, but remains vulnerable to malicious transactions between nodes [39]. This work proposes a fog model using a clustering algorithm to create sidechains for transaction monitoring, addressing scalability challenges and improving blockchain technology's limitations.

Future work could incorporate real-world data into simulations for more accurate results and identify issues not visible with synthetic data. Exploring different clustering algorithms or metrics, such as communication frequency, could provide new insights. Additionally, using alternative blockchain metrics, like response time, could offer a more comprehensive evaluation of the model's performance.

References

[1] Elhadad, A., Alanazi, F., Taloba, A. I., & Abozeid, A. (2021), Fog Computing Service in the Healthcare Monitoring System for Managing the Real-time Notification. *Journal of Healthcare Engineering*, 2022(1), 5337733, pp. 1–11. https://doi.org/10.1155/2022/5337733.

[2] Alazeb, A., & Panda, B. (2019), Ensuring Data Integrity in Fog Computing Based Health-Care Systems, *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, vol 11611, Springer, pp. 63–77. https://doi.org/10.1007/978-3-030-24907-6_6.

[3] Jiang, Y., Wang, C., Wang, Y., & Gao, L. (2019), A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management. *Sensors*, 19(9), 2042. https://doi.org/10.3390/s19092042.

[4] Yi, S., Qin, Z., & Li, Q. (2015), Security and Privacy Issues of Fog Computing: A Survey, *Wireless Algorithms, Systems, and Applications*, vol 9204, Springer, pp. 685–695. https://doi.org/10.1007/978-3-319-21837-3_67.

[5] Alazeb, A., & Panda, B. (2019), Maintaining Data Integrity in Fog Computing Based Critical Infrastructure Systems. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, Las Vegas, NV, USA, pp. 40-47. https://doi.org/10.1109/csci49370.2019.00014.

[6] Amoli, P., Plosila, J., Tenhunen, H., Hämäläinen, T., & Hosseinpour, F. (2016), An Intrusion Detection System for Fog Computing and IoT-based Logistic Systems using a Smart Data Approach, *International Journal of Digital Content Technology and its Applications*, 10.

[7] Ismail, L., & Materwala, H. (2020), Blockchain Paradigm for Healthcare: Performance Evaluation. *Symmetry*, 12(8), 1200. https://doi.org/10.3390/sym12081200.

[8] Ziegler, M. H., Grossmann, M., & Krieger, U. R. (2019). Integration of Fog Computing and

Blockchain Technology Using the Plasma Framework, *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Seoul, Korea (South), pp. 120-123. https://doi.org/10.1109/bloc.2019.8751308.

[9] Blockchain Metrics | Arcitura Patterns. (n.d.), *ARCITURA*. Retrieved on January 15, 2025, from http://patterns.arcitura.com/blockchain-patterns/blockchain-metrics.

[10] Gia, T. N., Ali, M., Dhaou, I. B., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2017), IoT-based Continuous Glucose Monitoring System: A Feasibility Study, *Procedia Computer Science*, 109, pp. 327–334. https://doi.org/10.1016/j.procs.2017.05.359.

[11] Kua, J., Loke, S., Arora, C., Fernando, N., & Ranaweera, C. (2021), Internet of Things in Space: A Review of Opportunities and Challenges from Satellite-Aided Computing to Digitally-Enhanced Space Living. *Sensors*, 21(23), 8117. https://doi.org/10.3390/s21238117.

[12] How Google Workspace Uses Encryption to Protect Your Data. (n.d.). *Google*. Retrieved on January 15, 2025, from https://services.google.com/fh/files/misc/google-workspace-encryption-wp.pdf.

[13] Fatoum, H., Hanna, S., Halamka, J. D., Sicker, D. C., Spangenberg, P., & Hashmi, S. K. (2021), Blockchain Integration with Digital Technology and the Future of Health Care Ecosystems: Systematic Review, *Journal of Medical Internet Research*, 23(11), e19846. https://doi.org/10.2196/19846.

[14] A Cautionary Tale: How a Bug in Dropbox Permanently Deleted 8,000 of My Photos. *PetaPixel*. Retrieved on January 15, 2025 from https://petapixel.com/2014/07/31/cautionary-tale-bug-dropbox-permanently-deleted-8000-photos/.

[15] Nakamoto, S., (2008), Bitcoin: A Peer-to-Peer Electronic Cash System, Available at *SSRN*: https://ssrn.com/abstract=3440802 or http://dx.doi.org/10.2139/ssrn.3440802.

[16] Foley, S., Karlsen, J. R., & Putniņš, T. J. (2019), Sex, Drugs, and Bitcoin: How Much Illegal Activity is Financed through Cryptocurrencies? *The Review of Financial Studies*, 32(5), pp. 1798–1853. https://doi.org/10.1093/rfs/hhz015.

[17] Jiang, Y., Wang, C., Wang, Y., & Gao, L. (2019), A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management, *Sensors*, 19(9), 2042. https://doi.org/10.3390/s19092042.

[18] Back, S.A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A.K., Poelstra, A., & Timón, J, (2014), Enabling Blockchain Innovations with Pegged.

[19] Alam, T. (2019), IoT-Fog: A Communication Framework using Blockchain in the Internet of Things, *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7(6). https://doi.org/10.36227/techrxiv.12657200.v1.

[20] Singh, P., Nayyar, A., Kaur, A., & Ghosh, U. (2020), Blockchain and Fog-Based Architecture for Internet of Everything in Smart Cities, *Future Internet*, 12(4), 61. https://doi.org/10.3390/fi12040061.

[21] Ngabo, D., Wang, D., Iwendi, C., Anajemba, J. H., Ajao, L. A., & Biamba, C. (2021), Blockchain-Based Security Mechanism for the Medical Data at Fog Computing Architecture of Internet of Things. *Electronics*, 10(17), 2110. https://doi.org/10.3390/electronics10172110.

[22] Agarwal, N. & Agarwal, G. (2017), Role of Cloud Computing in Development of Smart City. *International Journal of Science Technology & Engineering*, National Conference on Road Map for Smart Cities of Rajasthan, pp. 228-232.

[23] Devadass, L., Sekaran, S. S., & Thinakaran, R. (2017), Cloud Computing in Healthcare, *International Journal of Students' Research in Technology & Management*, 5(1), pp. 25–31. https://doi.org/10.18510/ijsrtm.2017.516.

[24] Choudhary, S., Jadoun, R., & Mandoria, H. (2016), Role of Cloud Computing Technology in Agriculture Fields, *Computer Engineering and Intelligent Systems*, vol 7, no. 3, pp. 1-7.

[25] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012), Fog Computing and its Role in the Internet of Things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* - MCC '12, ACM, Helsinki, Finland, pp. 13-16. https://doi.org/10.1145/2342509.2342513.

[26] Ema, R. R., Islam, T., & Ahmed, M. H. (2019), Suitability of Using Fog Computing Alongside Cloud Computing, *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Kanpur, India, pp. 1-4. https://doi.org/10.1109/icccnt45670.2019.8944906.

[27] Du, H., Leng, S., Wu, F., Chen, X., & Mao, S. (2020), A New Vehicular Fog Computing Architecture for Cooperative Sensing of Autonomous Driving, *IEEE Access*, vol 8, pp. 10997–11006. https://doi.org/10.1109/access.2020.2964029.

[28] Elhadad, A., Alanazi, F., Taloba, A. I., & Abozeid, A. (2022), Fog Computing Service in the Healthcare Monitoring System for Managing the Real-Time Notification, *Journal of Healthcare Engineering*, vol 1, pp. 1–11. https://doi.org/10.1155/2022/5337733.

[29] Quy, V. K., Hau, N. V., Anh, D. V., & Ngoc, L. A. (2021), Smart Healthcare IoT Applications Based on Fog Computing: Architecture, Applications, and Challenges, *Complex & Intelligent Systems*, vol 8, pp. 3805-3815. https://doi.org/10.1007/s40747-021-00582-9.

[30] Awaisi, K. S., Hussain, S., Ahmed, M., Khan, A. A., & Ahmed, G. (2020), Leveraging IoT and Fog Computing in Healthcare Systems, *IEEE Internet of Things Magazine*, 3(2), pp. 52–56. https://doi.org/10.1109/iotm.0001.1900096.

[31] Al-khafajiy, M., Baker, T., Asim, M., Guo, Z., Ranjan, R., Longo, A., Puthal, D., & Taylor, M. (2020), COMMITMENT: A Fog Computing Trust Management Approach, *Journal of Parallel and*

Distributed Computing, 137, pp. 1–16. https://doi.org/10.1016/j.jpdc.2019.10.006.

[32] Gu, K., Dong, X., & Jia, W. (2022), Malicious Node Detection Scheme Based on Correlation of Data and Network Topology in Fog Computing-Based VANETs, *IEEE Transactions on Cloud Computing*, 10(2), pp. 1215–1232. https://doi.org/10.1109/tcc.2020.2985050.

[33] Fantacci, R., Nizzi, F., Pecorella, T., Pierucci, L., & Roveri, M. (2019), False Data Detection for Fog and Internet of Things Networks, *Sensors*, 19(19), 4235. https://doi.org/10.3390/s19194235.

[34] Ahanger, T. A., Tariq, U., Ibrahim, A., Ullah, I., Bouteraa, Y., & Gebali, F. (2022), Securing IoT-Empowered Fog Computing Systems: Machine Learning Perspective. *Mathematics*, 10(8), 1298. https://doi.org/10.3390/math10081298.

[35] Mayer, A. H., Rodrigues, V. F., Costa, C. A., Righi, R. da, Roehrs, A., & Antunes, R. S. (2021), Fogchain: A Fog Computing Architecture Integrating Blockchain and Internet of Things for Personal Health Records. *IEEE Access*, vol 9, pp. 122723–122737. https://doi.org/10.1109/access.2021.3109822.

[36] Na, S., Xumin, L., & Yong, G. (2010), Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm, *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, IEEE, Jian, China, pp. 63-67. https://doi.org/10.1109/iitsi.2010.74.

[37] Poon, J. (2017), Plasma: Scalable Autonomous Smart Contracts.

[38] Dhaini, M., Jaber, M., Fakhereldine, A., Hamdan, S. and Haraty, R. A. (2021), Green Computing Approaches – A Survey. *Informatica* 45 (2021), pp. 1-12. https://doi.org/10.31449/inf.v45i1.2998.

[39] Alasady, A. S., Awadh, W. A., and Hashim, M. S. (2023), Non-Dominated Sorting Genetic Optimization-Based Fog Cloudlet Computing for Wireless Metropolitan Area Networks, *Informatica* 47 (2023), pp. 1–8, https://doi.org/10.31449/inf.v47i10.5118.