Article │ Received 7 February 2025; Accepted 9 May 2025; Published 21 May 2025
https://doi.org/10.55092/blockchain20250011

Blockchain and smart contracts for secure and transparent salary grade structure management

Ebenezer Essel Mensah, Richard Kwasi Ahiable, Jonah Nud-Worgbah and Kofi Sarpong Adu-Manu*

Department of Computer Science, University of Ghana, Legon-Accra, Ghana

* Correspondence author; E-mail: ksadu-manu@ug.edu.gh.

Highlights:

- Blockchain-based salary management.
- Smart contracts in public payroll.
- Government payroll transparency.
- Permissioned blockchain security.
- RSA encryption in enterprise systems.

Abstract: Blockchain technology benefits companies in handling various use cases, including real estate, voting, fitness tracking, intellectual rights, the Internet of Things (IoTs), and vaccine distribution. Several technologies proposed in the literature seek to support businesses, enterprises, and state institutions in improving their operations and services, primarily in the financial sector. Although the existing technologies provide the needed service, the "trust" issue remains challenging. This differs from salary management in some state institutions in developing countries, such as Ghana. This paper presents a novel approach by implementing a permissioned blockchain-based system using Hyperledger Fabric integrated with RSA encryption to address the transparency, trust, and fraud challenges in salary-grade structure management. Unlike existing blockchain payroll applications, this work explicitly targets the salary grade adjustment processes within state institutions, providing a real-world prototype validated with actual agency data. In this paper, we implemented the blockchain technology for salary management. We use the Hyperledger Fabric platform to build a trusted platform to aid State Institution X (*siX*) in sharing data, validating transactions, securing data, and auditing transactions among its stakeholders— a prototype design aimed at reducing the wage bill and ensuring transparency in the public service payroll. The results showed that blockchain operations increased transparency in the payroll system among stakeholders by 100%. The application developed was secure and could track all the changes made by the relevant stakeholders in salary management.

Keywords: blockchain technology; financial security; financial systems; Hyperledger fabric; salary management; institutions

1. Introduction

Recently, enterprises and institutions have increased the automation of their business processes and centralised data management to serve their clients better [1]. Businesses rely on recent technologies such as big data, machine learning, blockchain, and similar innovations in Industry 4.0. These technologies provide unprecedented opportunities to improve performance. Blockchain brings novelty, efficiency, enhanced process transparency, and real-time data-sharing and exchange [2]. Blockchain technologies have proven worthy and have served their purpose since their adoption in the financial sector, overcoming the trust challenge among parties [3].

Blockchain may be a viable answer to record-keeping problems in developing countries like Ghana. Today, digital technology leads to greater specialisation, better utilisation of knowledge on the one hand, and increased productivity [4]. Many multinational organisations are now decentralised, giving network thresholds considerable leverage. This tendency has prompted corporations and institutions to consider how decentralised information systems and applications communicate.

Furthermore, corporate goals may be jeopardised if the advanced data are shaky or require reliability. For example, the deceptive enrollment parts of a registration procedure may imply that individuals cannot establish their identification as a fundamental precondition for obtaining social benefits or that this opens the way to recognisable fraud that arises with identity concerns [5]. In enterprises, this issue of trust and transparency occurs, which blockchain technology can address. This paper comprehensively investigated the use of blockchain technology and smart contracts to manage salary-grade structures safely and transparently. The emphasis is on overcoming the difficulties of trusting salary management within Ghana's public organisations. This paper aims to improve stakeholder data sharing, transaction validation, data security, and auditing by deploying a blockchain-based system using the Hyperledger Fabric platform. The contributions of this paper include increased transparency, the development of a secure application, and the utilisation of an encryption and decryption technique to secure and encrypt the blockchain system. The blockchain technique implemented in this paper significantly improved transparency in the payroll system, achieving a 100% increase in transparency among stakeholders. The developed application ensured the secure tracking of all changes made in salary management by the relevant stakeholders. The RSA asymmetric encryption technique was utilised to secure and encrypt blockchain systems, providing data reliability.

The paper aims to design and implement a data management application prototype based on blockchain technology to explore the variability of using blockchain to share data, validate transactions, secure data, and audit transactions. This will help to reduce the wage bill and ensure transparency in the public service payroll. Transparency has been a significant issue in the huge wage bill of country D, as most public workers have found ways to sway officers in charge of the payroll system to change their grades. Blockchain technology has come to eliminate problems with transparency in payroll systems owing to its encryption and decryption abilities.

The novelty of this study lies in combining permissioned blockchain (Hyperledger Fabric) with RSA asymmetric encryption and RESTful API integration for real-world salary-grade structure management. To the best of our knowledge, this is among the first implementations tailored for transparency and fraud prevention in public sector payrolls of developing countries, focusing on role-based access and

auditability of grade changes. Furthermore, the system is validated with institutional data, providing practical insights into its scalability and performance.

The remainder of this paper is organised as follows. Section 2 describes the blockchain technology and discusses security issues. Section 3 presents the proposed encryption and decryption strategies. In Section 4, the system architecture and implementation are discussed. Section 5 describes the algorithm design and process flow. Section 6 presents the results and discussion, and Section 7 concludes the paper.

2. Blockchain technology

Blockchain technology is being increasingly researched as a solution to privacy, transparency, and security challenges owing to its inherent security, reliability, immutability, and transparency. This applies to various enterprise situations beyond the financial and banking sectors. For example, it may be used in supply chain systems, financial markets, healthcare, insurance, manufacturing firms, government agencies, the Internet, and agricultural product tracking [6]. In addition, blockchain technology can evaluate the transparency and traceability system (TTS) between nodes. It can operate on Ethereum, IBM Blockchain, Hyperledger Fabric, Hyperledger Sawtooth, R3 Corda, Tezos, EOSIO, Stellar, and Quorum [7].

Blockchain technology addresses digital transaction privacy, transparency, and security concerns using decentralisation, immutability, and cryptographic security [8]. Decentralising data storage improves privacy while reducing the risk of vulnerabilities and unauthorised access. The transparency of an immutable ledger promotes participant confidence, whereas cryptographic approaches provide strong security [9]. Blockchain platforms, such as Ethereum, Hyperledger Fabric, and R3 Corda, cater to specific needs. Ethereum supports smart contracts and DApps, whereas Hyperledger Fabric is ideal for enterprise use because of its modular architecture and scalability [10]. R3 Corda focuses on peer-to-peer transactions in regulated environments, offering high throughput and confidentiality for applications in finance and trade. Blockchain technology across platforms provides secure and efficient transaction recording, transforms industries, and enables new business models. Blockchain revolutionises supply chain management by offering a transparent platform for tracking goods and information and ensuring authenticity and compliance. In healthcare, it enables secure data sharing among providers, leading to informed decision-making and improved patient outcomes while streamlining administrative processes [11,12].

Securing transactions stored in a blockchain is crucial for maintaining data integrity, confidentiality, and authenticity. Encryption techniques safeguard sensitive information and prevent unauthorised access or tampering. By encrypting the data before they are added to the blockchain, participants can ensure that only authorised parties with the corresponding decryption keys can access and verify the transactions. This cryptographic security mechanism protects against malicious attacks and data breaches and enhances trust and confidence in the blockchain ecosystem. Furthermore, encryption helps mitigate data manipulation, fraud, and identity theft. By encoding transactional data using cryptographic algorithms, blockchain participants can ensure that any changes or modifications to the data are detectable and traceable, thereby preserving the immutability of the blockchain ledger. This transparency and immutability fosters accountability and trust among participants, reinforcing the reliability and credibility of the blockchain network.

In discussing the financial applications of blockchain technology, it is essential to acknowledge the role of cryptocurrency markets in popularising the adoption of blockchain. Cryptocurrencies such as Bitcoin and Ethereum have demonstrated blockchain’s ability to provide decentralised, transparent, and

secure financial transactions. These applications have showcased the potential of blockchain to disrupt traditional financial systems, influencing how blockchain solutions are considered for non-cryptocurrency domains, including enterprise salary management. A detailed analysis of the cryptocurrency market, its volatility, and its underlying blockchain mechanisms is presented in [13], highlighting blockchain technology's broader economic and technological significance.

3. Encryption and decryption in blockchain

3.1. RSA asymmetric encryption technique

The RSA asymmetric encryption technique is an essential cryptographic algorithm for securing blockchain transactions. The RSA uses public and private keys instead of symmetric encryption, which uses the same key for encryption and decryption. This asymmetric key pair ensures that data encrypted with the public key can only be decrypted with the accompanying private key, thereby increasing security and confidentiality. The RSA offers several advantages in the blockchain context, including robust protection against brute-force attacks and sophisticated cryptographic security.

The RSA algorithm relies on complex modular arithmetic and prime number factorisation, making it computationally infeasible for attackers to derive private keys from public keys. This ensures that the encrypted data remains secure, even against determined adversaries. Moreover, RSA is well suited for handling large datasets and high-volume transactions, making it ideal for scalable blockchain applications. By encrypting transactional data using RSA, blockchain participants can ensure that sensitive information remains confidential and tamper-proof, upholding the principles of privacy, transparency, and security in the blockchain ecosystem. Encryption and decryption are integral to blockchain technology, as they ensure transaction confidentiality, integrity, and authenticity. The RSA asymmetric encryption technique, with its robust security features and scalability, has emerged as a critical enabler for securing transactions on the blockchain and fostering trust and confidence among the participants.

In our implementation, we chose the RSA asymmetric encryption technique instead of the Elliptic Curve Digital Signature Algorithm (ECDSA), which is commonly used in public blockchains such as Bitcoin and Ethereum. Our system is built on a permissioned blockchain (Hyperledger Fabric) where all participants are known and authenticated entities. RSA is highly suitable in such environments due to its widespread adoption, ease of key management, and strong security guarantees. Additionally, RSA provides straightforward integration with enterprise security infrastructure and certificate authorities, aligning well with the Membership Service Provider (MSP) system in Hyperledger Fabric. While ECDSA offers performance benefits for public and large-scale anonymous networks, RSA remains a reliable and practical choice for permissioned, enterprise-grade applications like salary management systems. We have, therefore, used RSA to ensure secure transaction signing and data confidentiality in our implementation.

3.2. Modeling RSA for large databases

The RSA encryption method is tailored to large databases, ensuring high security through complex modular arithmetic and prime number vectors. The encryption process involves Euler's function and modular exponentiation, making brute-force attacks impractical because of the extensive computation

required. According to [14], blockchain technology is essential for salary management because it has a high potential to ensure data reliability. Blockchain allows transactions to be stored and viewed by all peers in the network, and securing and encrypting the transactions is critical. In this paper, the authors implemented the RSA asymmetric encryption technique to secure and encrypt logical blockchain systems that store large amounts of data in blocks. We modelled the RSA system for large databases as follows:

Considering salary data *N*, *where N is defined as a set of elements*, $N = \{n_1, n_2, n_3, \ldots, n_m\}$, the following computations were performed to accomplish the encryption and decryption procedures.

$$N = B \cdot C \tag{1}$$

From Equation (1), the set $N = (n_1, n_2, n_3, n_4, \ldots, n_m)$ may be modular numbers for open and closed switches, $B = (b_1, b_2, b_3, \ldots, b_m), C = (c_1, c_2, c_3, \ldots, c_m)$ are vectors composed of large prime numbers. Taking the vectors and assigning them to the elements in *N*, we arrive at Equation (2).

$$\begin{aligned} n_1 &= b_1 c_1 \\ n_2 &= b_2 c_2 \\ n_3 &= b_3 c_3 \\ n_m &= b_m c_m \end{aligned} \tag{2}$$

From Equation (2), we compute the Euler function for sets of integers.

$$\varphi(n_i) = (b_i - 1)(c_i - 1) \qquad i = 1,2,3, \ldots, m \tag{3}$$

In the order of numbers in the result of calculations (2), (3), we choose the numbers $E\ (e_1, e_2, e_3, \ldots, e_i)$ from the range of sets of numbers from 1 to $\varphi(n_i)$ as shown in Equation (3).

$$c_i e_i mod(\varphi(n_i)) := 1 \quad i = 1,2,3, \ldots, n \tag{4}$$

where $E\ (e_1, e_2, e_3, \ldots, e_i)$ is an integer (exponent).

As a result of Equation (4), the numbers, c_i, $i = 1,2,3, \ldots, n$ are computed. Hence, the RSA encryption algorithm for large databases is written as presented in Equation (5).

$$D_i = M_i^{Ei} mod(N_i) \qquad i = 1,2,3, \ldots, m \tag{5}$$

hence,

$$\begin{aligned} d_1 &= m_1^{e1} mod(n_1) \\ d_2 &= m_2^{e2} mod(n_2) \\ &\ldots \\ d_m &= m_m^{em} mod(n_m) \end{aligned} \tag{6}$$

D_i is the set of encrypted messages. In Equation (6), $d_1, d_2, d_3, \ldots, d_m$ represents the sum of the encrypted numbers used to calculate the number of encrypted messages presented in Equation (7).

$$\sum_{i=1}^{m} D_i = \left(\sum_{i=1}^{m} M_1^{e_1}\right) \tag{7}$$

where, D_i is the sum of the numbers and the parts of the standard encrypted text. Using Equations (4), (5), and (6), we derive the canonical equation of decryption (see Equation (8)) as follows:

$$\sum_{i=1}^{m} M_i = \left(\sum_{i=1}^{m} D_1^{e_1}\right) mod(N_i) \tag{8}$$

From Equations (7) and (8) using the decryption scheme $\{Ci, Ni\}$, a set of private keys of the system and a set of public keys, $\{Di, Ni\}$ are adopted. In applying electronic digital signatures in blockchain technology, open and closed keys are used, and calculations are performed as a result of

Equations (7) and (8). When applying logic blockchain technology, a private key is used to generate digital signatures on the data in the blocks. A public key was used to verify the electronic signature [14]. Storing data in blocks is calculated based on public and private keys. In this case, the inverse calculation requires large-scale calculations. To find a private key, one must iterate over 2^N combinations, where N is the *key* length. Even in the most modern high-performance clusters, brute-force *key* selection takes longer. For example, with a key length of 256 bits and a password brute-force rate of 1024 per second, it will take 1.23e + 67 years. Thus, the encryption method used in logistic-blockchain systems is highly secure. Another important task in protecting information that affects logistics blockchain solutions is to ensure user confidence [15]. A model that provides user trust is shown in Figure 1. All transactions between parties in such networks are fragmented and decentralised globally.

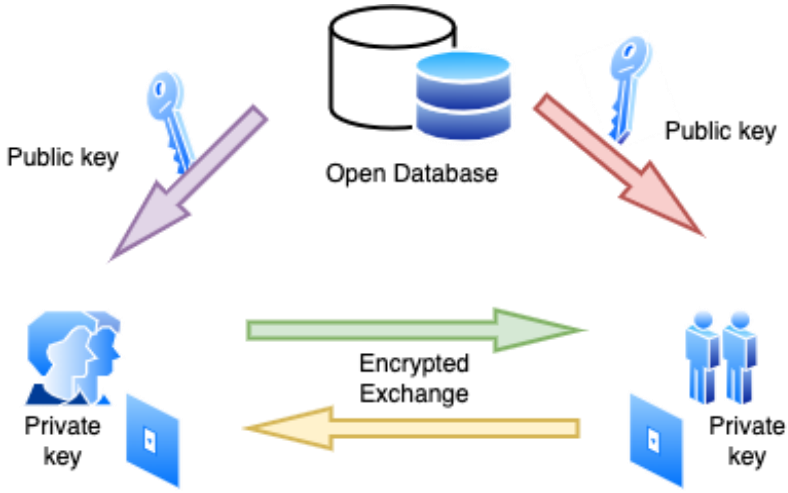

Figure 1. Model of data exchange between users.

3.3. Access control and security

3.3.1 Permissioned blockchain and access control

Access control in permissioned blockchains is critical to maintaining security and privacy. In systems such as Hyperledger Fabric, the Membership Services Provider (MSP) manages identities and ensures that only authorised participants access the network. The MSP utilises the Fabric Certificate Authority (CA) to issue digital certificates, which serve as credentials for participants, ensuring that only authenticated entities can initiate transactions and access data [11], [16]. This approach aligns with the principles of the ZeroTrust architecture, where the continuous verification of participants is crucial for enhanced security and privacy [16]. Each Hyperledger Fabric channel has access control policies that specify permitted actions for different roles. This granularity allows for a specific control over who can read, write, or update the ledger. For instance, in a salary management system, HR managers may be able to update salary records, whereas others may only view or approve these records [12], [17].

3.3.2. Membership Services Provider (MSP) and Fabric CA

The MSP and Fabric CA are central to Hyperledger Fabric's identity and access management systems. The MSP defines identity governance rules within the network, with each organisation having its own MSP to ensure uniform adherence to identity policies. Fabric CA issues digital certificates that authenticate users and devices and provides a secure credential management method [16]. Participants use these certificates to sign transactions, ensuring traceability and verifiability of all actions on the blockchain. This mechanism prevents unauthorised access and ensures that all activities are conducted by legitimate participants, thereby enhancing network security and trustworthiness [18,19].

3.3.3. Transaction consensus and ledger updates

Transaction consensus in Hyperledger Fabric involves multiple steps to ensure that only valid transactions are recorded. When a transaction proposal is initiated, it is sent to endorsing peers for validation, where they simulate the transaction and verify compliance with smart contract rules. Endorsed transactions are sent to the ordering service, packaged into blocks, and delivered to committing peers [20,21]. Committing peers verify transaction endorsements and update ledgers accordingly. This consensus mechanism ensures consistent validation and recording of transactions and maintains the ledger's integrity and accuracy. Hyperledger Fabric inherently prevents blockchain forks using a deterministic transaction endorsement and validation model. In this system, transactions are first endorsed by designated peers according to endorsement policies and then validated by all committing peers before being appended to the ledger. Suppose two conflicting transactions are submitted (e.g., two simultaneous updates to the same record). In that case, the Fabric validation process detects the conflict, and only one transaction is accepted based on ordering, while the other is automatically rejected without requiring manual intervention.

This design ensures ledger consistency, avoids chain forks, and maintains the integrity of the blockchain without human arbitration. By leveraging this robust process, Hyperledger Fabric ensures the blockchain remains secure, transparent, and tamper-proof [22,23].

4. System architecture and implementation

4.1. Network setup and smart contract design

In this paper, we utilised a well-designed architecture consisting of a network of one (1) channel (*myChannel*) and one smart contract (*gradeContract*), as shown in Figure 2, for our implementation. The contract is tailored to the business domain (managing grade structure), controls the lifecycle of a single asset type, and can be triggered by approved participant roles. The client application communicates with the Hyperledger Fabric system in three diverse ways: (1) querying the ledger, (2) sending a new transaction, and (3) receiving a report of the changes to the ledger.

To interact with the Fabric topology, we utilised the Hyperledger Fabric application SDK to interface with the Fabric topology. The SDK safeguards an application from the topology's complexities while permitting it to capitalise on it, irrespective of the number of nodes with which it must communicate or how its organisations, ledgers, smart contracts, approval policies, peers, and ordering services are built. Furthermore, as the network topology changes, SDK enables functionality to leverage

potential application advantages, such as enhanced reliability or robustness. For instance, if an applicant's organisation has two peers, the SDK will use the available peer if the other is not running, or if one peer fails, the SDK will switch to an accessible peer to complete execution. User applications take advantage of all network features without worrying about the network's implementation details.

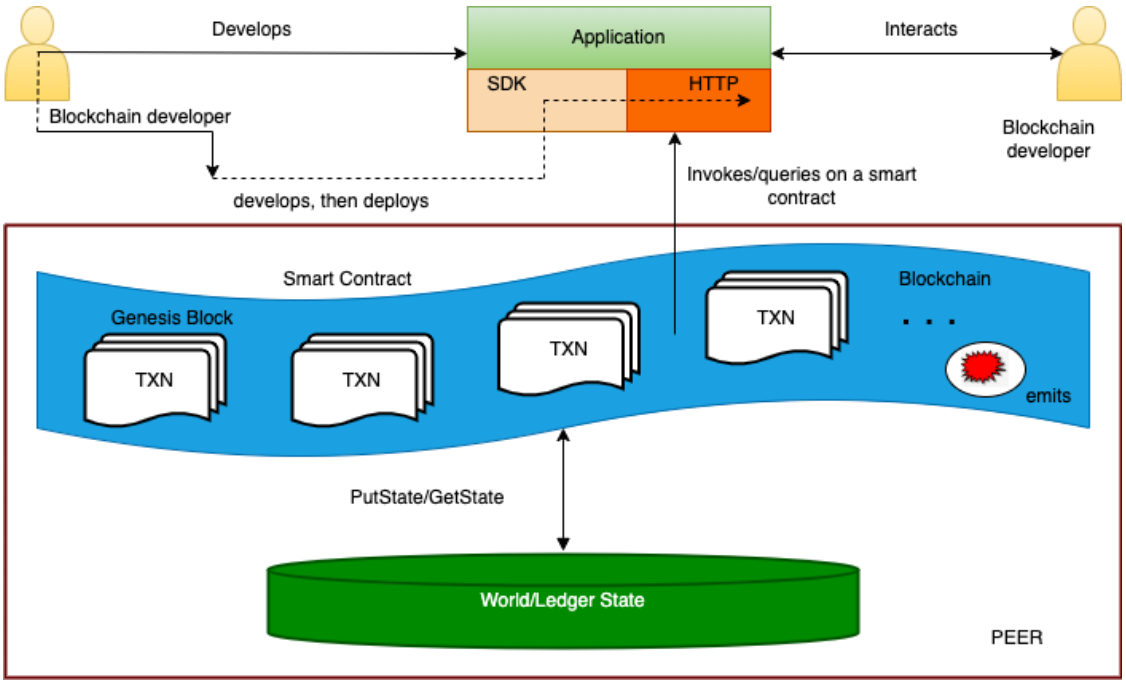

Figure 2. Hyperledger Fabric architecture.

4.1.1. The network participants

From Figure 3, we set up a network with one (1) organisation and two (2) peers. The network participants included state institutions (*siX*), (*siY*), and (*siZ*), as shown in Figure 3. The network consists of state institutions with varying access levels for the grade structure data.

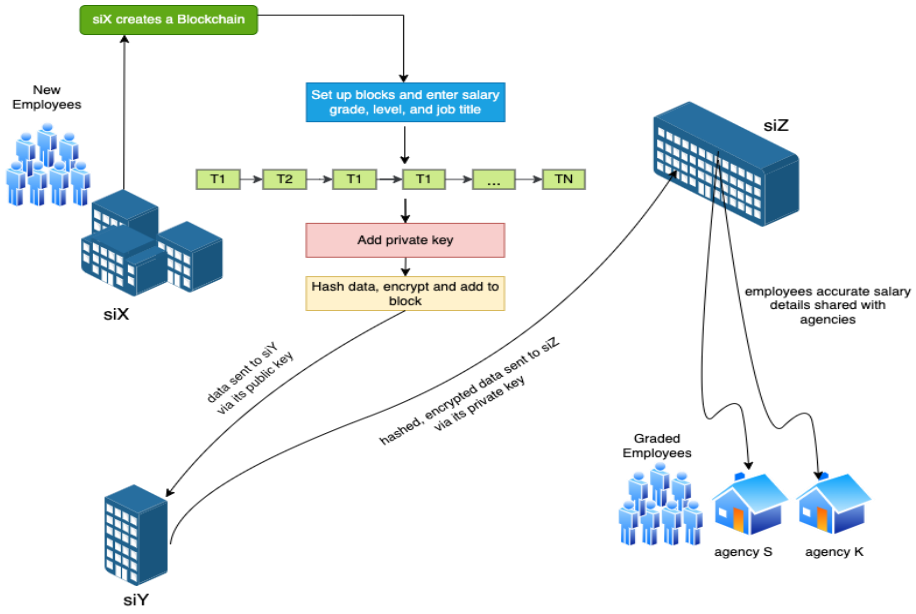

Figure 3. Blockchain for salary management.

4.1.2. Smart contract and transactions

This paper defines an asset as the grade structure in which users on the hyperledger network can read or write using a chaincode process. Transactions recorded on the network with limits to each user are categorised as create, read, update, and deletion transactions, as shown in Figure 4.

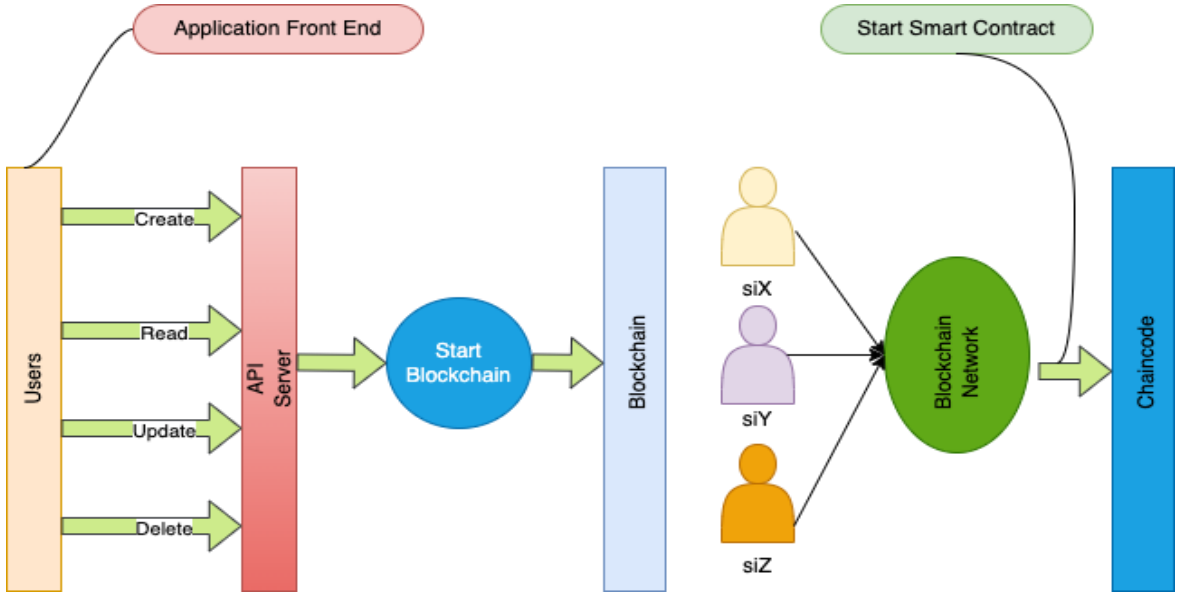

Figure 4. Blockchain model architecture for salary structure system.

The *Create Transaction (CT)* allows institutions to create new records in the ledger. In the context of this paper, *CT* can be used to add a new grade structure to the ledger. When users initiate a *CT*, they propose adding a new record to their ledger. This proposal is sent to all endorsing peers in the network to obtain agreement on the transaction's legitimacy. Once all endorsements have been verified, the road to building a block begins, and the ledger is updated.

An *Update Transaction (UT)* is a transaction type that allows users to modify an existing record in the ledger. In the context of this project, *UT* can be used to update the existing grade structure in the ledger. When users initiate a *UT*, they propose modifying the existing record in the ledger.

This proposal was sent to all endorsing peers in the network to obtain agreement on the transaction's legitimacy. Once all endorsements have been verified, the road to building a block begins, and the ledger is updated. To strengthen the security of transaction processing, each *UpdateTransaction* operation includes a timestamp field. This design ensures that transactions are uniquely identifiable in time, effectively mitigating the risk of replay attacks where an adversary could otherwise resend valid but previously recorded transactions. Additionally, to promote transparency and enable independent security verification, the smart contract (chaincode) developed for this system will be made publicly available upon the finalization of the project. This will allow for third-party auditing, fostering greater trust and accountability in the proposed blockchain-based salary management framework.

The *Delete Transaction (DT)* transactions allow users to remove existing records from the ledger. In the context of this project, the *DT* can be used to remove an existing grade structure from the ledger. When a user initiates a *DT*, they propose removing the existing record from the ledger. This proposal

was sent to all endorsing peers in the network to obtain agreement on the transaction's legitimacy. Once all endorsements have been verified, the road to building a block begins, and the ledger is updated.

A secure and permissible blockchain requires access control to function correctly. The access control technique is integrated into the contract and enforced during transaction processing of multiple endorsing peers, and the result is validated through transaction consensus. In contract- interacting application layers, additional access control measures can be used. The membership service provider (MSP) is critical for allowing fabric access control. Each Fabric network organisation may have one or more MSP providers. The MSP uses a fabric certificate authority (CA). The blockchain network participants have rules controlling how they access the data or the asset. *siX* has permission to create, update, and delete the grade structure. *siY* has access to read the grade structure and change its status (*i.e.*, Approve or Decline). *siZ* has only read access (that is, *siZ*'s role is to inform respective employees of the approved salary).

In the current prototype, we employed a single-channel architecture for simplicity and ease of demonstration. However, a multi-channel architecture will be essential for a production-grade deployment involving multiple institutions to achieve fine-grained data isolation and privacy among stakeholders. Each channel can host its ledger and smart contracts, ensuring that only authorised members within a particular channel have visibility and transaction rights. This design prevents cross-institutional data leakage and enhances confidentiality.

4.2. Hyperledger Fabric application SDK

Each peer in the Hyperledger Fabric network has a committer role by default during the channel, Smart Contract, and channel policy creation. A peer can perform various functions such as Endorser, Leader, and Anchor. The client application performs transactions within the network by sending transaction proposals or proposing transactions to endorse peers. The client application requires the customer SDK to know the rundown of all-embracing peers in the organisation. Each embracing peer has a chain code introduced, just as a duplicate of the record in harmony with any remaining parties. The transaction proposal is generated using the SDK API. This solicitation allows a chain code to be summoned to peruse or write in the record. The request is then packed into a valid package that the network accepts. A signature for this proposal was generated using the client's cryptographic credentials. After forming the request, it is shipped off to all underwriting peers in the organisation to obtain an agreement on the exchange's legitimacy. The road to building a block begins once all endorsements have been verified (see Figure 5). Kafka is utilised in this network to provide error detection and high-throughput, low-latency real-time feeds. Peers check the validity of the endorsement according to the chain code and update the ledger.

Figure 5. Updating the blockchain.

We selected Kafka as the ordering service due to its durability, high throughput properties, and extensive documentation support at the time of system design. However, we acknowledge that newer versions of Hyperledger Fabric recommend Raft for its simplicity and native crash fault tolerance (CFT) without external dependencies. Future implementations will consider comparative performance between Kafka and Raft, especially under different network latencies.

4.3. API integration and data exchange

This paper employed the RESTful (REST) API to facilitate communication between the Hyperledger and the client application. To connect the Hyperledger to the client application, an API is required to assist in the application connection process, as illustrated in Figure 6.

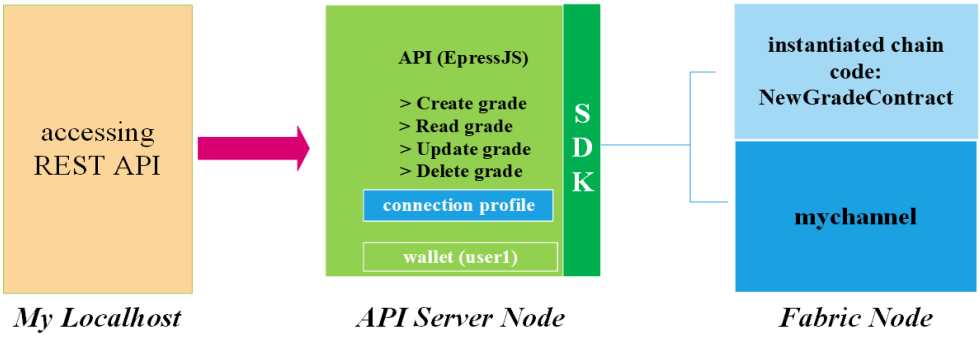

Figure 6. API integration architecture.

4.3.1. RESTful API with ExpressJs and SQLite3

A RESTful API, designed with ExpressJs and SQLite3, bridges the Hyperledger Fabric network and client applications. This setup ensures secure, robust, and seamless interaction, allowing users to query the ledger, initiate transactions, and receive updates without delving into the complexities of the network.

The API was designed using ExpressJs and SQLite3, two powerful tools for building web applications. ExpressJs is a fast, unopinionated, minimalist web framework for Node.js, specifying robust descriptions for web and mobile applications. SQLite3 is a lightweight and efficient database engine for embedded systems, mobile devices, and desktop applications. It offers a stand- alone file, is highly dependable and quick, and is designed to work in a battleship without internet access. Using these tools, we created a secure and reliable API enabling seamless communication between the Hyperledger Fabric network and the client application.

4.4. Communication between Hyperledger and client application

The communication between Hyperledger and the client application is vital to the functionality and performance of the system (*i.e.*, salary grade structure management) proposed in this paper. The interaction ensures secure, efficient, and transparent data exchange, providing a robust communication framework between the Hyperledger and the client's application. Using HTTP requests, the RESTful API enables the client application to query the ledger, initiate transactions, and receive updates without addressing the complexities of the underlying blockchain network. The API ensures that the client application interacts seamlessly with the blockchain network, supporting the system's functionality. It

also provides secure interactions through encryption and protocols. The system uses RSA asymmetric encryption to protect data during transmission, ensuring that sensitive information, such as salary grades and transaction details, is encrypted before being sent over the network. This encryption mechanism safeguards data from unauthorised access and tampering, thereby maintaining the integrity of the salary management system.

In our system, real employee identities (e.g., names, personal IDs) are stored off-chain in a secure database, with blockchain transactions referencing them through hashed identifiers. While this design improves performance and protects sensitive data from direct exposure on-chain, we recognize that an attacker could potentially de-anonymize the blockchain records by linking hashed IDs back to real individuals if the off-chain database is compromised. To address this risk, future system enhancements will implement salting techniques — adding random values to identifiers before hashing — to make reverse mapping more difficult. Additionally, adopting Decentralized Identity (DID) frameworks, where identity credentials are cryptographically secured and controlled by users, will be explored to enhance privacy and resilience against off-chain leaks further.

The algorithms presented in Section 7 facilitate efficient data exchange by employing hashing, encryption using private keys, and verification using public keys. These processes ensure that all salary grade and transaction records are transparent and tamper-proof. The API design supports high throughput and low latency, which are critical for system performance. By optimising these parameters, the system can handle a large volume of transactions and queries in real time, thereby ensuring smooth operation and user satisfaction. Smart contracts are integral to the system architecture, automating the execution of transactions based on predefined rules and conditions. This automation reduces the need for intermediaries, streamlines processes, and ensures compliance with organisational policies. The client application interacts with these smart contracts through the API, enabling the automated and secure execution of salary-related transactions. The system architecture, which includes a network of one channel and one smart contract (gradeContract), supports robust and efficient management of the grade structure. The client application's interaction with the Hyperledger Fabric system through the API allows querying the ledger, sending new transactions, and receiving updates on changes. This architecture ensures all stakeholders have real-time access to accurate and up-to-date information, promoting organisational transparency and trust.

5. Algorithm design and process flow

The algorithms presented in this paper facilitate secure data exchange among state institutions (*siX, siY, and siZ*) using blockchain and smart contracts. They involve hashing, encryption with private keys, and verification with public keys, ensuring transparent and tamper-proof record keeping of salary grades and transactions.

5.1. Creation, encryption and verification of data blocks

The algorithms described herein involve creating, encrypting, sending, and verifying data blocks among three state institutions (*siX, siY, and siZ*) using private and public keys. These data blocks contain information about the *gradeID*, *salary grade*, *level*, and *job title* of the employees of each institution, and they are implemented using blockchain and smart contracts. Blockchain technology allows for

secure, transparent, and decentralised data storage and transfer using a network of nodes that validates and records transactions in a shared ledger. Smart contracts are digital agreements written in code, stored in a blockchain, and executed automatically without intermediaries when certain conditions are met.

Algorithm 1: siX process Start

```
Start
   For each state institution (siY, siZ) do
      siX enters the gradeID
      siX enters salary grade, level, and job title
      siX enters the private key
      Hash the data
      Encrypt the hashed data
      Add the encrypted block to the blockchain
      Send the encrypted data to siY via its public key
   End For
End
```

```
# Code Representation of Algorithm 1

def siX_process(siY, siZ, gradeID, salary_grade, level, job_title, private_key):
   # Step 1: Hash the entered data
   hashed_data = hash_data(gradeID, salary_grade, level, job_title)

   # Step 2: Encrypt the hashed data using the private key
   encrypted_data = encrypt_data(hashed_data, private_key)

   # Step 3: Add the encrypted data block to the blockchain
   add_block_to_chain(encrypted_data)

   # Step 4: Send the encrypted data to siY using its public key
   send_data_to_siY(siY, encrypted_data)
```

Using blockchain and smart contracts, state institutions streamline their payroll systems, reduce errors and fraud, and ensure compliance with the agreed terms. For example, *siX* creates a smart contract that specifies each employee's salary grade, level, and job title, encrypts it with its private key, and sends it to *siY* using its public key. *siX* appends its private key to the entered data, and then the data are hashed, encrypted, and added to the blockchain. When a key fails, *siX* must provide a valid key to process the data for the chain. The data were hashed and encrypted by *siX* and decrypted by *siY* for verification and approval. *siY* decrypts the data with its private key, verifies its accuracy, and adds its data to the block. If any issues exist, *siY* sends an error message back to *siX* using another smart contract, as indicated in *Algorithm 1*. *siY* appends its private key when no issues exist. The data were hashed, encrypted, and added to the blockchain.

Algorithm 2: siY process

```
Start
   siY enters the private key
   If the private key is invalid then
      Display error message
   Else
      Decrypt the received data
      If data contains errors then
         siY enters a description of the error
         siY enters the private key
         Hash the error description
         Encrypt the hashed data
         Add the encrypted block to the blockchain
      Else
         siY enters the private key
         Hash the validated data
```

```
# Code Representation of Algorithm 2
def siY_process(private_key, data):
   # Step 1: Decrypt the received data
   decrypted_data = decrypt_data(data, private_key)

   # Step 2: Check if the decrypted data contains errors
   if contains_errors(decrypted_data):
      # Step 2a: Handle the error case
      error_description = enter_error_description()
      private_key = enter_private_key()

      # Step 2b: Hash and encrypt the error description
      hashed_data = hash_data(error_description)
      encrypted_data = encrypt_data(hashed_data, private_key)
```

```
        Encrypt the hashed data
        Add the encrypted block to the blockchain
        Send the encrypted data to siZ
      End If
    End If
End
```

```
    # Step 2c: Add the error block to the blockchain
    add_block_to_chain(encrypted_data)

else:
    # Step 3: Handle the valid data case
    private_key = enter_private_key()

    # Step 3a: Hash and encrypt the validated data
    hashed_data = hash_data(decrypted_data)
    encrypted_data = encrypt_data(hashed_data, private_key)

    # Step 3b: Add the validated block to the blockchain
    add_block_to_chain(encrypted_data)

    # Step 3c: Send the encrypted data to siZ
    send_data_to_siZ(siZ, encrypted_data)
```

However, *siY* must describe the error and then hash, encrypt, and add data to the chain when an error is detected. Otherwise, *siY* sends data to *siZ* using its public key, as described in *Algorithm 2*. *siZ* then decrypts the data with its private key, displays the information for each institution, and adds a block to the chain, as illustrated in *Algorithm 3*.

Algorithm 3: siZ process

```
Start
  siZ enters the private key
  If the private key is invalid then
    Show error message
    Go back to Step 2 (re-enter private key)
  Else
    Decrypt the received data
    For each institution (siX, siY, siZ) do
      Display gradeID
      Display job title
      Display institution name
      Display single-spine grade
      Display level
      Display amount
      Add the block to the blockchain
    End For
  End If
End
```

```
# Code Representation of Algorithm 3
def siZ_process(private_key):
  # Step 1: Ensure a valid private key is entered
  while True:
    if is_valid_key(private_key):
      break
    else:
      print("Invalid key. Please try again.")
      private_key = enter_private_key()

  # Step 2: Decrypt the received data
  decrypted_data = decrypt_data(data, private_key)

  # Step 3: Display information and add block for each institution
  for institution in [siX, siY, siZ]:
    display_information(institution, decrypted_data)
    add_block_to_chain(decrypted_data)
```

We used field-level encryption for sensitive data fields (e.g., salary grade, salary amount) instead of encrypting entire data blocks to balance security and system efficiency. This approach preserves the ability to perform efficient queries and searches on non-sensitive metadata fields without compromising data confidentiality.

5.2. Smart contracts for automated execution

This way, each institution would have a copy of the same ledger containing all employee payroll data. The ledger is immutable, meaning no one can alter or delete the data once it is recorded. The ledger would also

be verifiable, meaning that each institution could check the validity of the data by tracing its history on the blockchain. Smart contracts ensure data are processed according to predefined rules and conditions.

The novelty of these algorithms lies in using blockchains and smart contracts to create a distributed, automated, and secure system for managing payroll data among state institutions. This system offers several advantages over traditional methods, such as

(a) Reducing the need for intermediaries, such as third-party agencies, may charge employee fees and change grade levels to introduce errors in the salary system.

(b) Increasing the transparency and accountability of the transactions, as each state institution can access and verify the data on the blockchain.

(c) Enforcing compliance with the agreed terms and conditions, as the smart contracts execute automatically and irreversibly when specific criteria are met.

These algorithms represent a novel and innovative way of applying blockchain and smart contract technology to real-world problems. However, they may also face specific challenges, such as

(a) *Scalability*: As transactions and data blocks increase, blockchains may become slower or more congested.

(b) *Security*: The private keys and smart contracts may be vulnerable to hacking or malicious attacks.

(c) *Regulation*: The legal and regulatory frameworks for blockchain and innovative contract technology may vary across jurisdictions or sectors.

5.3. Error handling and data approval process

State agencies in Ghana manage sensitive data that require error handling and data approval processes to maintain the integrity and accuracy of the information exchanged. Secure data-exchange algorithms involving blockchain and smart contracts are critical for maintaining the reliability of the payroll system. Algorithm 1 initiates data block creation, encryption, and transmission to *siY*, with possible errors such as incorrect data entry or transmission issues. Algorithm 2 in *siY* focuses on verifying the decryption private key's validity, displaying an invalid error message, and prompting an error and description entry if decryption issues occur. The errors are re-encrypted with *siY*'s private key and added to the blockchain for transparency. Data underwent checks and additional information additions at each stage before passing to the next institution, ensuring accuracy and accountability. These error handling and data approval processes are vital for a secure, reliable, and transparent data exchange system that promotes timely error identification, data integrity maintenance, and accountable record-keeping among state entities.

In our system, Delete Transactions do not erase salary-grade records but instead logically mark records as "deleted." This ensures compliance with auditability and legal data retention requirements. Archival mechanisms are implemented to safeguard historical records even after deletion operations.

5.4. Evaluation metrics

The key metrics include throughput (transaction processing speed), latency (confirmation time), scalability (system capacity to handle growth), and security (protection against unauthorised access). These metrics are crucial for assessing the efficiency and reliability of a blockchain in various transaction volumes. In this section, we provide descriptions of the key metrics used to evaluate the performance of

the algorithms proposed in this study, which aim to measure the quality and efficiency of blockchain and smart contract systems.

(a) *Throughput*: Number of transactions or data blocks processed per unit time.

(b) *Latency*: The time it takes for a transaction or data block to be confirmed and added to the blockchain.

(c) *Scalability*: The ability of the system to handle increasing workloads without compromising performance or security.

(d) *Security*: The level of protection against unauthorised access, modification, or deletion of data on blockchain or smart contracts.

(e) *Reliability*: the degree to which the system operates correctly and consistently according to predefined rules and conditions.

(f) *Usability*: The system's ease of use and understanding for end users and developers.

Using these metrics, we discuss some aspects of the performance of the algorithms.

(a) The algorithms had low throughput and high latency, involving multiple steps of encryption, decryption, hashing, verification, and error handling for each data block. This could slow down the process and increase system costs.

(b) The algorithms seem to have high security and reliability, as they use private and public keys to ensure the authenticity and integrity of the data and smart contracts to comply with the agreed terms and conditions. This could reduce the risk of fraud, errors, and disputes among state institutions.

(c) These algorithms seem to have low scalability, as they may face challenges in handling large volumes of data or transactions or adapting to changing requirements or regulations. This can limit the applicability and functionality of the system in different scenarios and contexts.

(d) These algorithms seem to have low usability, requiring technical knowledge and skills to create, execute, and verify data blocks and smart contracts. This can make it difficult for non-experts or non-developers to use or understand.

The system was tested using real data from state *institution X*, resulting in a high approval rate for salary grades and a low decline rate due to standard rate discrepancies. The hyperledger calliper tool evaluated the system's performance, showing high throughput and low latency across various transaction rates.

5.5. Web portal and implementation

The system was developed using Hyperledger Fabric Blockchain technology for the backend and Python (Django) for the front end. The system satisfies the main objective of this project, from catering to the creation of the grade structure to ensure that the employee salary grade created at the entry-level is being approved by *siY* and paid precisely by *siZ*. The system was developed such that only *siX* has the authority to create and modify the details of a salary-grade structure. *SiY* reads all grades that are pending, approved, and declined. They alone have mandates to approve or decline their salary grades.

All declined salary grades were attached with notes on the issue. *siZ* has read-only permission, which allows them to view only the approved salary grades and to pay accordingly. Hyperledger Fabric supports the entire system in recording all transactions on the network, thus tracking who performs what on the network.

The corresponding user and timestamp were recorded for every grade structure created, approved, or declined. To use the system effectively, all users must have an account. Users who do not want to install an account with the system cannot use the online application simply by inputting the URL. The application

allows users to select the domain where the authentication should occur. Figure 7 shows *the application interface of siX* when entering a grade in the system. Figure 8 shows *the interface of siX* when approving a salary grade in the system. The web portal contains blockchain algorithms at the backend.

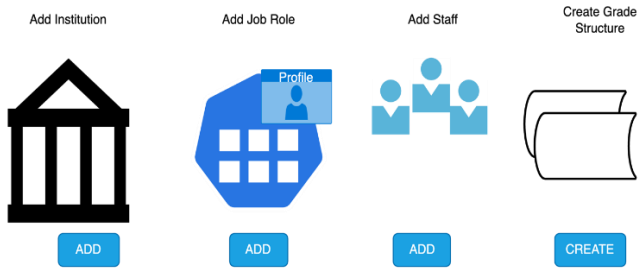

Figure 7. The application interface of the *siX*.

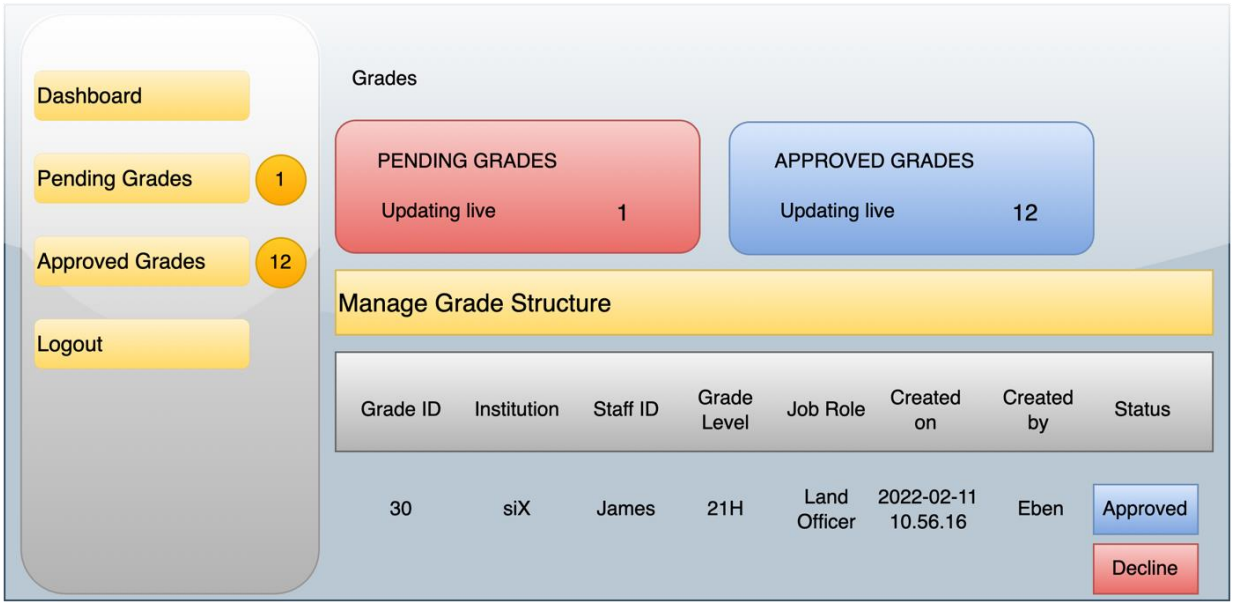

Figure 8. The application interface of the *siY*.

6. Results and discussion

6.1. Results and performance analysis

The proposed system was tested and evaluated using information and data from the *siX*. To verify how effective the approach works, two hundred and thirty-two (232) job titles were assigned to employees at the *agency S* with different salary grades. After entering the salary grade at *siX*, it goes through the system for approval from *siY*. Thirty-four (34) salary grades declined because they were below the standard rate assigned to the job titles.

One hundred and ninety-eight (198) were approved for payment, and the system paid the same number. Figure 9 shows a graphical representation of the experiment. The consistent count between Approved and Payment demonstrates the system's reliability in executing payments. Figure 9 shows the system's efficiency, effectiveness, and ability to accurately handle large volumes of data in salary management using blockchain technology.

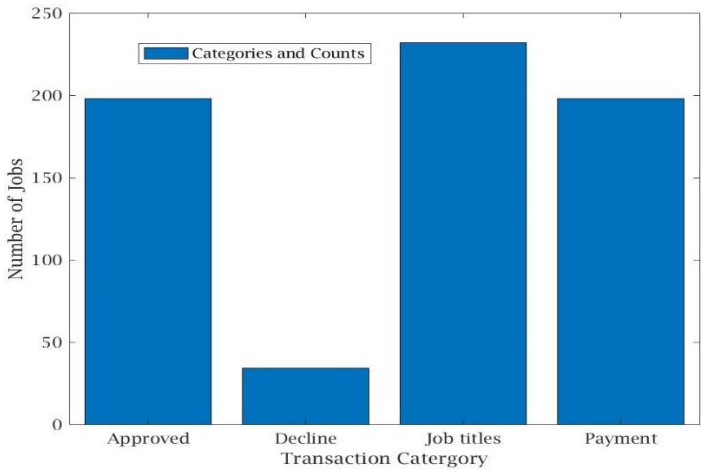

Figure 9. Job title assignment and approval.

Figure 10 shows an experiment on the blockchain application developed to verify the approach's effectiveness. Two hundred and seventy-three (273) job titles were assigned to employees at *agency K* with different grades. After entry, 13 declined because they were below or above the standard rate for their job titles. Two hundred sixty (260) were approved for payment, and the same number was forwarded to *siZ* for payment on the system. The job distribution presented in Figure 10 underscores the efficiency and effectiveness of the approval and payment processes in the salary management system, with minimal decline and a high rate of approved and processed job titles.

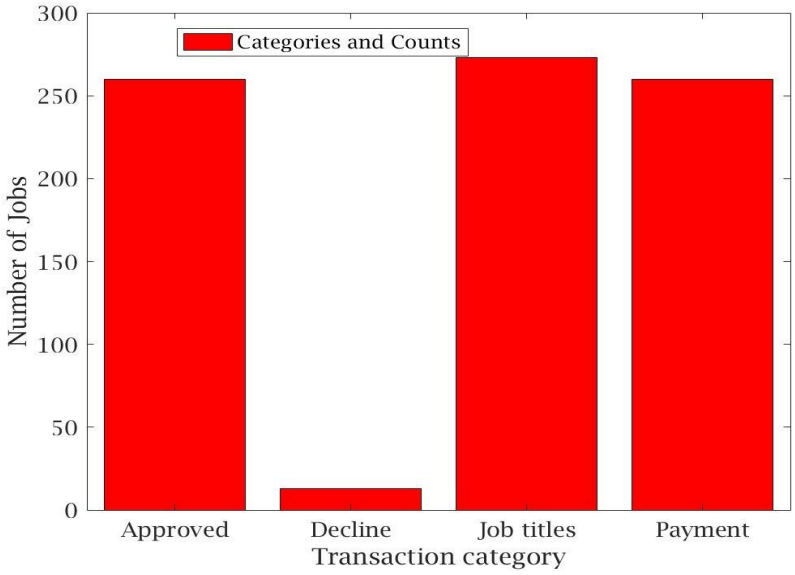

Figure 10. Job verification in *Agency K*.

Figure 11 illustrates the system performance metrics (throughput, latency, and scalability) across different transaction loads (100, 150, 200, 250, and 300 transactions). Throughput indicates the capacity of the system to process transactions. The throughput significantly increased as the load increased from 100 to 300. The significant increase in throughput suggests that the system's architecture and Hyperledger Fabric platform can efficiently handle increasing transaction volumes, ensuring timely processing of payroll transactions and maintaining operational efficiency. Latency measures delays in processing transactions. At an initial load of 100 transactions, the latency is moderate. As the load

increases to 150, 200, and 250 transactions, the latency also increases, reflecting the additional time required to process more transactions. However, at 300 transactions, there was a slight decrease in latency, which could indicate optimisations in the system's performance or reaching a threshold where the system's handling capacity improves. Scalability assesses the system's ability to expand and manage an increased workload. The scalability metric remained consistent at lower loads (100 and 150 transactions) and gradually increased as the transaction load reached 200, 250, and 300 transactions. A gradual increase in transaction loads indicates that the system is designed to scale effectively and maintain performance, even as the number of transactions increases.

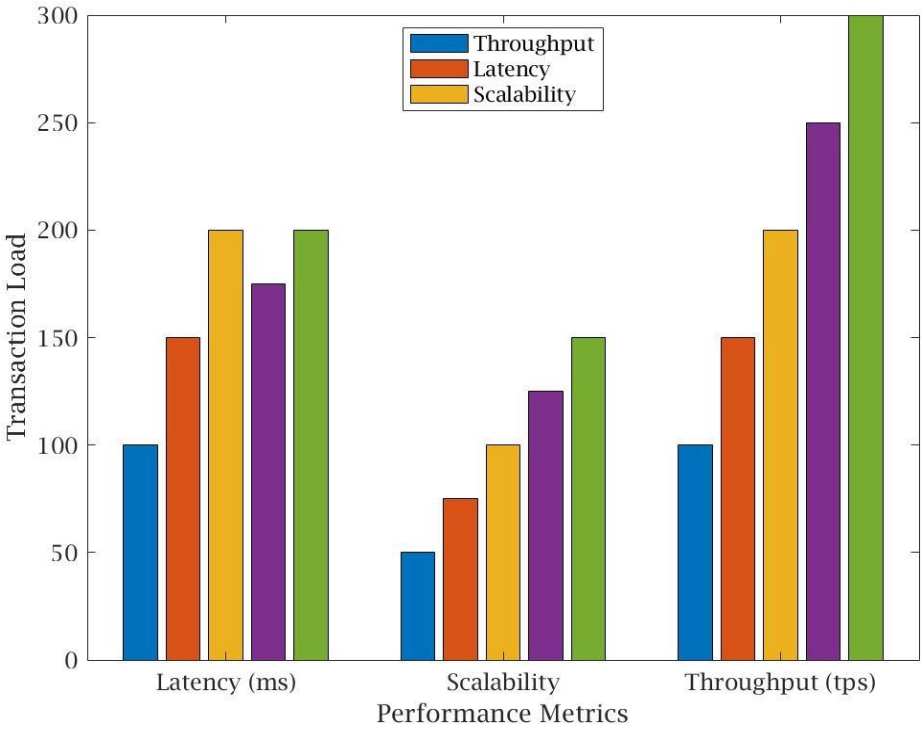

Figure 11. Impact of transaction rates on the blockchain.

Figure 12 illustrates the relationship between the transaction rate (tps), throughput (tps), and latency (ms) within a blockchain-based salary-grade structure management system. A linear correlation exists between the transaction rate and throughput. As the transaction rate increases from 1 to 10 tps, the throughput increases proportionally, indicating that the system can effectively handle higher loads.

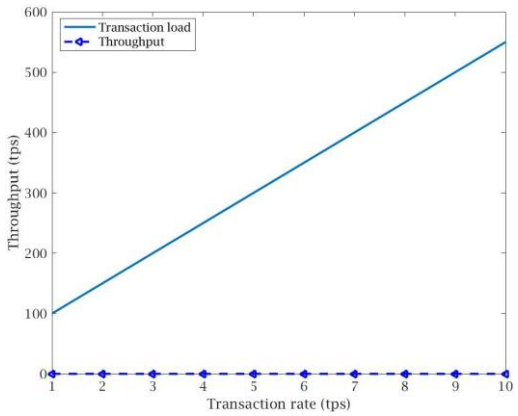

Figure 12. Impact of transaction rates on the blockchain.

The throughput started at 100 tps for a transaction rate of 1 tps and increased consistently, reaching approximately 500 tps at a transaction rate of 10 tps. This linear relationship demonstrates the scalability and efficiency of the system for transaction processing. The latency remained constant at 0 ms across all transaction rates. This stability suggests that the system is optimised to effectively manage latency, ensuring that the transaction-processing time does not increase with higher transaction rates. Maintaining low and stable latency is critical for the real-time performance of the system, particularly in a salary-management context where the timely processing of transactions is crucial. The system demonstrated high performance by maintaining high throughput and stable latency. This efficiency indicates the robust design and architecture of the blockchain-based system, leveraging Hyperledger Fabric to optimise transaction processing.

6.2. Findings and discussions

The findings of implementing blockchain technology in salary management are significant, particularly in improving transparency and security. Blockchain technology has effectively mitigated stakeholders' trust. The system built using the Hyperledger Fabric platform demonstrated a 100% increase in transparency in the payroll system, ensuring that all changes made by stakeholders in salary management were securely tracked and recorded [11], [16]. Encryption and decryption capabilities of blockchain technology are crucial for securing data. The RSA asymmetric encryption technique was utilised to encrypt and secure the blockchain system, which provided data reliability and enhanced the security of the salary-management system [16]. This ensures that sensitive information, such as employee salary grades and transactions, is protected from unauthorised access and tampering [12], [17].

The algorithms designed for this system facilitate secure data exchange among state institutions using hashing, encryption with private keys, and verification with public keys. This process ensures that the records of salary grades and transactions are transparent and tamper-proof [18], [20]. Smart contracts in this system automate the execution of transactions, ensuring compliance with predefined rules and conditions without intermediaries [19], [21].

Moreover, the system architecture, which includes a network of one channel and one smart contract (gradeContract), allows for robust and efficient management of the grade structure. The client application interacts with the Hyperledger Fabric system to query the ledger, send new transactions, and receive updates on changes to the ledger [22,23]. This architecture supports the system's goal of improving transparency and reducing the wage bill in public service payrolls. Integrating a RESTful API designed with ExpressJs and SQLite3 facilitates seamless communication between the Hyperledger Fabric network and client applications. This API ensures secure and robust interactions, enabling users to query the ledger, initiate transactions, and receive updates without dealing with network complexities [11], [16].

The performance of our system was critically analysed, focusing on throughput and latency. The results confirm that the proposed solution meets the intended purpose. Blockchain technology's encryption and decryption capabilities are crucial in securing data, and smart contracts ensure automated and secure transaction execution. The system's architecture supported efficient salary grade structure management, and the API integration facilitated robust and safe interactions with the blockchain network.

Implementing Hyperledger Fabric in this context highlights its potential for broader industry-wide cooperation. The secure, transparent, and efficient management of salary structures demonstrated in this study can be leveraged across various sectors to enhance business operations' efficiency and

performance. The blockchain-based approach can facilitate industry-wide data security and transparency standards, promoting the collaborative development of high-performance and reliable blockchain platforms. Overall, using blockchain technology and smart contracts to manage salary grade structures offers a promising solution to the issues of trust and transparency in salary management within public organisations. The system developed in this study demonstrates the potential of blockchain technology to improve data security, transparency, and efficiency in public service payroll management [16,17].

Although salary amounts are encrypted, metadata such as *gradeID* and *jobTitle* could potentially leak sensitive information through frequency analysis. Future system versions will incorporate pseudonymisation, random padding, or hashing techniques to protect against such inference attacks. Furthermore, blockchain data privacy challenges and solutions, including zero-knowledge proofs (ZKP) and regulatory audit key mechanisms, have been discussed based on recent studies [24–27].

7. Conclusion and future research directions

7.1. Conclusion

In this paper, we implemented a secure and transparent system (*i.e.*, salary grade structure management) using Hyperledger Fabric, a leading blockchain technology tool. Our findings demonstrate significant improvements in transparency and security of payroll management. The system mitigates stakeholder trust issues by ensuring that all the changes are tracked and recorded. Specifically, using Hyperledger Fabric increased the transparency of the payroll system by 100%, with encryption techniques, such as RSA asymmetric encryption, enhancing data security. This ensures that sensitive information such as employee salary grades and transactions is protected from unauthorised access and tampering. The designed algorithms facilitated secure data exchange among state institutions, ensuring that records of salary grades and transactions were transparent and tamper-proof. Smart contracts automate transaction execution, ensuring compliance with predefined rules without intermediaries. The system architecture, which includes a network of one channel and one smart contract (gradeContract), proved robust and efficient in managing the grade structure. The client application's interaction with the Hyperledger Fabric system enabled secure querying, transaction initiation, and ledger updates.

Furthermore, integrating a RESTful API with ExpressJs and SQLite3 facilitated seamless and secure communication between the Hyperledger Fabric network and client applications. This study contributes a unique, context-driven blockchain solution to public sector payroll transparency challenges. By integrating RSA encryption with Hyperledger Fabric's permissioned blockchain architecture and validating the system using real institutional data, we demonstrate the feasibility of applying blockchain technology beyond cryptocurrency markets into government administrative systems.

7.2. Future research directions

Future research should focus on several key areas to further enhance the system:

(1) *Scalability:* To test the algorithm's efficacy and the system implemented in this study, methods to scale the system to handle larger datasets and more complex salary structures without compromising performance and security will advance the use of blockchain and smart contracts in managing salaries worldwide. While the system was tested with a relatively small set of users, Hyperledger Fabric's

architecture is inherently designed to handle scalability requirements for larger and more complex deployments. Scalability can be achieved by adding more peers, orderers, and channels, allowing for load balancing and parallel processing of transactions. Hyperledger Fabric's modular approach separates transaction ordering, endorsement, and validation, providing flexibility to scale horizontally. We plan to perform extensive scalability evaluations in future work by simulating larger transaction volumes using Hyperledger Caliper, a blockchain benchmark tool. This will allow us to measure performance metrics such as throughput, latency, and resource consumption at scale, ensuring the system's robustness for deployment in real-world, large-scale salary management scenarios.

(2) *Integration with Other Technologies:* Explore integrating other emerging technologies, such as artificial intelligence and machine learning, to enhance data analysis and decision-making processes within the blockchain framework.

(3) *Interoperability:* Develop solutions to ensure interoperability between blockchain platforms, enabling seamless data exchange and cooperation across various blockchain networks.

(4) *Enhanced Privacy Mechanisms:* Research on advanced cryptographic techniques to improve the privacy and security of sensitive information within the blockchain system.

(5) *Real-world Applications:* Conduct pilot studies in various industries to evaluate the practical applications and benefits of the proposed system in real-world settings.

(6) *Simulating large-scale environments:* To further strengthen the scalability and performance of the system, future research will focus on simulating larger and more dynamic user environments. We plan to apply tools like Hyperledger Caliper to benchmark the system under varied transaction loads, ensuring high throughput and low latency performance at scale. Additionally, we aim to investigate the integration of machine learning techniques for predictive analytics in salary management and explore enhanced cryptographic techniques for greater privacy protection across blockchain-based systems.

(7) *Pseudonymization and frequency analysis defences:* Currently, metadata fields, such as *gradeID* and *jobTitle*, are not obfuscated in the system. While sensitive salary values are encrypted, metadata could still expose information through frequency analysis or inference attacks. Future system versions will incorporate pseudonymization techniques and defences against frequency analysis to enhance privacy protection, such as adding random padding, hashing sensitive metadata fields, or introducing dummy records. These methods will reduce the risk of adversaries deducing confidential information from metadata patterns, strengthening the system's overall data privacy and resilience against side-channel attacks.

Authors' contribution

Conceptualization, E.E.M.; methodology, E.E.M.; software, E.E.M.; validation, R.K.A., J.N.W.; formal analysis, R.K.A.; investigation, J.N.W.; resources, K.S.A.M.; data curation, E.E.M.; writing—original draft preparation, E.E.M.; writing—review and editing, K.S.A.M.; visualization, R.K.A., J.N.W.; supervision, K.S.A.M.; project administration, K.S.A.M. All authors have read and agreed to the published version of the manuscript.

Conflicts of interests

The authors declare no conflict of interest.

References

[1] Subramanian KR. Technology and transformation in communication. *J. Adv. Res. Electr. Electron. Eng.* 2018, 5(8):01–13.

[2] Li G, Xue J, Li N, Ivanov D. Blockchain-supported business model design, supply chain resilience, and firm performance. *Transp. Res. Part E* 2022, 163:102773.

[3] Wu H, Yao Q, Liu Z, Huang B, Zhuang Y, *et al.* Blockchain for finance: A survey. *IET blockchain* 2024, 4(2):101–123.

[4] Turk Ž, Klinc R. Potentials of blockchain technology for construction management. *Procedia Eng.* 2017, 196:638–645.

[5] Lemieux VL. Trusting records: is blockchain technology the answer? *Rec. Manage. J.* 2016, 26(2):110–139.

[6] Anjum A, Sporny M, Sill A. Blockchain standards for compliance and trust. *IEEE Cloud Comput.* 2017, 4(4):84–90.

[7] Du M, Ma X, Zhang Z, Wang X, Chen Q. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, October 05–08, 2017, pp. 2567–2572.

[8] Chentouf FZ, Bouchkaren S. Security and privacy in smart city: a secure e-voting system based on blockchain. *Int. J. Electr. Comput. Eng.* 2023, 13(2):1848.

[9] Rasheed S, Louca S. Blockchain-based implementation of national census as a supplementary instrument for enhanced transparency, accountability, privacy, and security. *Futur. Internet* 2024, 16(1):24.

[10] George JT. Hyperledger Fabric. In *Introducing Blockchain Applications: Understand and Develop Blockchain Applications Through Distributed Systems*, 1st ed. Berkeley: Apress, 2022. pp. 125–147.

[11] Pooja V, Vijay K, Raghavi V, Bhuvaneswaran B, Manohar E. Electronic health records & data management using Hyperledger Fabric in blockchain. In *2023 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, January 23–25, 2023, pp. 1–4.

[12] Kumar N, Dakshayini M. Secure sharing of health data using Hyperledger Fabric based on blockchain technology. In *2020 International Conference on Mainstreaming Block Chain Implementation (ICOMBI)*, Bengaluru, India, February 21–22, 2020, pp. 1–5.

[13] Wątorek M, Drożdż S, Kwapień J, Minati L, Oświęcimka P, *et al*. Multiscale characteristics of the emerging global cryptocurrency market. *Phys. Rep.* 2021, 901:1–82.

[14] Sharma, J. Blockchain technology adoption in financial services: Opportunities and challenges. In *Revolutionizing Financial Services and Markets Through FinTech and Blockchain*, 7th ed. USA: IGI Global, 2023.pp. 99–117.

[15] Sikeridis D, Bidram A, Devetsikiotis M, Reno MJ. A blockchain-based mechanism for secure data exchange in smart grid protection systems. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, USA, January 10–13, 2018, pp. 1–6.

[16] Thantharate P, Thantharate A. ZeroTrustBlock: enhancing security, privacy, and interoperability of sensitive data through ZeroTrust permissioned blockchain. *Big Data Cogn. Comput.* 2023, 7(4):165.

[17] Craß S, Lackner A, Begić N, Mirhosseini SAM, Kirchmayr N. Collaborative administration of role-based access control in smart contracts. In *2022 4th Conference on Blockchain Research &*

Applications for Innovative Networks and Services (BRAINS), Paris, France, September 27–30, 2022, pp. 87–94.

[18] Khan MY, Zuhairi MF, Ali T, Alghamdi T, Marmolejo-Saucedo JA. An extended access control model for permissioned blockchain frameworks. *Wireless Netw.* 2019, 26(7):4943–4954.

[19] Ponsam JG, Duvvuri S, Roy S. Electronic healthcare management system using blockchain technology. In *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, Kollam, India, August 10–11, 2023, pp. 869–877.

[20] Breitwieser H, Leszak M. A distributed transaction processing protocol based on majority consensus. In *Proceedings of the First ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, Ottawa, Canada, August 18–20, 1982, pp. 224–237.

[21] Alkhamisi AO, Alboraei F. Privacy-aware decentralized and scalable access control management for IoT environment. *J. King Abdulaziz Univ. Comput. Inf. Technol. Sci.* 2019, 8(1):71–84.

[22] Noh S, Shin SU, Rhee KH. PyRos: a state channel-based access control system for a public blockchain network. *Secur. Commun. Netw.* 2020, 2020(1):8891183.

[23] Vijayaraj A, Prahalathan P, Reddy VG, S D, Reddy D V A. Legal documentation system using blockchain and interplanetary file system. In *2024 International Conference on Computing and Data Science (ICCDS)*, Chennai, India, April 26–27, 2024, pp. 1–6.

[24] Wang X, Garg S, Lin H, Piran MJ, Hu J, *et al*. Enabling secure authentication in industrial IoT with transfer learning empowered blockchain. *IEEE Trans. Ind. Inf.* 2021, 17(11):7725–7733.

[25] Wang X, Garg S, Lin H, Hu J, Kaddoum G, *et al*. Toward accurate anomaly detection in industrial Internet of Things using hierarchical federated learning. *IEEE Internet Things J.* 2021, 9(10):7110–7119.

[26] Wang X, Garg S, Lin H, Hu J, Kaddoum G, *et al*. A secure data aggregation strategy in edge computing and blockchain-empowered Internet of Things. *IEEE Internet Things J.* 2020, 9(16):14237–14246.

[27] Wang X, Garg S, Lin H, Hu J, Kaddoum G, *et al*. Heterogeneous blockchain and AI-driven hierarchical trust evaluation for 5G-enabled intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* 2021, 24(2):2074–2083.