Blockchain-Based IoT Framework with In-Depth Security Analysis and Performance Benchmarks for Real-World Healthcare Fog Applications

Ehsan Dastani, Mahdi Akbari Zarkesh, Mahdi Davoodzadeh, Bardia Safaei*, and Ali Movaghar

Abstract—Resource restrictions in healthcare IoT devices necessitates cloud services for data processing, which is prone to single point of failure, delays, and security risks. Mean- while, exploiting blockchain can enhance security, and data ownership while providing decentralization. Optimal solutions must guarantee confidentiality, data integrity, and access control while also being cost-effective, scalable, and compatible with existing systems. Therefore, we present EdgeLinker, a blockchain-based IoT framework that uses Proof-of-Authority consensus, integrates smart contracts for access control, and employs advanced cryptographic algorithms for secure data communication between edge and fog devices. While implementing a real-world fog testbed for performance evaluations, this paper conducts a scrutinized security analysis. This analysis includes but is not limited to Sybil attacks, consensus manipulation, replay attacks, 51% attacks, traffic analysis, message spoofing, unauthorized access, Distributed Denial of Service (DDoS), and transaction malleability. EdgeLinker has shown enhanced security and privacy at reasonable costs, making it a cost-effective and practical solution for healthcare fog applications. Compared to the state-of-the-art, EdgeLinker achieves a 35% improvement in data read time without significant changes in blockchain write-time and provides better throughput in both reading and writing transactions. Furthermore, it shows energy and resource consumption improvements and channel latency in secure and non-secure modes.

Index Terms—IoT, Network, Fog Computing, Healthcare, Blockchain, Smart Contracts, Security, Privacy.

I. Introduction

Nowadays, the internet of things (iot) is recognized as the next era of communication, as it enables physical objects to create, receive, and exchange data seamlessly. Iot applications focus on automating various tasks and aim to empower inanimate physical objects to operate without human intervention [1]–[3]. Due to the high potential of iot in diverse applications, the variety of these devices is rapidly increasing [4], [5]. In many of these applications, e.g., vehicular ad-hoc networks (vanet), flying ad-hoc networks (fanet), remote healthcare monitoring (rhm), industrial automation, and ecosystem monitoring, the real-time processing or transmission of data generated by iot devices is highly critical. While these applications mostly rely on cloud services for the processing and storage of data, since cloud services fail to provide the required level of performance in such applications, the concept of fog computing was introduced [6], [7].

fog computing is an emerging paradigm that extends computational, communication, and storage resources closer to the network edge [8], [9]. Compared to cloud, fog computing can respond to latency-sensitive service requests from end-users with restricted energy resources and relatively low traffic demands [10]. Specifically, the integration of iot and fog computing in healthcare can enable remote patient monitoring, real-time analytics, and decision support systems. This integration enhances patient care and reduces the need for frequent hospital visits.

Iot has the potential to enhance healthcare outcomes by improving the analysis of patient data, reducing costs, and increasing the overall quality of care within healthcare systems. These devices can collect and transmit data in real-time, enabling healthcare providers to monitor patients remotely and make more informed decisions regarding their health status. However, the adoption of iot in healthcare raises concerns regarding data privacy and security. Untrusted environments, where parties engage without mutual trust, can be vulnerable to cyberattacks and data breaches. In healthcare, this vulnerability could have serious implications, as sensitive patient information may be exposed to risks such as data leakage, extortion, or misuse. To address these concerns, healthcare organizations must implement robust security measures to protect patient data. These measures include data encryption, access control, and regular system testing to identify and mitigate vulnerabilities. Beyond security, untrusted environments in healthcare require a high level of trust among patients, healthcare providers, and iot device manufacturers. Patients need assurance that their data is being collected and used appropriately, while healthcare providers must trust the accuracy and reliability of the data they receive. Iot device manufacturers, in turn, must maintain transparency about how patient data is collected and utilized.

a potential solution to address these challenges is the use of blockchain technology. Blockchain is a distributed ledger that

E. Dastani, M. A. Zarkesh, M. Davoodzadeh, B. Safaei, and A. Movaghar are with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. e-mail:{ehsan.dastani98, mahdi.akbarizarkesh, mahdi.davoodzadeh81, bardiasafaei, movaghar}@sharif.edu

Manuscript received x xx, xxxx; revised x xx, xxxx.

allows parties to exchange information in an untrusted environment without the need for intermediaries, ensuring the integrity of their data [11]. In healthcare, blockchain can be leveraged to create a secure and transparent system for sharing patient data. Patients can control who has access to their information, while healthcare providers can trust that the data they receive is accurate and up-to-date. Accordingly, to achieve the outlined objectives—enhancing system security and authentication while integrating iot data with blockchain—we have recently proposed a novel framework called edgelinker to be used in healthcare fog applications [12]. within this framework, iot devices periodically transmit patient data, e.g., heart rate, through the network. Concurrently, medical professionals require access to a patient's historical health data to make well-informed clinical decisions.

To facilitate this, the framework employs an access-controlled smart contract that securely stores patients' medical data in an array, ensuring access is restricted to authorized personnel. Initially, a smart contract is deployed on the blockchain to serve as the data repository. Patients transmit their health data to the blockchain via a secure communication channel directed to miners. Upon verification of the data and confirmation of the user's access rights, the information is recorded within the smart contract. The framework also enables patients to dynamically manage access to their data, allowing them to grant or revoke permissions to specific individuals or organizations at any time. When authorized individuals request access to the stored data, they initiate a transaction through the secure communication channel to the miners. Provided they hold the requisite access rights, they are permitted to retrieve the data. This approach ensures robust data security, privacy, and controlled access in real-time healthcare applications.

Given the intrinsic characteristics of blockchain technology, optimizing read and write times is crucial for improving the scalability of the proposed system. Real-world experiments reveal that the edgelinker framework exhibits exceptional efficiency across multiple dimensions. For instance, it achieves a 35% reduction in read times compared to state-of-the-art solutions, while maintaining negligible differences in write times on the blockchain. Moreover, edgelinker demonstrates superior scalability, evidenced by its linearly enhanced throughput as the number of fog nodes increases. This scalability is attributed to the integration of a proof-of-authority (poa) consensus mechanism on the blockchain, which facilitates secure access control through smart contracts [13]. The framework further incorporates advanced cryptographic algorithms to ensure secure communication between iot edge devices and the fog layer, particularly in healthcare applications.

Edgelinker leverages docker swarm technology, enabling efficient load balancing and orchestration of fog nodes. The experimental analysis highlights a minimal 1.32ms overhead for message transmission within the secure communication channel—merely 0.2ms greater than that of a non-secure channel. This marginal increase is negligible relative to the overall processing time, affirming the feasibility of using secure channels to bolster the framework's security without compromising scalability. Additionally, the framework achieves reduced cpu and ram utilization compared to existing works, further solidifying its position as a resource-efficient solution for healthcare applications. This combination of efficiency, security, and scalability underscores edgelinker's potential as a transformative tool in the domain of iot-based healthcare systems. In this paper, while we give a more comprehensive description of the proposed edgelinker framework compared to our initial study, we provide a more scrutinized security analysis of this platform, lacking in our foundational research. This security analysis includes but is not limited to sybil attacks, consensus manipulation, replay attacks, 51% attacks, traffic analysis, message spoofing, unauthorized access, distributed denial of service (ddos), and transaction malleability. Furthermore, by providing more detailed performance evaluations, we show-case its admirable functionality within real-world healthcare fog applications.

the structure of this paper is organized as follows: section ii presents a thorough review and critical evaluation of major existing studies from multiple perspectives. Section iii delves into a detailed explanation of the architecture and components of the proposed framework. The outcomes and performance analysis of the framework are discussed in section iv. In section v, the future directions are outlined to address potential advancements. Lastly, the paper is concluded with outlining the key findings and implications in section vi.

II. Literature Review

In the healthcare sector, safeguarding data security, managing transactions, and ensuring data integrity are vital for the effective operation of data-driven systems. Blockchain technology presents a viable solution to these challenges by providing secure, transparent, and immutable data management. Bruneo et al. [14] introduced Stack4Things, a fog-centric framework offered as a Platform-as-a-Service (PaaS) designed for efficient IoT application deployment and operation. This framework allows users to manage IoT infrastructure, control nodes remotely, virtualize their functions, and establish network overlays. Liang et al. [15] focused on cloud services with their development of ProvChain, a blockchain-based architecture that collects, stores, and validates cloud data provenance. By embedding provenance data into blockchain transactions, their approach ensures data authenticity across three stages: collection, storage, and validation. In a comprehensive review, Gordon and Catalini [16] analyzed how blockchain could transform healthcare by shifting control of data from institutions to patients. Their study highlighted the potential of blockchain to reframe the industry by integrating diverse data sources, applications, and stakeholders. Sial et al. [17] argued for the benefits of blockchain and smart contracts in streamlining healthcare processes. They emphasized that securely storing healthcare data in a distributed ledger could prevent information tampering and reduce data loss. Addressing privacy concerns, Vora et al. [18] proposed a blockchain-based system to manage electronic health records. Their framework aims to meet the requirements of patients, healthcare providers, and other stakeholders by securing private information such as names and addresses. Zhang et al. [19] explored the initial

integration of blockchain and smart contracts in healthcare. Their research highlighted the potential applications and deployment challenges associated with blockchain, demonstrating its promise for improving healthcare management. A review of this prior research underscores the broad applicability of blockchain for enhancing data security, privacy, and integrity in healthcare. Blockchain's decentralization, transparency, and immutability address critical concerns, such as data breaches and unauthorized access. A comparative analysis of these studies is provided in Table I, summarizing their methodologies and outcomes.

III. Proposed Methodology

A. Architecture

To optimize computational efficiency at the network edge and minimize service latency, a three-tier architecture is essential. Figure 1 depicts the proposed framework's architecture, consisting of a cloud data center, fog layer, and IoT devices in the users layer. The cloud data center serves as a high-capacity computational resource. It collects data from the fog layer when required, performs intensive computations, and returns the results to the fog layer for distribution and utilization. The fog layer comprises devices that include a blockchain ledger, smart contracts, and a global monitoring system. Fog nodes unction as miners, tasked with collecting, processing, and validating data blocks and messages. This creates a distributed computing ecosystem. Additionally, fog nodes verify newly added blocks to prevent fraudulent transactions, ensuring the blockchain's security and integrity. The user layer consists of IoT devices equipped with sensors, actuators, and various data acquisition technologies. These devices gather vital patient information and transmit it to the fog layer, positioned at the network edge, for processing and analysis.

B. Fog Layer

The fog layer incorporates a blockchain ledger, smart contracts, and a global monitoring system. Each fog node operates as a miner, responsible for validating incoming messages. Once a message is validated, the fog node generates a new block containing the processed data. This newly created block is subsequently propagated to other fog nodes within the network for further validation, ensuring data integrity and consistency across the system.

1) **Blockchain:** Given the critical importance and sensitivity of medical data, as well as the need to maintain data integrity and stability, we propose using blockchain as a distributed storage solution for such data. Additionally, to automate existing processes and enable the personalized use of the proposed framework, our blockchain must support user-specific applications. Therefore, the proposed framework must incorporate support for smart contracts. Finally, the system should demonstrate high speed and scalability. To meet these requirements, we propose adopting the Proof of Authority (PoA) consensus algorithm [13] in our blockchain. This algorithm is an effective method that permits only authorized nodes—referred to as validator nodes—to participate in the block-mining process.

To prevent unauthorized access and visibility of the blockchain's state, a semi-private network is employed for implementing the proposed framework. In this setup, only miners can access the network's history for validation purposes, while the history remains hidden from others. It is important to note that this does not disrupt the functionality of IoT devices. These devices can connect to the network whenever needed, invoke the required smart contracts, and then disconnect from the network afterward. Additionally, events related to smart contracts, the creation of new blocks, and updates can be monitored externally beyond the fog layer.

The structure of the proposed blocks includes a block generation timestamp, user messages containing added hashes, and a hash of the previous block. When miners aim to generate a new block, they aggregate the user messages,

Algorithm 1: Permission-Based Access Control

```
Input: Transaction T
Define: _permission: a mapping from Permission to List[Address],
PERMITTER_PERMISSION = 0x00;
Function Initialize():
    _permission[PERMITTER_PERMISSION].add(T.sender);
end
Function HasPermission(permission, address):
    return _permission[permission].has(address);
end
Function GrantPermission(permission, address):
    if HasPermission(PERMITTER_PERMISSION, T.sender) then
        _permission[permission].add(address);
    end
end
Function RevokePermission(permission, address):
    if HasPermission(PERMITTER_PERMISSION, T.sender) then
        _permission[permission].remove(address);
    end
end
```

the hash of the previous block into the header of the new block, and generate a hash of the entire block. This hash is then signed with their private key. Finally, the new block is broadcast across the network for validation. Upon approval, the block is added to the ledger of the fog nodes. Maintaining and operating the blockchain by miners requires an incentive system to encourage individuals to allocate their computational resources for execution and processing. To address this, a local cryptocurrency is employed as a dual-purpose solution: ensuring fair access to computational resources for IoT devices and incentivizing miners for transaction validation and net- work support. This approach

promotes network stability while balancing resource allocation and miner rewards.

2) **Smart Contracts:** Smart contracts are integral to automating processes within the framework. However, to ensure data confidentiality, all smart contracts designed for data storage must incorporate the access control mechanism, which will be detailed in subsequent sections of this discussion.

Algorithm 1 demonstrates access control in smart contracts using access permissions. This method restricts access to specific data and functionalities to authorized individuals only, thereby enhancing data storage security. Employing an access control system provides an ideal solution for creating a secure and trustworthy environment for data management.

To automate processes in traditional systems, the use of "events" is enabled. Events can be tracked by programs running on cloud servers and other network nodes, facilitating indirect interactions. By utilizing events, it is possible to implement pipelines spanning from data collection to processing and delivery. These pipelines offer a secure and reliable approach for gathering data from sensors and IoT devices such as heart rate sensors, medical devices, and other types of data sensors, followed by data processing.

Additionally, with the help of oracles, it becomes feasible to receive data from external environments and make decisions based on it.

3) **Global Monitoring System:** Smart contracts operate by receiving a message sent to their address, which triggers their execution if predefined conditions are satisfied. These contracts, however, are limited to accessing only the data explicitly sent to them or predefined within their logic. This limitation poses challenges in scenarios requiring the detection of fraudulent patterns or security breaches, where comprehensive analysis of all incoming and outgoing transactions is essential for generating alerts.

Algorithm 2: Message Transmission Through Secure Channel

Input:
- m: Message with Nonce to be sent
- SK_s: Sender's private key
- PK_r: Receiver's public key

Output:
- c: Encrypted and signed message

Compute shared key $K \leftarrow DH(SK_s, PK_r)$ using authenticated Diffie-Hellman;
Compute hash $H(m)$ of the message;
Sign hash: $sig \leftarrow Sign(H(m), SK_s)$;
Concatenate message and signature: $m' \leftarrow m \parallel sig$;
Encrypt concatenation: $c \leftarrow Encrypt(m', K)$;
Send encrypted message c to receiver;

To address this issue, the proposed framework incorporates a robust monitoring system. This system is deployed across all fog nodes, enabling real-time oversight of network activity. In its current iteration, the monitoring system focuses on generating alerts when an invalid block is detected within the network. Future enhancements aim to introduce more sophisticated and nuanced conditions for identifying and responding to suspicious or unauthorized behaviors, thereby strengthening the overall security of the framework.

C. Secure Communication Layer

This research proposes a secure communication channel designed to ensure device identification, authentication, authorization, and efficient message transmission. The solution is intended to be cost-effective and minimize the volume of transmitted messages. Recognizing that asymmetric encryption increases message size while symmetric encryption benefits from faster hardware implementation, a hybrid approach combining these two methods is adopted. This approach balances security, efficiency, and performance. To validate message transmission and ensure non-repudiation, all entities within the system are equipped with a pair of public and private keys. The public key, used for identification, is publicly accessible, while the private key remains confidential and is used for decrypting messages. Each transmitted message includes a timestamp, a nonce, and an identification field. The timestamp ensures temporal validity, while the nonce, a unique value associated with each public key and incremented with every transaction, prevents replay attacks and maintains transaction order. The identification field, represented by the sender's public key, provides a clear identifier for each entity. This comprehensive design strengthens the security, authenticity, and reliability of the communication channel.

Before sending a message, the sender executes the steps outlined in Algorithm 2, which include the following:

- Calculating the hash of the message.
- Signing the computed hash using their private key to ensure authentication and appending it to the end of the message.
- Encrypting the entire message using the Diffie-Hellman key exchange to prevent eavesdropping on the message content.

Once these steps are completed, the sender transmits the message to the receiver. Figure 2 illustrates an example of a transmitted message.

On the receiver's side, the process detailed in Algorithm 3 ensures secure message handling and sender authentication. Initially, the message is decrypted using the Diffie-Hellman key. Subsequently, the hash value of the decrypted message is computed. In parallel, the hash received alongside the message is decrypted using the sender's public key. If the computed hash matches the decrypted hash, the message integrity is confirmed, and the sender's identity is authenticated. Following successful decryption and authentication, the received nonce value is verified against the last stored nonce value in the blockchain. This step ensures the prevention of duplicate messages and preserves the correct sequence of transactions, thereby maintaining the reliability and security of the communication system.

To accommodate legacy devices, the proposed framework allows these devices to leverage the computational resources of the fog layer as an intermediary or proxy for transmitting

their requests to the network. This approach enhances the security of data messages generated by legacy devices while enabling continued use of their existing capabilities. By acting as a secure interface, the fog layer ensures compatibility and extends the functionality of older devices within the modern network architecture.

The EdgeLinker framework represents significant progress in overcoming key challenges related to security, privacy, and scalability. By employing a three-layer architecture and

Algorithm 3: Message Reception Through Secure Channel

Input:
- c: Encrypted and signed message
- K: Shared DH key
- PK_s: Sender's public key

Output:
- m: Original message if authentication succeeds
- Error otherwise

Decrypt with shared key: $c' \leftarrow Decrypt(c, K)$;
Parse c' as $m \parallel sig$;
Compute $h \leftarrow Hash(m)$;
Compute sender hash: $h' \leftarrow Decrypt(sig, PK_s)$;
if $h' = h$ **then**
  return m;
else
  return Error;
end

fog computing resources, the framework substantially reduces service latency. Additionally, the use of blockchain ensures immutability and reliable record-keeping of information. The implementation of the POA algorithm within the blockchain layer enhances security, improves energy efficiency, and ensures greater scalability. Integrating smart contracts with access control shifts data ownership from organizations to individuals. Moreover, the use of smart contracts and pipeline capabilities modernizes traditional methods and eliminates the need for a centralized organization to manage and validate operations. The adoption of a secure communication channel guarantees data integrity, privacy, security, and authentication features. Most existing studies lack the implementation of algorithms and designed models in real world environments. Therefore, a comprehensive analysis of the proposed framework's security is presented in the following sections. Subsequently, the operational cost of the proposed solution will be calculated and analyzed.

IV. Evaluation and Analysis

In this section, to evaluate the proposed work, we first compare it with the baseline approach in the healthcare ecosystem, which involves using a centralized database for storing and protecting existing data. Then, we compare our work with one of the prominent existing solutions aimed at integrating cloud and fog infrastructures, namely Fogbus [28], which closely resembles our proposed approach.

To demonstrate the applicability of the proposed system in healthcare and its potential for patient health monitoring, we utilized the Galaxy Watch 4 Classic as an IoT device. The smartwatch sends the patient's average heart rate to the network every minute. In this scenario, a doctor intends to retrieve the patient's heart rate history from the past hour to make an informed decision. To facilitate this, a smart contract with access control capabilities was employed, storing the user's heart rate data in an array and allowing access only to authorized individuals. The smart contract used is presented in Algorithm 4. All communications in this experimental scenario were conducted via Wi-Fi, with a tablet serving as the doctor's device for viewing the heart rate data.

Figure 3 illustrates the sequence of requests associated with

Algorithm 4: Storing Patient Heart Rate Data with Access Control

Input: Extend PBAC contract;
Define: *__data* as List[uint32], WRITER as $0x01$, READER as $0x02$;
Procedure *Initialize()*:
  Call superclass Initialize;
  grantPermission(WRITER, T.sender);
  grantPermission(READER, T.sender);
end
Procedure *AddEntry(heartRate)*:
  if hasPermission(WRITER, T.sender) **then**
    __data.add(heartRate);
    Trigger Event("addEntry");
  end
end
Procedure *GrantWriterPermission(T, address)*:
  grantPermission(WRITER, address);
end
Procedure *RevokeWriterPermission(T, address)*:
  revokePermission(WRITER, address);
end
Procedure *GrantReaderPermission(T, address)*:
  grantPermission(READER, address);
end
Procedure *RevokeReaderPermission(T, address)*:
  revokePermission(READER, address);
end
Function *ReadDataT*:
  if hasPermission(READER, T.sender) **then**
    return *__data*;
  end
end

the proposed framework, starting from the request to create a smart contract for data storage, followed by granting access to the doctor for data retrieval, and finally, the doctor's request to access the data.

Initially, a smart contract is created for data storage. The user then sends their data to miners through a secure communication channel. If the information is verified and the user has the appropriate permissions, the data is stored in the smart contract. At any time, the user can request to grant access to their data to an individual or organization. Additionally, the user can revoke access from individuals at their discretion. When someone requests to read the data, they send a transaction to the miners via a secure communication channel. If they have the required permissions, they are granted access to the data.

We now aim to provide a summary of the laboratory setup used to evaluate the performance of the proposed system. This setup includes information about the test network, frontend, back-end, server, tools, hosts, programming language, and

integrated development environment (IDE), which are briefly presented in Table II. In this framework, the front-end was developed using React Native, a Web3-compatible framework that facilitates direct communication with the blockchain. The back-end implementation utilized Node.js version 18.16, while the smart contracts were created using Solidity within the Remix IDE + Thor environment. To ensure user-friendly interaction and compatibility with multiple networks, the Sync 2 wallet was selected for its intuitive interface and extensive support. Furthermore, the framework incorporates a customized version of the IBFT 2.0 consensus algorithm, chosen for its robustness and ability to withstand node failures, enhancing the system's reliability and resilience.

The customised IBFT-2.0 implementation in our framework relies on PoA because it meets all operational constraints of a healthcare fog-to-cloud network that must deliver real-time vital-sign data while remaining audit-ready and energy-efficient:

1) Deterministic, sub-second finality — a single round of signatures by a small validator set lets clinicians react to heart-rate alarms within seconds.
2) Minimal resource and energy footprint — validators only sign blocks; no mining, staking, or heavy view-change traffic, keeping CPU/RAM use lowest among baselines and raising device energy by just 0.7 %.
3) Permissioned trust & legal accountability — validators are auditable hospital entities, blocking Sybil attacks at admission and satisfying health-data regulations.
4) Linear scalability for 5–20 fog nodes — throughput scales nearly linearly, whereas BFT protocols face quadratic message overhead and PoW/PoS add confirmation delays.
5) Security fit for healthcare — identity-bound validators deter 51 % cartels, and nonces/timestamps stop repla attacks while IBFT-2.0 tolerates node failures.

In short, PoA delivers the low-latency, low-power, and regulator-friendly consensus that our real-time healthcare deployment demands. Other schemes—such as delegated PoS, asynchronous BFT, or DAG-based ledgers—could address specific scenarios (e.g., larger validator pools or cross-chain interoperability) and merit future exploration, but a comprehensive evaluation of every alternative was beyond the scope of this work.

To evaluate the performance, experiments were conducted using hardware specifications that included five cloud servers in a data center, each with 16 CPU cores and 64 GB of RAM. Docker Swarm was utilized for load management, with each node allocated 2 CPU cores and 8 GB of RAM. Additionally, in the traditional method, the number of databases was fixed at one. The performance of the proposed solution was compared with a centralized cloud database, a distributed fog-based database, and FogBus. The results indicate the superiority of the proposed solution in terms of latency and scalability. To ensure accuracy, the statistics represent the average of five executions.

A. Security Analysis

1) Security Services: Security services are the fundamental protection goals—confidentiality, integrity, availability authentication, and non-repudiation—that any trustworthy healthcare network must provide.

In a healthcare environment, these services translate into concrete assurances for clinicians, patients, and inspectors. Confidentiality keeps diagnoses private, integrity guarantees traceable clinical outcomes, availability preserves life-critical connectivity, authentication binds devices and staff to verifiable identities, and non-repudiation delivers a legally defensible audit trail.

Confidentiality. All user data are protected by role-based access control policies embedded in smart contracts, allowing only authorised entities to read sensitive information [29]. A cardiologist, for example, must satisfy both her on-chain role and an off-chain consent token before a patient's electro-cardiogram is decrypted—removing reliance on a central IT operator and recording the access immutably.

Integrity. Cryptographic hashes at both message and block level ensure that any modification is detectable; the approach follows the dynamic integrity-verification scheme for smart homes in [31]. Because every sensor reading is committed with a Merkle proof, even subtle bit-flips introduced by malware are detected immediately, preventing clinicians acting on tampered clinical evidence.

Availability (including DDoS). The permissioned blockchain's redundancy keeps the network functional even if individual nodes fail or are overwhelmed. Empirical studies show that validator whitelisting plus edge-level rate limiting improves IoT resilience against volumetric DDoS [32], [45]. In practice, if a subset of gateways are saturated, the remaining validators automatically rebalance traffic so that medication-dispensing devices continue to operate within safe latency budgets.

Authentication. Every device owns a public–private key pair; signatures are verified in line with NIST guidance for blockchain access-control systems [33]. This enables zero-touch onboarding: once a glucometer's hardware security module proves control of its private key, it is recognised across the hospital network without further manual configuration.

Non-repudiation. Signatures on both individual messages and whole blocks bind every transaction to its origin, meeting audit requirements in [33]. If a dosage command is later questioned, forensic investigators can unambiguously attribute the command to the prescribing physician and verify that it was unaltered en route to the infusion pump.

2) Attacks and Countermeasures: The following catalogue assesses classical attacks and their mitigations; advanced threats such as quantum-computing assaults lie outside the present scope and are flagged for future study. A layered defence-in-depth posture counters each attack vector at multiple points—on-device, at the network edge, and on-chain. Where possible, countermeasures are preventive (e.g., cryptographic enforcement); otherwise they are detective with automated containment triggers. The table below expands on how the controls just enumerated map to concrete threats.

Eavesdropping. All communications use end-to-end

encryption; intercepted data are unreadable without the decryption keys [30]. Forward secrecy additionally ensures that the later compromise of a long-term device key does not expose archived telemetry, closing a common gap in legacy HL7 tunnels.

Packet Dropping. Edge/fog monitoring can raise alerts when expected traffic is absent, as proposed for blockchain-based IoT sensor networks [45]. Dropped-packet heuristics incorporate adaptive thresholds so that transient Wi-Fi congestion does not overwhelm clinicians with false alarms, yet sustained suppression of life-critical data streams is escalated within seconds.

Identity Spoofing (Impersonation). Because every node signs its messages, spoofing requires stealing a private key—an attack the IEEE-Access evaluation in [43], [47] found practically infeasible when hardware key storage is used. Tamper-evident secure elements erase keys on physical intrusion, while remote attestation lets validators refuse connections from devices whose firmware is out of date or unsigned.

Insertion Attack. Fake blocks are rejected during Po validation; cloning-attack analyses confirm that signature and round-timing checks foil such attempts [35], [37]. Combined with a rotating leader schedule, this removes any single validator as a predictable choke point, thereby neutralising

targeted insertion attempts.

Linkage Attack. Periodic key rotation makes transaction linkage harder, but studies of blockchain deanonymization show that metadata can still leak identity [42], [46]. To further muddy adversarial inference, padded payload sizes and timing obfuscation can be enabled for high-sensitivity workflows such as mental-health counselling.

Attacks on Consensus (Sybil, 51% and Manipulation).

- **Sybil.** Creating many fake nodes undermines redundancy [34]. PoA restricts validator IDs, and a randomised- authenticator design further mitigates cloning [35]. Ad- mission contracts also require a stake deposit, making large-scale Sybil creation economically unattractive.
- **51% Attack.** If colluding validators exceed half the authority set, they can rewrite history; the risk model of [40] applies even in PoA. Regular audits and distributed governance reduce this threat. In addition, cross-hospital notarisation checkpoints lock block histories, forcing would-be attackers to break external signatures as well—a substantially higher bar.
- **Order/Censorship Manipulation.** Unfair ordering attacks on PoA have been demonstrated [36], [37]; monitoring plus penalty rules are therefore recommended. Anomaly detectors watch for consistently delayed prescriptions or lab results, automatically slashing the validator that introduces a statistically significant bias.

Replay Attacks. Including nonces and timestamps prevents duplicates; cross-shard replay countermeasures are formalized in [38]. Nonce windows are chosen conservatively so that devices with intermittent connectivity, such as ambulance rigs, remain compatible without sacrificing security.

Traffic Analysis. Encrypted metadata can still leak patterns; padding or jitter can mitigate the risks highlighted in [42], [46]. The network orchestrator dynamically toggles padding profiles based on observed load, balancing privacy with bandwidth constraints in rural clinics.

Message Spoofing. Digital signatures and on-chain certificates defeat spoofed messages [43], [47]. Validators additionally compare firmware version attestations to catch

supply-chain compromises in which a legitimate key is embedded in rogue code.

Unauthorized Access. Validator admission controls, hardware key protection, and continuous auditing follow best practices in [29], [33]. Role-revocation logic propagates within a single block interval, preventing terminated contractors from accessing medical devices even if their local credentials are cached.

Transaction Malleability. Hash-binding and nonce inclusion render malleability exploits ineffective [39]; additional safe-guards prevent double-spending variants [44]. A side benefit is compliance with upcoming EU eHealth directives that mandate tamper-proof audit trails for every prescription change.

EdgeLinker's permissioned PoA ledger, layered encryption, and robust key management collectively address confidentiality, integrity, availability, and a broad spectrum of attacks. Residual gaps such as packet-dropping can be closed with intrusion-detection and alert modules in future work.

B. Performance Analysis

The performance of EdgeLinker was benchmarked against a centralized cloud database, a distributed fog database and FogBus [28]. The results demonstrated that EdgeLinker outperformed these solutions in both latency and scalability. To ensure the accuracy and reproducibility of the findings, all reported statistics represent the averages of five independent runs.

The evaluation of EdgeLinker's performance focused on three key metrics: processing delay, processing time, and throughput (TPS). Processing delay refers to the time required to record, store, or retrieve transaction data on the blockchain or storage system, measured from the moment the data is received by the node until the processing is complete. Processing time is the duration between the user's initial request and the receipt of the corresponding response. Throughput (TPS) quantifies the number of transactions validated, executed, inalized, and confirmed per second after consensus is achieved in the network. These metrics are critical for assessing the efficiency of the proposed framework compared to alternative approaches.

Additionally, the study plans to analyze EdgeLinker's energy consumption in comparison to other methods, further high-lighting its overall effectiveness and sustainability.

The experimental results depicted in Figure 4 demonstrate

that EdgeLinker significantly outperforms the Edge database in terms of read processing time. Specifically, with 5 nodes, the performance is on average 23.8% better, with 10 nodes 31% better, with 15 nodes 33.2% better, and with 20 nodes 35.1% better. Similarly, as shown in Figure 6, throughput improves with an increasing number of nodes across all four examined methods. However, in alternative methods—fog database, cloud database, and FogBus—the presence of bottlenecks slows the rate of throughput improvement. Once these methods reach the maximum capacity of the nodes, the rate of improvement diminishes.

As illustrated in Figure 5, the fog database outperforms EdgeLinker by an average of 4.7% with a single node. However, as the number of nodes increases to five, this performance gap narrows to 2.3%, with EdgeLinker achieving comparable performance to the fog database. When the number of nodes exceeds ten, the performance of EdgeLinker declines due to the additional data exchange required to establish a distributed chain and finalize blocks. In the FogBus solution, the Proof-of-Work (PoW) consensus algorithm stabilizes the time needed to solve block creation challenges as the number of tasks increases. Consequently, with 15 nodes and over 500 tasks, FogBus slightly surpasses EdgeLinker, but the performance difference is minimal, amounting to less than 0.2%. Figure 7 highlights that the throughput of EdgeLinker improves with the addition of nodes, peaking at five nodes.

Beyond this point, the throughput declines due to increased coordination overhead among nodes. Conversely, FogBus demonstrates superior scalability, benefiting from more effective load distribution among incoming nodes. As a result, its capacity and throughput continue to increase at a higher rate compared to other solutions as the number of nodes expands. These results underscore the scalability trade-offs and operational efficiency of EdgeLinker in different deployment scenarios.

As shown in Figure 8, the fog and cloud database solutions exhibit higher RAM consumption compared to other approaches. This is primarily due to the inherent use of memory for caching data and managing background tasks required for data maintenance. Similarly, the FogBus solution demands more memory than EdgeLinker, as it involves additional computations to solve complex puzzles. Among all the solutions evaluated, EdgeLinker demonstrates the lowest RAM consumption, utilizing memory efficiently to cache newer blocks and maintain the overall state of the blockchain, thereby enhancing processing speed.

Furthermore, Figure 9 illustrates that the FogBus solution consumes more CPU resources than EdgeLinker, particularly when performing the intensive computations necessary to create new blocks.

Regarding the secure communication channel, the use of encryption and hashing mechanisms leads to slightly increased processing and transmission times, especially for larger data packets. As depicted in Figure 10, the time increase is approximately 0.2ms, which is negligible when compared to the total processing time, further supporting the feasibility of using secure channels without significantly impacting system performance.

Energy consumption for devices encompasses the processor, network, sensors, and display components. The implementation of a secure channel increases the size of the transmitted packets and requires additional computations, leading to an average energy consumption increase of 0.7% for data transmission compared to the baseline approach. This was measured by initially charging a device to 80%, sending 60 messages at one-minute intervals, and then recording the remaining charge. The results are illustrated in Figure 11.

If EdgeLinker is deployed on an existing public blockchain network such as VeChain, each transaction incurs a gas cost, which depends on the complexity of the smart contract operations and the level of network congestion. Gas fees on on VeChain are paid in VTHO tokens, where 1000 gas equals 1 VTHO. To estimate deployment costs, the framework was tested in a local environment, and the calculated gas fees were scaled based on a VTHO price of $0.001. A detailed breakdown of the implementation and interaction costs associated with the smart contract is provided in Table III. This analysis highlights both the computational efficiency and financial feasibility of deploying EdgeLinker in real-world environments.

V. Limitations and Future Research Studies

Despite the encouraging results obtained with the EdgeLinker healthcare fog prototype, several constraints remain. Its scalability has only been verified under moderate workloads, so behaviour at hospital scale—where transaction rates, patient volumes and validator counts surge simultaneously—remains untested. The blockchain layer also introduces measurable latency and extra energy consumption, which could impede sub-second access to vital data during emergencies, especially if each sensor reading is written to its own block. Interoperability with standard electronic-health-record formats such as HL7 or FHIR is still ad-hoc, and the system offers no on-chain mechanism for resolving disputes over data ownership or access rights once records are stored. The security analysis is confined to classical threats, leaving quantum-enabled cryptanalysis and large-scale DDoS campaigns for future consideration. Finally, although Proof-of-Authority minimises computational effort, maintaining a semi-private validator set over time and fully documenting performance-evaluation procedures both require deeper investigation.

Future work will therefore focus on several complementary directions. First, outgoing data streams will be classified as sensitive or non-sensitive so that full cryptographic safeguards are applied only where necessary, reducing latency and energy overheads. Second, comprehensive mathematical models and large-scale simulations will be developed to stress-test scalability across varying transaction loads and network diameters. Third, hybrid emergency-aware architectures—combining priority off-chain channels with periodic blockchain reconcilia tion—will be explored to guarantee real-time access during critical events. Fourth, energy optimisation techniques such as

batching or aggregating biomedical samples before block creation, together with tuned block-interval parameters, will be evaluated. Fifth, interoperability layers for HL7/FHIR will be implemented alongside arbitration smart contracts and private-key-splitting schemes that prevent any single entity from unilaterally controlling clinical records. Sixth, resilience will be enhanced by integrating DDoS-specific defences and investigating quantum-resistant cryptographic primitives that can be adopted without redesigning the ledger. Finally, the benchmarking suite will be expanded with transparent workload descriptions, reproducible scripts and detailed reporting of latency, throughput and energy metrics, paving the way for generalisation of the framework to other safety-critical IoT scenarios such as smart-city infrastructure, oil-and-gas monitoring and industrial automation.

VI. Conclusion

In this study, we proposed EdgeLinker; a comprehensive IoT framework that uses Proof-of-Authority consensus, integrates smart contracts on the blockchain for access control, and employs advanced cryptographic algorithms for secure data communication between edge and fog devices in healthcare applications. In addition to a detailed performance evaluation of this novel framework, we give an in-depth security analysis from various perspectives. Given that healthcare data is primarily analyzed by researchers and doctors with fewer new data entries, the proposed framework proves to be a practical solution due to its excellent data reading performance. Furthermore, its lower cost and system load, compatibility with legacy devices, and support for user-customized applications enhance its practicality. This framework not only improves existing processes but also facilitates the operational deployment of blockchain technology in the healthcare sector. Its ability to integrate seamlessly with current systems ensures minimal disruption, making it an attractive option for healthcare providers aiming to enhance data security and efficiency.

References

[1] V Hassija, V Chamola, V Saxena, et al. ”A survey on IoT security: application areas, security threats, and solution architectures,” IEEE Access, Vol. 7, pp. 82721-82743, 2019, DOI: https://doi.org/10.1109/ACCESS.2019.2924045

[2] P Sadeghi, B Safaei, K Talaei, et al. ”Towards a reliable modulation and encoding scheme for Internet of Things communications,” in Proceedings of the 13th IEEE International Conference on Application of Information and Communication Technologies (AICT), pp. 1-6, 2019, DOI: https://doi.org/10.1109/AICT47866.2019.8981775

[3] B Safaei, H Taghizade, AMH Monazzah, et al. ”Introduction and evaluation of attachability for mobile iot routing protocols with markov chain analysis,” IEEE Transactions on Network and Service Management, Vol. 19, no. 3, pp. 3220–3238, 2022, DOI: https://doi.org/10.1109/TNSM.2022.3176365

[4] B Safaei, AMH Monazzah, MB Bafroei, et al. ”Reliability side-effects in Internet of Things application layer protocols.” in Proceedings of the International Conference on System Reliability and Safety, IEEE, 2017, DOI: https://doi.org/10.1109/ICSRS.2017.8272822

[5] M Shirbeigi, B Safaei, AAM Salehi, et al. ”A cluster-based and drop-aware extension of RPL to provide reliability in IoT applications,” in Proceedings of the IEEE International Systems Conference (SysCon), pp. 1-7, 2021, DOI: https://doi.org/10.1109/SysCon48628.2021.9447112

[6] SA Chamazcoti, B Safaei, SG Miremadi. ”Can erasure codes damage reliability in SSD-based storage systems,” IEEE Transactions on Emerging Topics in Computing, Vol. 7, no. 3, pp. 435-446, 2017, DOI: https://doi.org/10.1109/TETC.2017.2693424

[7] B Safaei, SG Miremadi, SA Chamazcoti. ”Implicit effect of decoding time on fault tolerance in erasure coded cloud storage systems,” in Proceedings of the Inter- national Computer Science and Engineering Conference (ICSEC), pp. 1-6, 2016, DOI: https://doi.org/10.1109/ICSEC.2016.7859937

[8] A Motamedhashemi, B Safaei, AMH Monazzah, et al. ”DATA: Throughput and Deadline-Aware Genetic Approach for Task Scheduling in Fog Networks,” IEEE Em- bedded Systems Letters, 2023, DOI: https://doi.org/10.1109/LES.2023.3348499

[9] A Motamedhashemi, B Safaei, AMH Monazzah, et al. ”FUSION: A Fuzzy-Based Multi-Objective Task Management for Fog Networks,” IEEE Access, Vol. 12, pp. 152886-152907, 2024, DOI: https://doi.org/10.1109/ACCESS.2024.3480360

[10] M Mukherjee, L Shu, D Wang, et al. “Survey of fog computing: Fundamental, network applications, and research challenges,” IEEE Communications Surveys & Tutorials, Vol. 20, No. 3, pp. 1826–1857, 2018, DOI: https://doi.org/10.1109/COMST.2018.2814571.

[11] I Bashir, “Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more,” Packt Publishing Ltd, https://www.packtpub.com/en-eg/product/mastering-blockchain-9781839213199

[12] M A Zarkesh, F Dastani, B Safaei, et al. “EdgeLinker: Practical Blockchain-based Framework for Healthcare Fog Applications to Enhance Security in Edge-IoT Data Communications,” in Proceedings of the 5th CPSSI International Symposium on Cyber-Physical Systems (Applications and Theory) (CPSAT), IEEE, pp. 1–8, 2024, DOI: https://doi.org/10.1109/CPSAT64082.2024.10745419.

[13] https://apla.readthedocs.io/en/latest/concepts/consensus.html

[14] D Bruneo, S Distefano, F Longo, et al. “Stack4Things as a fog computing platform for Smart City applications,” in Proceedings of the IEEE Conference on Computer Communications Workshops, pp. 848–853, 2016, DOI: https://doi.org/10.1109/INFCOMW.2016.7562195.

[15] X Liang, S Shetty, D Tosh, et al. “ProvChain: A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability,” in Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 468–477, 2017, DOI: https://doi.org/10.1109/CCGRID.2017.8.

[16] W J Gordon and C Catalini, “Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability,” Computational and Structural Biotechnology Journal, Vol. 16, pp. 224–230, 2018, DOI: https://doi.org/10.1016/j.csbj.2018.06.003.

[17] A A Siyal, A Z Junejo, M Zawish, et al. “Applications of blockchain technology in medicine and healthcare: Challenges and future perspectives,” Cryptography, Vol. 3, No. 1, 2019, DOI: https://doi.org/10.3390/cryptography3010003.

[18] J Vora, A Nayyar, S Tanwar, et al. “BHEEM: A blockchain-based framework for securing electronic health records,” in Proceedings of the IEEE Globecom Workshops, 2018, DOI: https://doi.org/10.1109/GLOCOMW.2018.8644088.

[19] P Zhang, J White, D C Schmidt, et al. “Design of blockchain-based apps using familiar software patterns to address interoperability challenges in healthcare,” in Proceedings of the Conference on Pattern Languages of Programs, 2017, DOI: https://doi.org/10.5555/3290281.3290304.

[20] N Chen, Y Chen, X Ye, et al. “Smart city surveillance in fog computing,” in Advances in Mobile Cloud Computing and Big Data in the 5G Era, pp. 203–226, 2017, DOI: https://doi.org/10.1007/978-3-319-45145-9_9.

[21] S Yi, Z Hao, Z Qin, et al. “Fog computing: Platform and applications,” in Proceedings of the IEEE Workshop on Hot Topics in Web Systems and Technologies, 2015, DOI: https://doi.org/10.1109/HotWeb.2015.22.

[22] B Shen, J Guo, and Y Yang, “MedChain: Efficient healthcare data sharing via blockchain,” Applied Sciences, Vol. 9, No. 6, 2019, DOI: https://doi.org/10.3390/app9061207.

[23] A Azaria, A Ekblaw, T Vieira, et al. “MedRec: Using blockchain for medical data access and permission management,” in Proceedings of the International Conference on Open and Big Data, 2016, DOI: https://doi.org/10.1109/OBD.2016.11.

[24] P Bhattacharya, S Tanwar, U Bodkhe, et al. “BinDaaS: Blockchain-based deep-learning as-a-service in healthcare 4.0 applications,” IEEE Transactions on Network Science and Engineering, Vol. 8, No. 2,

pp. 1242–1255, 2019, DOI: https://doi.org/10.1109/TNSE.2019.2961932.
[25] A Yazdinejad, G Srivastava, R M Parizi, et al. "Decentralized authentication of distributed patients in hospital networks using blockchain," IEEE Journal of Biomedical and Health Informatics, Vol. 24, No. 8, pp. 2146–2156, 2020, DOI: https://doi.org/10.1109/JBHI.2020.2969648.
[26] P Sharma, R Jindal, and M D Borah, "Blockchain-based cloud storage system with CP-ABE-based access control and revocation process," Journal of Supercomputing, Vol. 78, No. 6, pp. 7700–7728, 2022, DOI: https://doi.org/10.1007/s11227-021-04179-4.
[27] L Ouyang, Y Yuan, Y Cao, et al. "A novel framework of collaborative early warning for COVID-19 based on blockchain and smart contracts," Information Sciences, Vol. 570, pp. 124–143, 2021, DOI: https://doi.org/10.1016/j.ins.2021.04.021.
[28] S Tuli, R Mahmud, S Tuli, et al. "FogBus: A blockchain-based lightweight framework for edge and fog computing," Journal of Systems and Software, Vol. 154, pp. 22–36, 2019, DOI: https://doi.org/10.1016/j.jss.2019.04.050.
[29] S Namane and I Ben Dhaou, "Blockchain-Based Access Control Techniques for IoT Applications," Electronics, Vol. 11, No. 14, 2022, DOI: https://doi.org/10.3390/electronics11142225.
[30] H Kim, J Park, and K Kim, "A Secure End-to-End Communication Framework for Cooperative IoT Networks Using Blockchain," Scientific Reports, Vol. 15, 2025, DOI: https://doi.org/10.1038/s41598-025-96002-w.
[31] C Chen, L Wang, Y Long, et al. "A Blockchain-Based Dynamic and Traceable Data Integrity Verification Scheme for Smart Homes," Journal of Systems Architecture, Vol. 130, 2022, DOI: https://doi.org/10.1016/j.sysarc.2022.102677.
[32] Q Lin, P Sharma, and C Y Park, "Blockchain-Based Solutions to Mitigate DDoS Attacks in IoT," Sensors, Vol. 22, 2022, DOI: https://doi.org/10.3390/s22031094.
[33] National Institute of Standards and Technology, "Blockchain for Access Control Systems," NIST IR 8403, 2022, DOI: https://doi.org/10.6028/NIST.IR.8403.
[34] J R Douceur, "The Sybil Attack," in Proc. IPTPS, 2002, DOI: https://doi.org/10.1007/3-540-45748-8_24.
[35] R Wang, Y Li, and X Chen, "An Efficient Proof-of-Authority Consensus Scheme Against Cloning Attacks," Computer Communications, Vol. 205, 2024, DOI: https://doi.org/10.1016/j.comcom.2024.107975.
[36] Q Wang, R Li, Q Wang, et al. "Exploring Unfairness on Proof of Authority: Order Manipulation Attacks and Remedies," arXiv:2203.03008, 2022, DOI: https://doi.org/10.1145/3488932.3517394.
[37] P Ekparinya, V Gramoli, and G Jourjon, "The Attack of the Clones Against Proof-of-Authority," in Proc. NDSS, 2020, DOI: https://doi.org/10.14722/ndss.2020.24082.
[38] A Sonnino, S Bano, M Al-Bassam, et al. "Replay Attacks and Defenses Against Cross-Shard Consensus in Sharded Blockchains," in Proc. IEEE EuroS&P, 2020, DOI: https://doi.org/10.1109/EuroSP48549.2020.00026.
[39] M Decker and R Wattenhofer, "Bitcoin Transaction Malleability and Mt. Gox," in Financial Cryptography, 2014, DOI: https://doi.org/10.1007/978-3-319-11212-1_18.
[40] I Eyal and E G Sirer, "Majority Is Not Enough: Bitcoin Mining Vulnerable," in Financial Cryptography, 2014, DOI: https://doi.org/10.1007/978-3-662-45472-5_28.
[41] L Vasek and T Moore, "Tracking the Popularity and Profits of Virtual Currency Scams," in Financial Cryptography, 2015, DOI: https://doi.org/10.1007/978-3-662-47854-7_4.
[42] S Meiklejohn, M Pomarole, G Jordan, et al. "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names," in Proc. IMC, 2013, DOI: https://doi.org/10.1145/2504730.2504747.
[43] A H Khan, H Ikram, C M Ahmed, et al. "Energy Level Spoofing Attacks and Countermeasures in Blockchain-Enabled IoT," in Proc. IEEE GLOBECOM, 2022, DOI: https://doi.org/10.1109/GLOBECOM48099.2022.10001609.
[44] J Zheng, H Huang, Z Zheng, et al. "Adaptive Double-Spending Attacks on PoW-Based Blockchains," IEEE Transactions on Dependable and Secure Computing, 2024, DOI: https://doi.org/10.1109/TDSC.2023.3268668.
[45] K G Arachchige, P Branch, and J But, "An Analysis of Blockchain-Based IoT Sensor Network Distributed Denial-of-Service Attacks," Sensors, 2024, DOI: https://doi.org/10.3390/s24103083.
[46] Z Wu, J Liu, J Wu, et al. "TRacer: Scalable Graph-Based Transaction Tracing for Account-Based Blockchain Trading Systems," IEEE Transactions on Information Forensics and Security, 2023, DOI: https://doi.org/10.1109/TIFS.2023.3266162.
[47] W Lv, X Qiu, and L Meng, "Blockchain Localization Spoofing Detection Based on Fuzzy AHP in IoT Systems," EURASIP Journal on Wireless Communications and Networking, 2022, DOI: https://doi.org/10.1186/s13638-022-02094-7.

Ehsan Dastani M.Sc. Student in Computer Engineering from Sharif University of Technology (SUT), Tehran, Iran. He is a member of the Re- liable and Durable IoT Application and Network Laboratory (RADIAN). His main research interests include IoT vulnerability detection, binary analysis, and deep learning.

Mahdi Akbari Zarkesh received his Master's degree in Computer Engineering from Sharif University of Technology (SUT), Tehran, Iran, graduating in 2023. His primary research focuses on cloud computing, blockchain technology, and software development methodologies. He is particularly interested in designing scalable and secure cloud architectures, integrating blockchain for enhancing data integrity and transparency, and advancing software development practices, with an emphasis on agile methodologies and DevOps for improving system efficiency and quality.

Mahdi Davoodzadeh is a B.Sc. student in Computer Engineering at Sharif University of Technology (SUT), Tehran, Iran. He is a member of the Reliable and Durable IoT Application and Network Laboratory (RADIAN). His research interests focus on the reliability, resilience, and robustness of systems and networks.

Bardia Safaei received his Ph.D. in Computer Engineering from the Sharif University of Technology, Tehran, Iran, in 2021. He was a visiting researcher at the Chair for Embedded Systems, Karlsruhe Institute of Technology (KIT), Germany, from 2019 to 2020. He is a faculty member in the Computer Engineering Department at Sharif University of Technology, where he directs the Reliable and Durable IoT Applications and Networks Laboratory (RADIAN). His research focuses on power efficiency and dependability in IoT, wireless sensor networks, mobile ad-hoc networks, and cloud computing. Dr. Safaei received the ACM/SIGAPP student award at SAC'19 and was a member of the National Elites Foundation from 2016 to 2020. He is serving as an editor in Scientia Iranica Transactions on Computer Science & Engineering and Electrical Engineering. He has also served as executive chair of the 28th CSI International Computer Conference and is a board member of the Cyber-

Physical Systems Society of Iran (CPSSI). Additionally, he reviews for several prestigious journals and conferences, including IEEE Transactions on Mobile Computing, IEEE Transactions on Vehicular Technology, IEEE IoT Journal, IEEE Transactions on Cloud Computing, ACM Transactions on Storage, IEEE ICC, ACM/IEEE DAC, IEEE Sensors Conference, and IEEE WF-IoT.

Ali Movaghar received the BS degree in electrical engineering from the University of Tehran in 1977, and the MS and PhD degrees in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1979 and 1985, respectively. He was a professor with the Department of Computer Engineering with the Sharif University of Technology in Tehran, Iran and has been on the Sharif faculty and now he is a visiting Professor of University of Michigan. He visited the Institut National de Recherche en Informatique et en Automatique in Paris, France and the Department of Electrical Engineering and Computer Science with the University of California, Irvine, in 1984 and 2011, respectively. He worked with ATT Information Systems in Naperville, Illinois in 1985-1986, and taught with the University of Michigan, Ann Arbor, in 1987-1989. His research interests include performance/dependability modeling and formal verification of wireless networks, and distributed realtime systems. He is a senior member of the ACM.

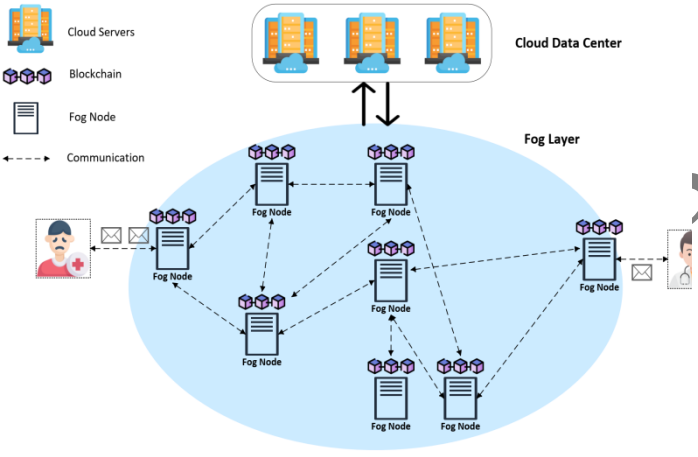


Figure 1: Architecture of EdgeLinker: Cloud Data Center, Fog Layers, End Users.


```
{
    "sender": "0xBa6B65f7A48636B3e533205d9070598b4faF6a0C",
    "nonce": 33,
    "req": {
        "heart-bit": 87,
        "time": 1688193528
    },
    "smart-info": {
        "address": "0x67fD63f6068962937EC81AB3Ae3bF9871E524FC9",
        "func": "ADDENTRY",
        "args": [
            "heart-bit"
        ]
    }
}
```

Sign Hash:

BMz15D10o+ebYrOG07X8p/4BBOwgyiBEjTjlBCID1tF9h/JVDfsGo7a4+XEdbae7krN/WEdjNw2xNc1+1lUA3NbMPW
DKJv6PR3uvoV5uE7W0q4mzfX2hZEoEQZpGsXiGiWzHT4Ql6lBwrM6YNWr7h9iWX9mHnoJQ2JhiARyQqA8YGB5k
75TCFWByqFh/8j7Kh9zfoeV/FTMrXhYB2A+peF8D+2l1F3D9apr72hNBBJ0nopguWgYXbxwm1sK6yAU5l9Z7m9P75
ezAXspeCG1BlMl5qP3YhdmPo2lxmrfbEgkaANa1vj1xWJ5+Awl7MB8rxxR6lhXLHwy7+2tM/qgeqA==

Figure 2: An example of a message transmitted through a secure channel.

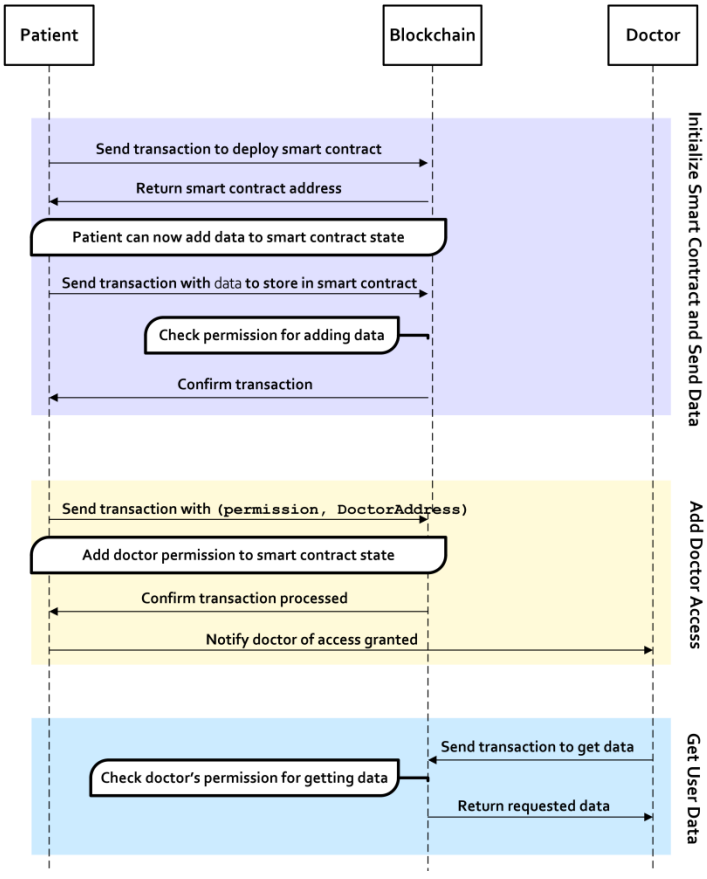


Figure 3: Sequence of requests in the experimental scenario.

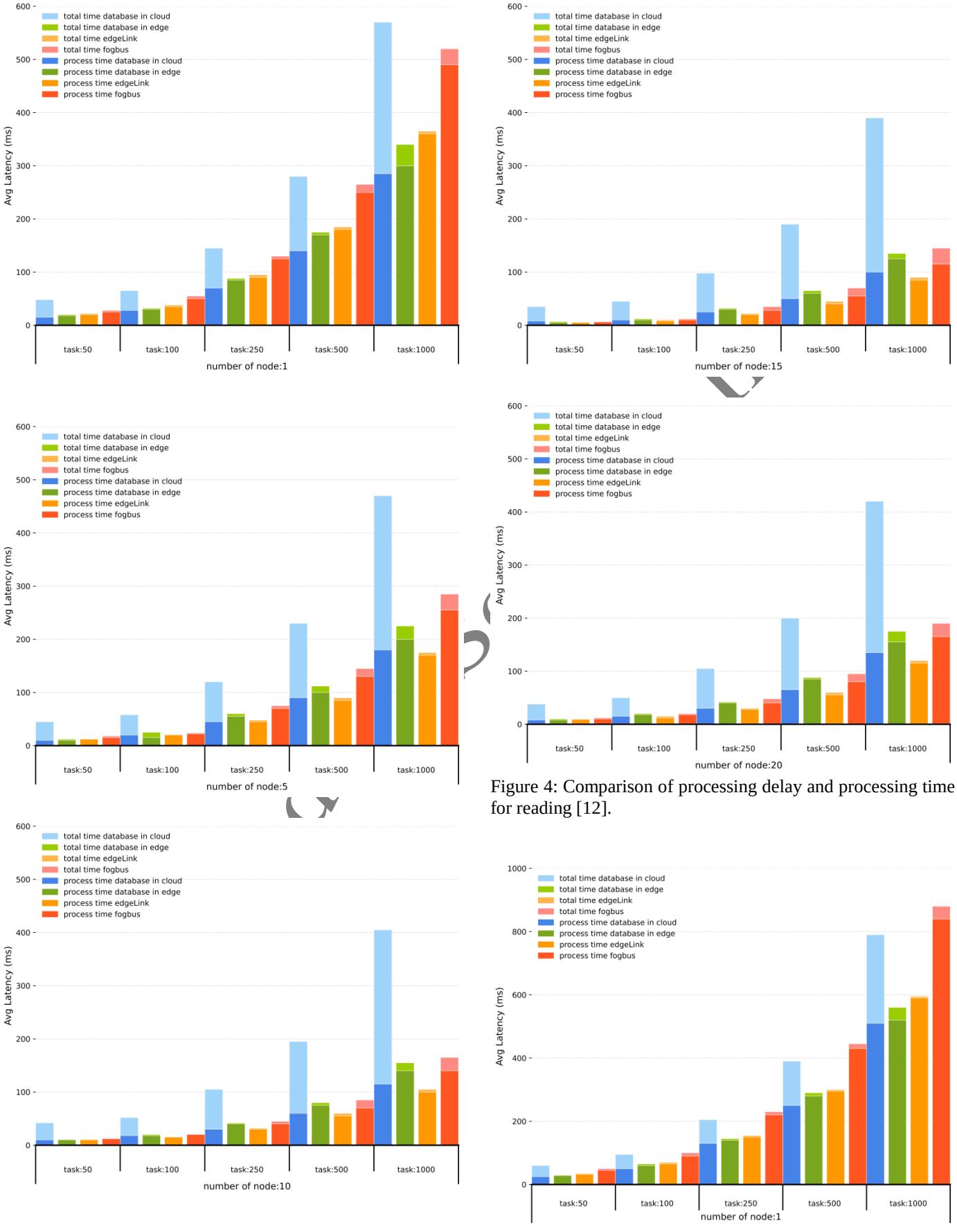
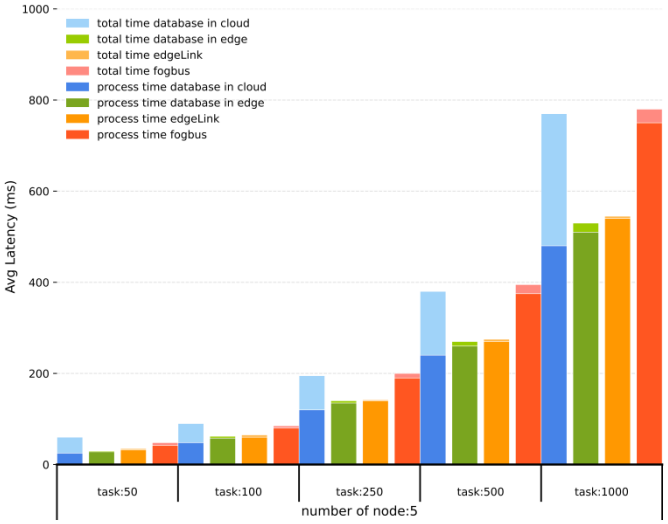
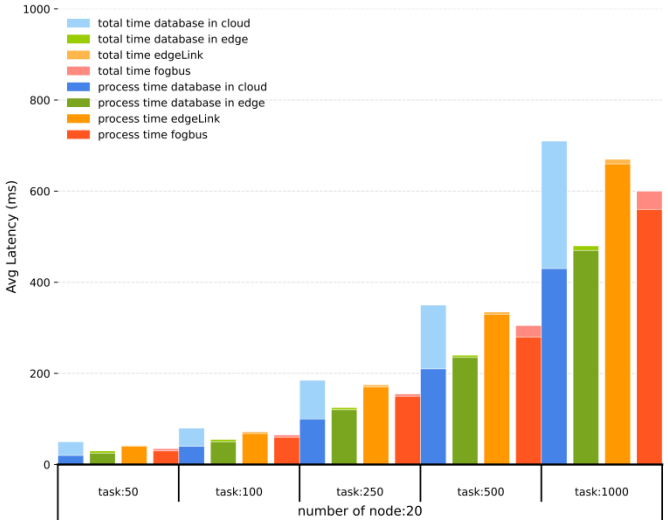


Figure 4: Comparison of processing delay and processing time for reading [12].





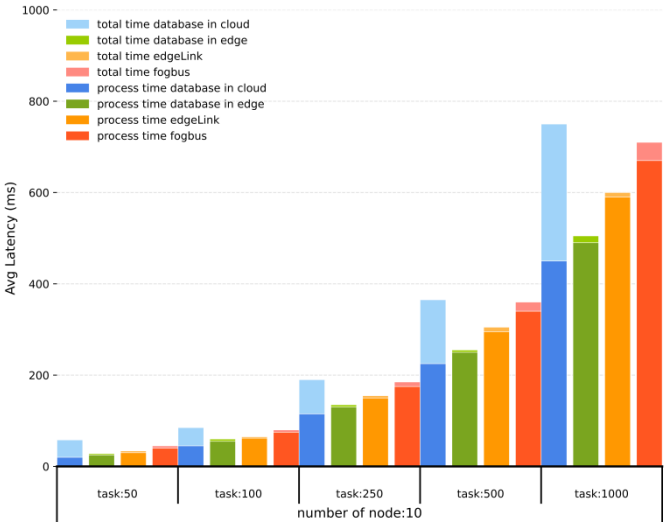


Figure 4: Comparison of processing delay and processing time for writing [12].

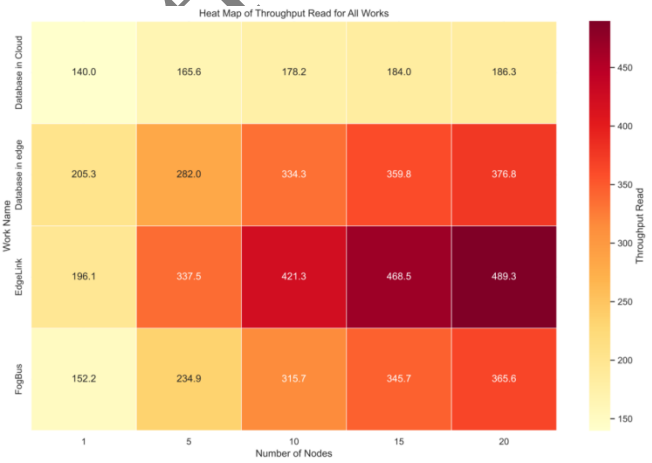
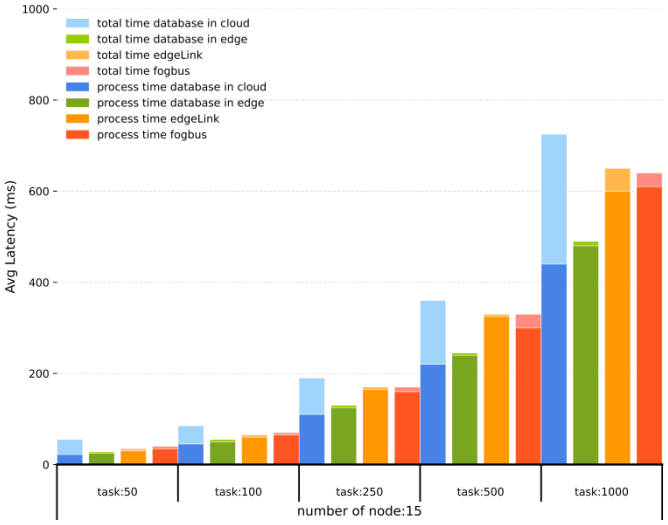


Figure 5: Read throughput analysis.



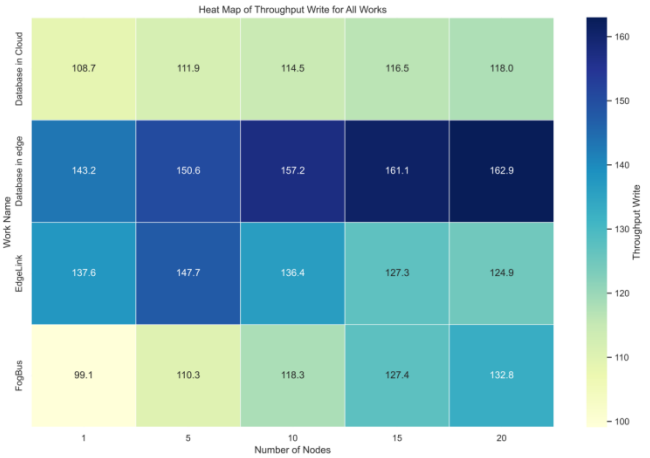


Figure 7: Write throughput analysis.

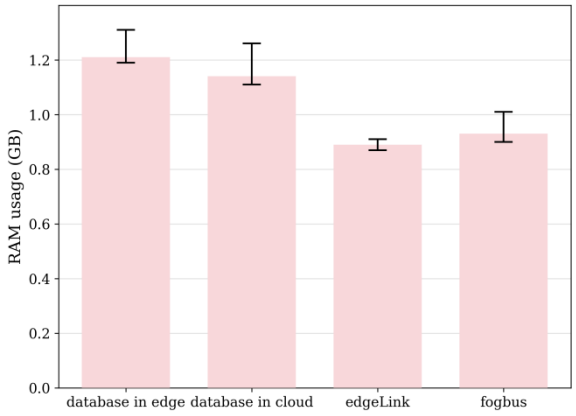


Figure 8: RAM consumption analysis.

Figure 10: Comparing message transmission time overhead in baseline (right) and secure communication channel (left) with EdgeLinker.

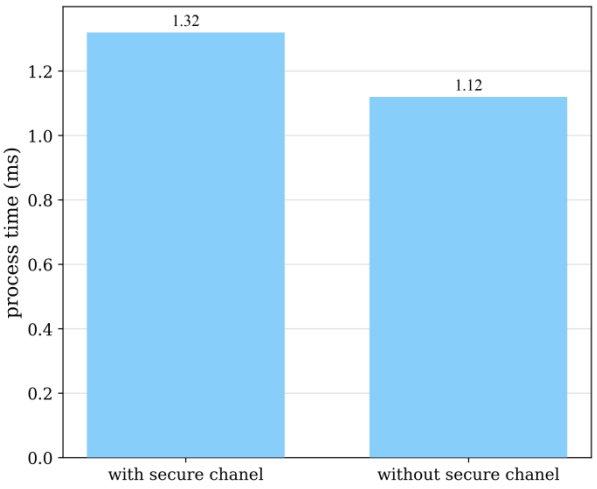


Figure 11: Comparing energy consumption in baseline (right) and secure communication channel (left) with EdgeLinker.

Table I: A comparison between the major existing studies and showcasing the features of EdgeLinker

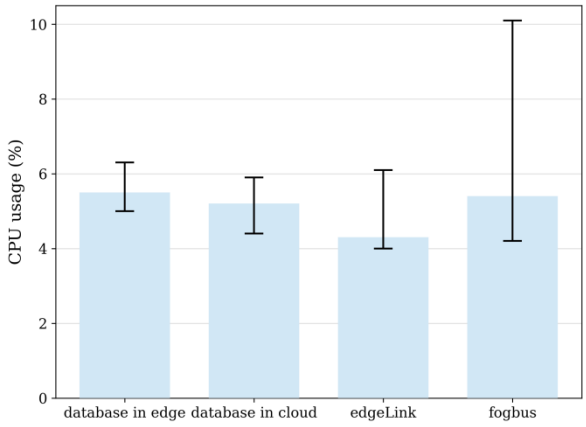
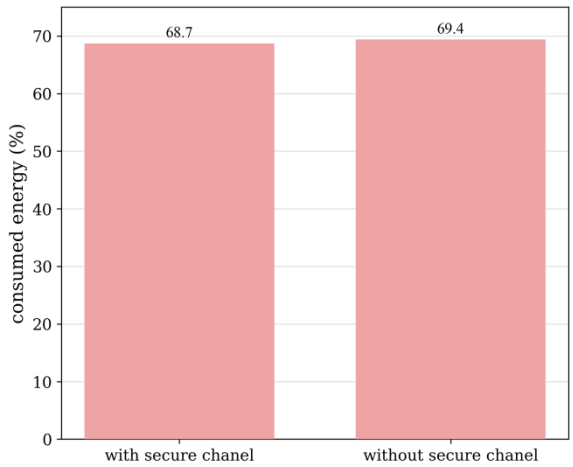


Figure 9: CPU consumption analysis.



Author(s)	Integration			Platform-Independent	Security Features			Multi-Application	Operational Feasibility	Decentralized Management
	IoT	Fog	Cloud		Integrity	Authentication	Confidentiality			
Chen et al. [20]	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Bruneo et al. [14]	✓	✓	✗	✓	✗	✓	✓	✓	✓	✗
Yi et al. [21]	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓
Liang et al. [15]	✓	✗	✓	✗	✗	✓	✓	✗	✗	✗
Shen et al. [22]	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓
Vora et al. [18]	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓
Azaria et al. [23]	✓	✗	✓	✗	✓	✓	✓	✗	✗	✓
Bhattacharya et al. [24]	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Yazdinejad et al. [25]	✓	✗	✓	✗	✓	✗	✗	✗	✗	✓
Sharma et al. [26]	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓
Ouyang et al. [27]	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
EdgeLinker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table II: System setup

Parameter	**Setup**
Front-end	React Native
Back-end	Node.js 18.16
Wallet	Sync 2
Deployment of Smart Contracts	Remix IDE + Thor
Consensus Algorithm	IBFT 2.0
Fog Node CPU Cores	2
Fog Node RAM	8GB
IoT device	Galaxy Watch 4 Classic
IoT device	Samsung Tablet
Communication Technology	Wi-Fi

Table III: Cost of deployment and use of smart contract

Cost in USD	**Gas Cost**	**Operation**
0.71	701382	Smart Contract Deployment
0.05	48182	Adding New Data
0.02	23521	Granting Permission
0.02	21984	Revoking Permission