IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 July 2023, accepted 11 July 2023, date of publication 14 July 2023, date of current version 21 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3295696

RESEARCH ARTICLE

A Blockchain-Based Framework for Supervision of Livelihood Issues: Proof of Concept With Optimized Consensus

JINYUE XU[1], CAIJIAN HUA[1,2], AND YAN ZHANG[1,2]

[1]School of Computer Science and Engineering, Sichuan University of Science and Engineering, Yibin 644000, China
[2]Key Laboratory of Higher Education of Sichuan Province for Enterprise Informationalization and Internet of Things, Sichuan University of Science and Engineering, Yibin 644000, China

Corresponding author: Caijian Hua (hwacj@suse.edu.cn)

This work was supported in part by the Innovation Fund of Postgraduate of Sichuan University of Science and Engineering under Grant Y2022174; in part by the Sichuan University of Science and Engineering for Talent Introduction Project, under Grant 2017RCL59; and in part by the Key Laboratory of Higher Education of Sichuan Province for Enterprise Informationalization and Internet of Things Plan Project under Grant 2022WYY01.

ABSTRACT Supervision of Livelihood Issues (SLI) refers to the supervision and participation of the general public in government actions and public affairs, with a particular focus on issues closely related to people's lives. However, current regulatory systems often suffers from information imbalances, low authenticity, and credibility. To address these challenges, this study proposes a blockchain-based SLI framework that leverages the decentralized, tamper-proof, traceable, and transparent nature of blockchain technology to provide a reliable platform for SLI. Furthermore, we designed an information data structure on the blockchain and presented the information flow process. We developed smart contracts that utilize automation capability to automatically aggregate, analyze, and publish warnings based on on-chain data, enabling the evaluation of relevant departments' performance. Moreover, in the context of SLI applications, we propose an optimized consensus algorithm called the SLI-PBFT. This algorithm is based on the PBFT consensus protocol and incorporates dynamic scalability mechanisms, scoring models, and simplified consistency protocols to enhance the consensus efficiency of SLI applications. Finally, the experiments were conducted. We developed an SLI prototype system and compared and analyzed the performance of SLI-PBFT with other PBFT-based consensus algorithms within the SLI prototype system. The experimental results demonstrated the feasibility of the SLI framework. Moreover, SLI-PBFT exhibits significant improvements in throughput and reduced latency compared to other algorithms in the context of SLI applications. In addition, they possess strong fault tolerances and security capabilities. This study provides theoretical and practical guidance for the SLI in real-world scenarios.

INDEX TERMS Supervision of livelihood issues, blockchain, smart contract, consensus algorithm, SLI-PBFT, secure and trustworthy, data sharing, prototype system.

I. INTRODUCTION

Livelihood issues encompass a range of challenges that directly impact the lives of citizens, spanning various domains such as agriculture, education, and healthcare. Ensuring citizens'well-being in terms of life and property safety, physical and mental health, and basic rights holds immense significance. Effectively addressing livelihood issues requires collaborative efforts from the government and diverse sectors of society, wherein SLI assumes an indispensable role as a pivotal mechanism for citizen engagement in social governance.

SLI refers to the supervision of government administrative organs, public service institutions, and social issues by citizens through various forms and means in multiple fields to safeguard the basic livelihood rights and interests of the

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

people, promote the government's fulfillment of duties in accordance with the law, and enhance its transparency and credibility [1].

Traditional SLI methods often have inherent limitations. For instance, there are challenges in acquiring information, typically relying on personal experiences, media reports, and investigations conducted by non-governmental organizations. However, these sources of information may not be comprehensive or reliable, and the inadequacy or inaccuracy of the information can potentially undermine the effectiveness of supervision. Moreover, the timeliness and dynamic nature of supervisory information are lacking. Traditional SLI relies primarily on feedback from individual cases and post-incident handling. Consequently, supervision outcomes may not be timely enough to promptly identify issues and facilitate timely interventions. Additionally, it fails to provide a comprehensive understanding of long-term trends and changes in livelihood issues [2].

With the rapid development and popularization of information technology, digitization has become inevitable in various fields. Digital SLI is of great significance to people's well-being, as it can not only ensure that the immediate interests and legitimate rights of the people are not infringed, but also improve the government's governance capacity and efficiency, and provide reference for government decision-making [3], [4]. The application of big data technology in public SLI has also become a measure by which local governments can promote digital transformation [1], [4].

Compared to traditional supervision methods, big-databased SLI leverages machine learning and data mining techniques to analyze and model large-scale data, facilitating the identification of data trends and anomalies. Moreover, big data technologies enable real-time processing of massive data streams, beyond static datasets. Through real-time data processing and analysis, the most up-to-date information can be promptly obtained, allowing for timely issue detection and corresponding measures. Big data-based SLI issues typically adopt distributed computing and storage architectures, such as Hadoop and Spark. The distributed architecture divides the data into smaller chunks and processes them in parallel, thereby accelerating data processing speed.

Big data technology has broken the time and space constraints of traditional supervision methods, expanded the spatial scope of supervision and enhanced the accuracy of issue detection. Some government departments have established big data SLI platforms, which have achieved certain results. However, despite its significance, there are some problems in applying big data technology to the field of SLI [5]:

- Data authenticity and credibility
  Big data contain a large amount of information related to SLI, including reporting data and processing results. It typically relies on centralized data warehouses and control systems, which carry the risk of single-point failure. Furthermore, these data storage systems are relatively more susceptible to the risks of human interference and tampering, leading to decreased transparency and credibility of data in SLI work. This is highly likely to result in trust issues among various stakeholders, such as ordinary citizens, social organizations, and relevant departments, ultimately leading to conflict.
- Data Source and Security
  Owing to the involvement of multiple environments, multiple data sources, and complex data flow, the data sources in big data can have different formats, structures, and identification methods, making data tracing complicated. In addition, the risks of unauthorized access and data leakage pose a threat to data security. Attackers can exploit various technical means to infiltrate data storage and transmission processes, leading to technical vulnerabilities in big data governance.

The utilization of new technologies to improve or overcome the drawbacks of SLI of big data is an important issue that requires careful consideration. Blockchain technology is a promising solution that can provide higher data credibility, stronger security, and better traceability [5].

Blockchain is a distributed ledger technology that links data in the form of blocks, forming a chain-like sequence [6]. It employs cryptographic techniques to ensure data security, with each block containing a hash value related to the preceding block. The hash value serves as a unique identifier obtained through encryption operations on the data from the previous block. Any tampering with data on the blockchain results in a change in the hash value, leading to rejection by other nodes in the network [7]. Thus, blockchain possesses immutability, making it more secure and reliable than big data technologies. Because of the provision of a publicly verifiable distributed ledger, each node in the blockchain network possesses a complete copy of the ledger. The consensus mechanism ensures that network nodes reach an agreement on transactions and data and participants within the blockchain ecosystem can constantly monitor and trace the source and transaction history of data [8], [9]. The execution of smart contracts [10] is automated based on predefined rules and conditions. When specific conditions are met, the corresponding operations are executed automatically, thereby reducing human intervention and improving the execution efficiency. This transparency and trustworthiness enhance the trust of all parties involved in the data on the blockchain, resolving the issues of information authenticity and credibility often encountered in big data technologies.

Furthermore, the PBFT consensus algorithm [11] is commonly used in consortium blockchains and has certain advantages in addressing Byzantine fault tolerance issues. However, it still has some shortcomings.

- First, in the PBFT algorithm, when a node joins or leaves, it triggers a view change operation that may cause a temporary network delay. The network needs to be reconfigured to achieve a new consensus, which undoubtedly reduces the efficiency of the consensus.

- Second, the PBFT algorithm selects the primary node in that order. Although this is fair, it may allow malicious nodes to continuously become primary nodes, resulting in wastage of network resources. While malicious primary nodes can be identified and overthrown by other nodes, frequent changes in primary nodes increase system overhead and reduce consensus efficiency.
- Third, the PBFT algorithm incurs high performance overhead. To achieve fault tolerance, multiple rounds of message exchange are required, where each node needs to broadcast messages to other nodes and wait for a sufficient number of responses before proceeding. This increases the communication and computational burden on the system, resulting in longer delays and a lower throughput.

In response to the shortcomings of the PBFT consensus algorithm mentioned above, this study proposes the introduction of a dynamic scalability mechanism that allows nodes to join and leave the blockchain network dynamically without requiring network reconfiguration or restart. Additionally, a scoring model was introduced to calculate the score for each node based on its performance. This model allows high-scoring and reliable nodes to serve as primary nodes, thereby increasing the stability of primary nodes and reducing the frequency of primary node changes. Furthermore, the consensus protocol was optimized. Using the scoring model, the selection of primary nodes becomes more stable, thereby significantly reducing the probability of Byzantine faults. This allows for the simplification of the message exchange process in the PBFT consensus algorithm, reducing the computational burden on the network and consequently lowering latency while increasing throughput.

Therefore, the main objective of this research is to propose a conceptual framework supported by blockchain for SLI and to optimize the PBFT consensus algorithm in SLI business scenarios. The aim was to achieve more efficient, secure, transparent, and real-time supervision and management, providing a more reliable and trustworthy SLI service to safeguard people's livelihood rights. The contributions of this study are as follows:

1) This study proposes a blockchain-based SLI conceptual framework, that prevents data tampering (such as case processing results and assessment data) and enhances the transparency and credibility of SLI.
2) Smart contracts were designed in this study for the aggregation, analysis, and alerting of on-chain data, as well as the evaluation and assessment conducted by relevant departments in the context of SLI.
3) In the application context of SLI, this study presents an efficient consensus mechanism called SLI-PBFT based on an optimized PBFT consensus algorithm. It ensures security and stability while reducing the complexity of network communication and improving consensus efficiency.
4) A prototype system for the SLI development was implemented using the Hyperledger Fabric (HLF) framework in a laboratory environment.

The remaining sections of this paper are organized as follows: Section II discusses the research related to blockchain and SLI. Section III introduces the conceptual framework of SLI. Section IV presents the SLI-PBFT consensus algorithm. Section V describes the development of a blockchain-based SLI prototype system. Section VI introduces the experimental testing of the SLI-PBFT consensus algorithm. Finally, Section VII summarizes this study and provides an outlook on future research.

II. RELATED RESEARCH

In this section, we discuss SLI, the application of blockchain technology in this field, and the research and improvement directions of consensus algorithms pursued by researchers.

A. SLI

Traditional SLI methods include media supervision [12], accountability systems [13], and social organization supervision [14]. These methods rely on manual inspections and reporting. Supervisors spend a lot of time and energy on field investigations and information collection. The subjectivity of supervisors may also affect the objectivity of the supervision results. Simultaneously, the objects of supervision may also conceal problems through various means, thus avoiding their discover by supervisors [15].

To address the drawbacks of traditional monitoring methods, an increasing number of researchers are focusing on big data monitoring methods. Compared to traditional monitoring methods, big data monitoring methods have many advantages [2], [4]. In terms of SLI, big data monitoring methods have been widely used in fields such as food safety, agriculture, and environmental protection. For example, Zhang et al. [16] designed an unqualified rate evaluation index for food safety issues, and used big data processing and regional grid management to assist food safety supervision. Liu et al. [17] used big data technology to collect, process, and analyze data in real-time with the aim of strengthening public supervision of the security situation of agricultural products networks. Jin and Jin [18] proposed a technology that uses big data to monitor environmental pollution in ecological economic zones. A model was constructed for the quality of online monitoring data on ecological environmental pollution.

Previous research has mainly focused on proposing various monitoring methods, but lacks consideration for maintaining the security and reliability of data. Big data technology typically stores the data obtained in SLI applications in a centralized system and fully trusts a single data management entity, such as the government. However, in the field of SLI applications, as many participants are involved, such as government functional departments, discipline inspection and supervision departments, citizens, and social organizations,

such a single centralized system is more prone to problems. For example, the emergence of data security issues may cause the public to question the quality of the government work data. To address these issues, a transparent, tamper-proof, and traceable SLI system must be developed.

B. BLOCKCHAIN AND ITS APPLICATIONS IN THE FIELD OF SLI

The history of blockchain technology can be divided into three stages: 1.0, 2.0, and 3.0+ [19]. The Bitcoin network was launched in 2008, marking the beginning of the blockchain 1.0 era [20], during which various cryptocurrencies were developed. With the successful implementation of smart contracts on the blockchain, Ethereum ushered in the blockchain 2.0 era in 2014 [21]. However, Ethereum's performance and high costs significantly limit the development of highly customized enterprise applications. The 3.0+ era was created to describe enterprise-customized blockchain technology. HLF is a representative project, and HLF has been identified as one of the best blockchain solutions for building applications [22], [23]. Meanwhile, HLF, as an enterprise-grade consortium blockchain framework, possesses high scalability, robust privacy and permission controls, pluggable consensus mechanisms, flexible smart contract support, and focus on enterprise-level requirements. Therefore, it is employed in most consortium blockchain applications that require authorization and permission to participate [24].

Blockchain applications have been explored by researchers in the field of SLI. Tao et al. [25] applied blockchain technology to the field of food supervision and proposed a hierarchical multi-domain blockchain (HMDBC) network structure. This structure supports the collaborative governance, regulation, and correction of malicious nodes among regional nodes. They also developed a prototype system to address the issue of inadequate regulation in the food supervision system. Peng et al. [26] applied blockchain in the rice supply chain in the agricultural sector and proposed the multi-blockchain rice refined supervision model (MBRRSM) framework, which allows for detailed supervision of rice quality and safety at various stages of the supply chain. It provides secure data transmission and usage at different privacy levels. They also built a prototype system based on the MBRRSM framework and validated its credibility. Li et al. [27] focused on the transportation sector and introduced a blockchain-based Intelligent Transportation System (ITS) architecture that was compatible with traditional ITS infrastructure and services. They developed a prototype system through conceptual verification to better supervise the privacy protection of vehicles and user information in the transportation field. Zhao et al. [28] and Zhong et al. [29] proposed blockchain-based environmental monitoring models. They utilized the blockchain's distributed storage mode to ensure secure data sharing in monitoring activities. They optimized environmental monitoring models from the perspectives of consensus algorithms and smart contracts to supervise and control the behavior of relevant parties in terms of pollutant emissions and environmental protection while preventing data tampering. They conducted case studies to validate the feasibility of the proposed frameworks and implemented blockchain-based prototypes.

These examples demonstrate that many researchers are utilizing blockchain technology's transparent, trustworthy, and efficient supervision mechanisms in various livelihood sectors. The widespread application of blockchain in the field of SLI has brought about numerous positive impacts and has showcased its immense potential and prospects. Consequently, many researchers have integrated blockchain and artificial intelligence technologies into the domain of livelihood supervision. This comprehensive application further enhances the effectiveness and feasibility of supervision, leading to more innovations and advantages in SLI.

For example, in the healthcare sector, Ameen et al. [30] explored the combination of artificial intelligence (AI) and blockchain technology in the context of the Internet of Medical Things (IoMT). In doing so, IoMT has the potential to become the foundation of future healthcare systems, where all medical devices are connected to the Internet and operate under the supervision of medical experts. AI technologies, such as machine learning and deep learning excel in making accurate predictions and face challenges in terms of security and privacy. To address this, the distributed nature of blockchain and technologies such as the interplanetary file system (IPFS) are often utilized to store data, whereas cryptographic techniques of blockchain are employed to design identity verification and key protocols for controlling access permissions, ensuring the integrity, security, privacy, and transparency of data. Mohammed et al. [31] discussed the application of machine learning and blockchain technology to a dynamic IoMT system. They proposed a federated learning-based blockchain-enabled task scheduling (FL-BETS) framework aimed at addressing data fraud in distributed IoMT systems. This study provides an innovative solution that combines federated learning and blockchain technology, making a significant contribution to digital healthcare applications in dynamic IoMT systems, particularly focusing on the application of blockchain in fraud analysis and data privacy. Experimental results demonstrate that FL-BETS outperforms existing machine learning and blockchain mechanisms in fraud analysis, data verification,and energy and latency constraints in healthcare applications. Lakhan et al. [32] acknowledged the increasing use of (ar)ial intelligence AIbased-based digital intelligent healthcare solutions. However, these solutions often focus solely on predicting and classifying different diseases, while neglecting performance handling and data privacy issues. Therefore, they constructed an Energy-Efficient Distributed Federated Learning Offloading and Scheduling (EDFOS) platform within a blockchain-based network to address the latency, privacy, and security concerns of healthcare applications on the platform. The study also presented offloading and scheduling schemes that minimize energy consumption

to ensure blockchain security on all computing nodes and meet the quality of service requirements of applications, optimizing power consumption in remote healthcare applications. Experimental validation confirmed that the EDFOS platform is an effective solution for addressing power consumption and data privacy issues in healthcare applications.

It is evident that many researchers have begun exploring the application of blockchain technology in the field of SLI. Various frameworks and solutions have been proposed to ensure data security and privacy. These studies typically focused on specific domains. However, the uniqueness of blockchain technology lies in its broad applicability across various fields, thus possessing potential for cross-domain applications. Therefore, in this study, we utilized blockchain technology to collect and store integrated supervisory complaints, whistleblowers, and advisory information from various aspects of people's livelihoods. By recording this information on the blockchain, relevant departments can not only share information in real-time and collaborate to resolve cases swiftly but also avoid cumbersome procedures and communication processes, thereby enhancing efficiency and response speed. Furthermore, the public can track and verify the progress of case handling, thereby achieving a more just, transparent, and efficient mechanism to address livelihood issues.

C. CONSENSUS ALGORITHM

A consensus mechanism is crucial in blockchain technology because it affects the processing capability and security of the blockchain. Existing blockchain consensus algorithms can be broadly divided into PoX and Byzantine fault-tolerant (BFT) algorithms. PoX algorithms mainly include PoW [20], PoS [33], and others, which are generally used in public blockchains to achieve consensus through token investment. Private chains primarily employ the distributed consensus algorithm RAFT [34] for consensus; however, RAFT cannot solve the Byzantine fault tolerance problem. In the context of consortium chains, where there is no need to introduce token incentive mechanisms on the chain, most consortium chains currently use Byzantine fault-tolerant protocols. In 1999, Castro and Liskov [11] proposed the PBFT consensus algorithm, which significantly reduced the computational waste exhibited by classical blockchain consensus algorithms. Currently, the PBFT consensus algorithm is being widely applied to consortium chains. However, as blockchains become more popular and evolve, PBFT faces challenges such as poor flexibility, the possibility of malicious nodes being repeatedly elected as primary nodes, and high communication complexity. In response to these issues, many researchers have conducted research and made improvements to the PBFT consensus algorithm. Lei et al. [35] proposed a reputation-based RBFT consensus algorithm based on PBFT. This algorithm establishes a reputation value for each node, which reduces the participation of nodes with low reputation, thereby limiting the behavior of malicious nodes. Xu and Wang [36] proposed an improved consensus algorithm called VS-PBFT based on fuzzy sets. It partitions the network nodes using a consistent hashing algorithm, and selects a local primary node from each partition to participate in the global consensus. Nodes can dynamically join and leave, greatly increasing the network flexibility. Zhong et al. [37] proposed a secure and efficient blockchain distributed consensus algorithm called ST-PBFT based on consortium chains. They introduced a grouping method based on the consistency hash principle to divide consensus nodes into groups, enabling parallel processing of IP transactions within transaction consensus groups, thereby reducing the communication complexity and improving the throughput of the algorithm. They also proposed a node reputation evaluation model to prevent Byzantine nodes from being repeatedly elected as the primary nodes. Liu et al. [38] proposed a practical Byzantine fault-tolerant algorithm called P-PBFT based on PBFT, grouping, and credit voting. They optimized the consistency protocol in the original algorithm and partitioned the network nodes into different consensus sets based on the response speed, thereby alleviating the network communication complexity. They also introduced a credit model and voting mechanism to dynamically update the user status based on the behavior of consensus nodes, assessing user reliability, and selecting nodes with high credit as primary nodes.

In summary, previous optimizations of the PBFT consensus algorithm have mainly focused on enhancing the flexibility of the blockchain network, improving the primary node selection, and reducing the communication complexity. Therefore, considering the background of SLI in this research, the following optimization ideas can be explored beyond the PBFT. Considering the wide range of users involved in SLI, dynamic participation or withdrawal is likely to occur. Therefore, the dynamic capability of the PBFT consensus algorithm can be extended. For supervisory entities in SLI, false reporting may exist, so it may be necessary to introduce a scoring model to specifically restrict false reporting and prevent malicious nodes from frequently becoming primary nodes or even expelling them directly from the network. Furthermore, the probability of the occurrence of malicious nodes is reduced owing to the introduction of the scoring model. Consequently, unnecessary communication processes can be omitted, thereby reducing the communication complexity. With a lower probability of malicious node occurrence, unnecessary communication processes can be omitted, thereby reducing the communication complexity.

III. CONCEPTUAL FRAMEWORK OF SLI BASED ON BLOCKCHAIN

In this section, we introduce the conceptual framework of SLI based on blockchain technology. We discuss the architecture of the blockchain system, the roles and responsibilities of participants, and the process of information flow within the chain. Finally, we design relevant smart contracts to support the SLI framework.

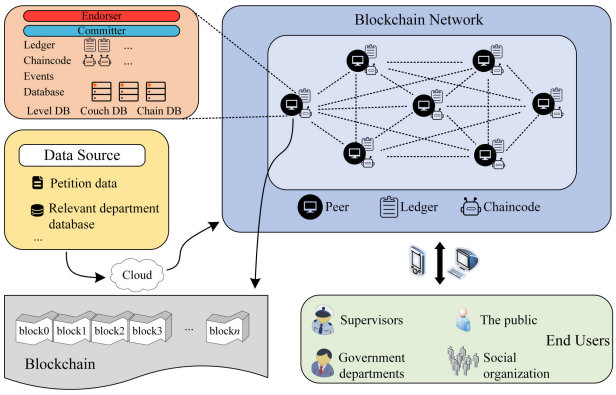

FIGURE 1. Overall conceptual framework.

TABLE 1. Responsibility of each participant.

Participants' responsibility	Governments	Supervisors	Citizens	Organizations
Participation in blockchain	✓	✓	✓	✓
Uploading data to blockchain	✓	✓	✓	✓
Deploy smart contracts	✓	✓		
View on-chain data	✓	✓	✓	✓
Maintain network security and stability	✓	✓		✓
Analyze data		✓		

A. OVERALL CONCEPTUAL FRAMEWORK

As shown in Fig. 1, this study proposes a conceptual framework for implementing SLI using consortium chain technology. A consortium chain, as a specific type of blockchain technology, offers the benefit of allowing only authorized and permitted participants to join the chain, thereby providing higher credibility and reliability. The framework primarily consists of three components: data sources, blockchain participants, and blockchain network. The data sources encompass two main categories. The first category is petition data, which refers to information expressed by citizens and social organizations to relevant departments or institutions through means such as letters, phone calls, and online platforms, regarding their supervisory opinions, complaints, whistleblowing, and other related matters. The second category comprises databases maintained by relevant departments, which collect data through their own investigations, monitoring, statistics, and other means. The roles and responsibilities of participants are outlined in Table 1. The main focus of this study is how to utilize the blockchain network to achieve effective SLI and enhance the monitoring capabilities of citizens and social organizations towards government institutions, public service organizations, and social problems.

This framework aims to address citizens' needs to supervise government institutions, public service organizations, and societal issues in different forms and ways. Specifically, the framework can be applied to the supervision and resolution of the following issues. The first is the feedback and resolution of livelihood issues. Blockchain can provide a trustworthy data platform for citizens and social organizations, allowing participants to submit problems related to livelihood areas through forms. Examples include environmental protection, market supervision, and healthcare. For instance, citizens can report instances of factories discharging wastewater indiscriminately, cases of inaccurate weighing in the market, and instances of doctors in hospitals violating regulations by accepting bribes. All of these issues can be recorded on the blockchain. Government institutions will address these matters on-site and publish the outcomes in the chain for supervision by the public. Second, this framework can be used to supervise the quality and efficiency of public services. Citizens can record their experiences and evaluations of blockchain and assess and monitor public services. Third, the framework can supervise the administrative efficiency and integrity of government departments. Citizens can monitor the operations of government institutions through blockchain networks, thereby promoting governmental transparency and clean governance.

In the SLI framework, to accommodate the diverse forms and means of citizen supervision of government agencies, public service organizations, and social issues, different roles are granted varying permissions and functionalities. This can be achieved using the following approaches:

- Multi-channel architecture
  Multiple channels can be created for different types of supervisory activities. Each channel has an independent ledger and can define its own members and related smart contracts, thus restricting access and participation in transactions and consensus processes only to specific organizational nodes. For example, a channel dedicated to citizen complaints and reports can be created, as well as separate channels for government agencies' supervision and social issue monitoring. Each channel has its unique participants and chaincodes that enable data separation and isolation.
- Participant permission management
  Different types of participants were assigned different permissions. Citizens can join citizen complaints and report channels as regular participants, whereas government agencies and supervisory bodies can join the corresponding channels as privileged participants for effective communication and collaboration. This ensures that participants can access and participate only in channels and data relevant to their supervisory roles, thus enabling permission control.
- Customized chaincode
  The chaincode can be customized for each channel based on specific supervision requirements. The chaincode can define relevant functionalities and rules according to specific supervision scenarios to meet the diverse needs of citizens and relevant organizations. For example, in the citizen complaint channel, chaincode logic can be defined as receiving, processing, and tracking the complaint information.

The core objective of the framework is to meet the diverse needs of citizens in supervising government agencies, public service organizations, and social issues through different forms and means. Specifically, the framework can be applied to feedback and resolution of livelihood issues, quality and efficiency supervision of public services, and administrative efficiency and integrity supervision of government departments. To accommodate different supervision requirements, the framework employs strategies such as multi-channel architecture, participant permission management, customized chaincode, and transaction record auditing. Through this framework, citizens and relevant organizations can engage in supervision activities on a blockchain platform with high trustworthiness and transparency, thereby enhancing supervision capabilities and promoting transparency and effectiveness in governance.

B. HLF AND ON-CHAIN INFORMATION FLOW

1) HLF

This study employed the HLF architecture, an open-source system that is both modular and scalable, to establish and operate permissioned blockchain networks. The architecture employs two authorization mechanisms: the namely Membership Service Provider (MSP) and the Fabric-Certificate Authority (Fabric-CA) [39]. MSP, a modular component of HLF, offers identity services to all the participants. Fabric-CA manages membership by performing operations such as the registration, addition, and revocation of member certificates. Within the SLI conceptual framework based on blockchain, registration requests to join the network must be sent by all participants via the client. E-Cert and T-Cert were then issued to participants as two types of certificates. Once admitted to the blockchain network, participants can access and process information within the bounds of their authorized scope.

2) ON-CHAIN INFORMATION FLOW

The following aspects were considered when designing the on-chain information data structure in the SLI framework:

1) Data types
   Determine the types of data to be recorded in the SLI framework, including complaint data, investigation and processing data, and supervision and evaluation data. Complaint data depends on specific SLI requirements and issues, such as problems in different domains (market supervision, ecological protection, transportation, corruption, rural revitalization, education and healthcare, housing and urban development). The investigation and processing data included the investigation and processing results of the relevant agencies. Supervision and evaluation data consisted of assessments by citizens, social organizations, and other relevant government departments.
2) Data fields
   Defining the fields and their relationships in the smart contract to ensure that the recorded data contain necessary information. In the contract, the Go language structure is used to define the structure of data fields, whereas mapping is used to implement key-value storage for quick data retrieval and lookup. Data fields include timestamps, locations, problem descriptions, participant information, and tamper-proof signatures of the supervised data.
3) Data association
   This is a crucial step in ensuring the correlation between the on-chain data. For example, when dealing with complaint data, it is necessary to associate it with the processing results of the relevant departments to form a complete historical record of the problem. Therefore, the SLI framework uses hash values as unique identifiers for each piece of data to establish the associations. Each data block contains the unique hash value of the previous block, and by checking the hash values of the nodes on the chain, the integrity and order of the data can be verified, establishing associations between data elements.

To ensure the consistency, integrity, traceability, and privacy of the data, the following measures were taken:

1) Consensus mechanism
   Blockchain ensures the consistency and integrity of data through a distributed consensus mechanism. The SLI framework adopts the SLI-PBFT consensus algorithm, which requires participants in the network to reach consensus on the state of the data. Only blocks that have undergone consensus can be accepted and added to the chain, preventing malicious nodes from tampering with data and ensuring data consistency. Specific details of the SLI-PBFT consensus algorithm are discussed in the next section.
2) Cryptographic hash function
   Cryptographic hash functions are used in the SLI framework to ensure the data integrity. The hash function converts the data into a unique hash value of fixed length. Any modification to the data results in a different hash value, which can be detected by other nodes. The SLI framework utilizes the SHA-256 hash function, which accepts an input of arbitrary length and generates a 256-bit hash value. The integrity of the data can be verified by storing the hash value of the data in blocks, thereby ensuring its authenticity and completeness.
3) Digital signatures
   Digital signatures were used to verify the authenticity and integrity of the data. By employing the ECDSA digital signature algorithm, the sender can digitally sign the data using their private key, generating a unique signature value. The recipient can verify the integrity of the data and identity of the sender using the sender's public key. Because the signature value is unique and associated with the sender's private key, anyone can verify the validity of the signature using a publicly shared public key, thus confirming that the

data originated from the sender. The traceability of a data source enhances its credibility.

4) Timestamp
In the SLI framework, the main node responsible for the consensus process generates new blocks and adds timestamps. Upon receiving these blocks, other nodes verify the validity of the timestamps and accept them, thereby ensuring time consistency throughout the network. The use of a timestamp mechanism associates each transaction or data record with a specific point in time, ensuring the chronological order of data and providing the traceability of historical data.
5) Privacy channels
The HLF chain used in the SLI framework supports privacy channels, allowing for the creation of different and independent blockchain channels within the network. These channels can be used to share and interact with private data among specific participants without revealing or granting access to other participants, thereby preventing unauthorized access and tampering.
6) Private datasets
The HLF chain used in the SLI framework enables smart contracts to maintain private data sets. These data sets isolate certain data among specific participants, without being written into the global state of the blockchain and made public to all participants in the network. Data storage for these datasets typically resides in distributed databases, such as CouchDB or LevelDB.

The combination of these mechanisms enables the SLI framework to address potential vulnerabilities effectively and protect data security. Distributed consensus, hash functions, digital signatures, timestamps, privacy channels, private data sets, and other mechanisms collectively ensure the integrity, privacy, and security of data on the blockchain.

In the blockchain-based SLI framework, there are four main types of transactions:

- Information on policies, regulations, and announcements issued (<Type=0, P_{id}, P_s, T_g, σ_g>).
- Information on reports and complaints (<Type=1, C_{cf}, C_{id}, C_s, T_c, σ_p>).
- Information on progress and results (<Type=2, C_{cf}, C_{id}, C_p, C_r, T_r, σ_g, σ_s>).
- Information on assessment and notification (<Type=3, E, E_{id}, E_d, E_e, T_e, σ_g, σ_s, σ_o>).

The meanings of the symbols for the on-chain information can be found in Table 2.

A schematic diagram of the information flow on the blockchain is illustrated in Fig. 2, and the specific details are as follows [40]:

1) Transfer the information data on livelihood issues or the relevant department's database to the client, and the client converts the information data into on-chain transactions.

TABLE 2. The meaning of symbols in on-chain information.

Text	Describe
P_{id}	Unique identifier for each policy
P_s	The latest status of the policy, which can be "in effect" or "invalidated"
T_g	The time when the policy is published.
σ_g	The signature of the government administrative department that can be publicly verified.
C_{cf}	The field in which the case is located, such as transportation, medical care, education
C_{id}	A unique identifier for each case
C_s	The processing status of the case
T_c	The time when the report, complaint or feedback is initiated
σ_p	The signature of the citizen or organization that can be publicly verified
C_p	The processing progress of the case.
C_r	The processing results and effects, including responses to the complainant, punishment or rectification of the complained target
T_r	The time when the report, complaint or feedback is initiated
σ_s	The signature of the government administrative department that can be publicly verified
E	Information on notification and praise or criticism and warning
E_{id}	A unique identifier for each event
E_d	The government administrative department, supervisory department, or other corrupt department that is assessed or notified
E_e	Evidence supporting the occurrence of the event, such as assessment reports, notification letters, supervisory investigation results
T_e	The start time of the assessment or notification
σ_o	The signature of the social organizations that can be publicly verified

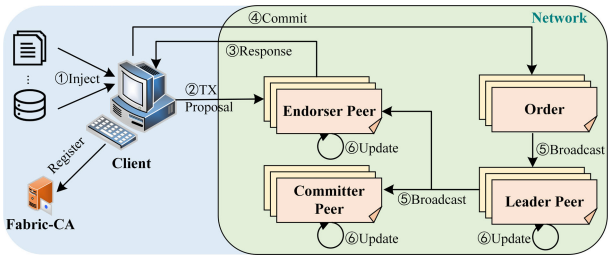

FIGURE 2. Information flow on the chain.

2) Client submits a transaction
The client submits this signed transaction to the endorsement node in the Fabric network using tools such as Fabric SDK or CLI.
3) Transaction verification
The endorsement node runs a simulation to verify the validity of the transaction proposal. If the following conditions are met:
   - The information format is correct

- It has never been submitted before
- The MSP signature used is valid
- The submitter is authorized to perform the proposed operation on the channel (which can be used to partition the state of the blockchain network)

Then, the endorsement node inputs the transaction proposal as a parameter to generate the transaction result, including the response value, read set, and write set. Finally, the endorsement response is sent back to the client.

4) Transaction packaging
After the client collects enough endorsement responses, the endorsement transaction proposal response and transaction are submitted to the orderer node. The orderer node is responsible for packaging the transactions into blocks and sorting them.
5) Block broadcasting
After a block is packaged, it is broadcast to the peer nodes of the entire network. These nodes verify each transaction included in the block, including the transaction signature, endorsement, endorsement policy, etc., to ensure that each transaction complies with the rules on the chain.
6) Blockchain state update
If all transactions pass verification, the transaction is approved and written to the blockchain. Each peer node records the transaction on its own ledger and synchronizes between each block to keep all node states in sync.

C. SMART CONTRACTS

Smart contracts, also known as chaincodes, are a type of program code that can automatically execute specific tasks to assist participants in a blockchain network, such as performing complex logic and recording data [10]. Once smart contracts are deployed on the blockchain, their code is permanently stored on the blockchain and can be accessed and executed by the nodes in the network.

The automation features of smart contracts in blockchain are achieved in the following ways:

- Conditions and Triggers
Smart contracts can set conditions and triggers for achieving automated functionality. Conditions are predefined specific criteria, whereas triggers are mechanisms that activate contract execution when conditions are met. When conditions are satisfied, triggers automatically activate the contract, enabling it to perform corresponding operations.
- Event Listening
Smart contracts can listen to specific events occurring in the blockchain and automatically execute corresponding actions based on the event occurrences. By subscribing to and listening to events, contracts can automatically respond to specific state changes or interactions, thus enabling automated functionality.
- Data-driven
Smart contracts can interact with data on the blockchain and automatically execute the corresponding actions based on data changes. Contracts can read, write, and process data stored on the blockchain, perform operations, and analyze the data according to predefined logic and algorithms, thus achieving automated behavior.

In this study, three types of smart contracts were designed: control, supervision, and evaluation contracts, as shown in Fig. 3.

The control contract is responsible for regulating access to resources, enforcing policies and rules, and determining whether users are authorized to interact with or contribute information to the blockchain network. The execution process of the control contract is as follows: The user sends a request to the control contract, such as accessing resources, performing operations, or submitting information. The contract then determines whether to allow the user to perform the requested operation based on the user's permission level or role. If the user is authorized to perform an operation, the contract verifies and executes the operation or rejects it based on predefined policies and rules. If the request involves accessing a data source, the contract simultaneously checks the availability of the data source and the user's permissions to determine whether the user can access the data source.

The supervision contract includes functions such as classified information collection, data analysis and summarization, and system alerts. For the classified information collection function, it is crucial to determine the types and attributes of the different types of classified information that need to be collected. For example, in the field of livelihood issues, classification can be performed according to several major areas such as market supervision, ecological protection, transportation, corruption, rural revitalization, education and healthcare, housing, and urban development. Each category of information is assigned to the corresponding data structures, including fields and types, to ensure data consistency and accuracy. Users submit supervised content and corresponding category information to the contract through an interactive interface. For the data analysis and summarization functions, when the smart contract obtains relevant supervised data, it utilizes aggregation functions, sorting, filtering, and calculations to process and analyze the data. Regarding the system alert function, the data analysis and summarization results are sent to the respective blockchain participants or applications through an event notification mechanism. This is achieved through custom events that trigger corresponding alerts and notifications based on predefined conditions.

For the evaluation contract, each department can submitt the data through an interactive interface and store it on the blockchain in a structured manner to ensure transparency and immutability of the data. The contract calculates and summarizes the performance data based on predefined evaluation indicators and weights. In this case, a weighted average algorithm was used to calculate the overall score of each department based on the importance of different

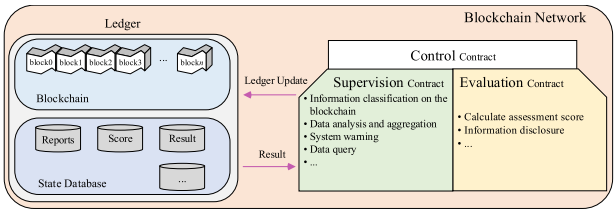

FIGURE 3. Smart contracts for SLI based on blockchain.

indicators. The calculation results serve as the outputs of the evaluation contract. In addition, the evaluation contract provides an interface for users to query and view the performance evaluation results of each department. Users can select specific departments and evaluation periods using an the interactive interface to monitor the corresponding performance scores and evaluation reports. This enhances transparency and encourages participation and supervision in performance evaluation. In the evaluation contract, thresholds and conditions were also set. When the performance score of a department reaches or exceeding a specific value, the contract can automatically trigger a notification mechanism to inform the relevant parties of the performance results. This is achieved by setting custom events and conditions.

Logic and detailed description of the smart contracts are shown in Table 3.

IV. SLI-PBFT: PBFT CONSENSUS ALGORITHM OPTIMIZATION UNDER SLI APPLICATION

All nodes in a blockchain network should maintain identical copies of the blockchain data. A consensus mechanism is needed to ensure the consistency of blockchain data among all nodes. Various consensus mechanisms have been designed for different systems and purposes, such as PoW [20], PoS [33], and PBFT [11]. Suggestions for selecting consensus algorithms for existing consensus protocols applied in blockchain and different blockchain application scenarios can be found in [41] and [42], among others. In this study, the PBFT consensus algorithm was optimized in an SLI application framework based on blockchain.

A. PBFT

PBFT is a general solution for guaranteeing the consistency of distributed systems and Byzantine fault nodes, mainly addressing the problem of malicious nodes sending incorrect information to other nodes and disrupting the normal operation of the system. The PBFT algorithm provides a fault tolerance of $(n-1)/3$ while ensuring the security and reliability of the system, allowing up to 1/3 of the nodes to fail. PBFT requires nodes to maintain a shared state and all nodes remain consistent; hence, it requires a consensus protocol, view conversion protocol, and checkpoint protocol [11].

The consistency protocol is the core protocol that enables the PBFT algorithm to achieve consensus by dividing consensus nodes into primary and backup nodes. Only one primary node is responsible for sorting requests from the clients. The backup nodes execute the requests in the order determined by the primary node, ensuring that the order of executing requests on each node is consistent, thereby guaranteeing the consistency of the block content. The execution process of the PBFT algorithm consistency protocol is illustrated in Fig. 4.

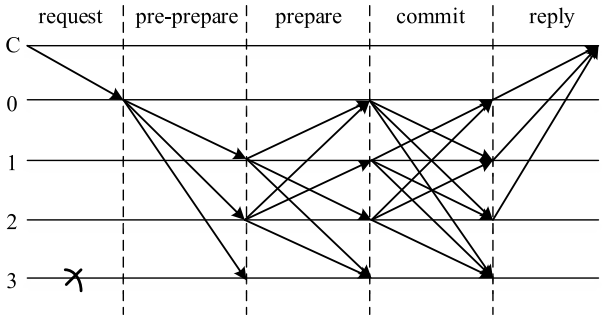

FIGURE 4. Execution process of PBFT consensus algorithm.

The PBFT algorithm selects the primary nodes in order, which may allow malicious nodes to become primary nodes consecutively, and waste network resources. Although malicious primary nodes can be recognized and overturned by other nodes, frequent changes in primary nodes will increase the system overhead and reduce the consensus efficiency. In the consensus phase, the consistency protocol requires three rounds of broadcast communication in all cases to achieve security in asynchronous mode, which leads to a high network communication overhead. Moreover, the PBFT algorithm has poor dynamism because nodes cannot dynamically join or leave the cluster, making the algorithm less flexible in specific applications.

B. SLI-PBFT

The SLI-PBFT algorithm is based on PBFT and introduces dynamic scalability mechanisms and a scoring model, along with optimization of the consensus protocol. These components are key parts of the SLI-PBFT consensus algorithm that address the issues present in PBFT to some extent. Next, we discuss the working characteristics and processes of the SLI-PBFT algorithm.

1) DYNAMIC SCALABILITY MECHANISM

The main function of the dynamic mechanism is to allow nodes to join and exit the consensus network dynamically without the need to restart the blockchain network, effectively improving the flexibility of the consensus.

- Node Dynamic Joining Mechanism

1) AddNode phase
   If a new node wants to join the network cluster to participate in subsequent consensus phases, it should first send its digital certificate, public key, request to join the cluster, and timestamp to all consensus nodes in the cluster and then initiate the AddNode message to apply for joining.

TABLE 3. The logic and detailed description of the smart contract.

Contract business logic	Contract method	Describe
Information classification on the blockchain	InfoClsUpChain()	Define data classification standards and upload data
Data analysis and aggregation	DataAnalysis()	Statistically analyze on-chain data
System warning	SetThreshold()	Monitor relevant cases and automatically trigger alerts when the threshold is reached
Data query	Query()	Query relevant on-chain data
Calculate Assessment Score	CalculateScore()	Assess and evaluate the work of relevant departments
Information disclosure	UpdateInformation()	Update public information and push it automatically

2) AgreeAdd phase
When a consensus node receives an AddNode request initiated by a new node, it performs simple legality checks on the request message and checks for duplicate requests to ensure that the request is legitimate. If the identity and legitimacy of the new node are verified, the consensus node sends an authentication message, AgreeAdd, to the new node to join the network cluster. When the new node receives $2f+1$authentication messages, it is allowed to join the cluster.
3) RequestsSyn phase
After the new node completes joining, it actively sends a data synchronization request RequestSyn and broadcasts it to other nodes. The consensus node then sends the current node th list and status information to the new node to facilitate data synchronization.
4) UpdateNet phase:
The main node publishes the UpdataNet information to all nodes in the cluster. When all consensus nodes receive the message, they update the total number of nodes N and the view v within the blockchain cluster, completing the process of adding a new node. When the view and total number of nodes are updated, the consensus nodes provide feedback to the main node. When the main node receives $2f+1$ messages, indicating that the network has completed the integration of the new node, the dynamic node addition consensus behavior is completed.

The dynamic node-addition process is shown in Fig. 5, where "New" represents a new node.

- Node Dynamic Exit Mechanism

1) DelNode phase
When a node actively requests an exit, it broadcasts a DelNode message to other nodes, including information such as ID and exit timestamp. This enables the other nodes to recognize and process requests.
2) AgreeDel phase
Upon receiving the Del-request message, other nodes calculate the view and total number of nodes after deleting the node. They then broadcast their agreement to delete the node to other nodes in the blockchain network, thus avoiding node misjudgment and data inconsistency owing to message delay or other reasons.

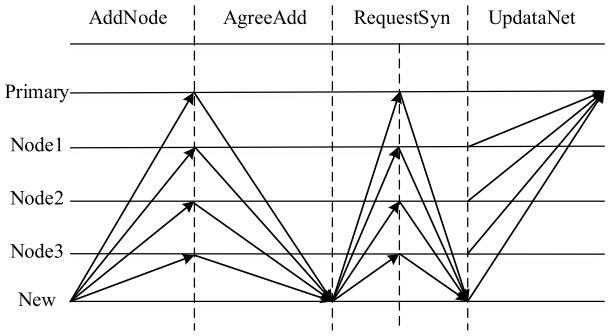

FIGURE 5. Dynamic joining proceduce of nodes.

If $2f+1$ AgreeDel messages are collected, the node is deleted.
3) Exit phase
When the node is ready to exit, it must send its status information and data to other nodes so that they can reallocate tasks and data.
4) UpdataNet phase
After a node exits, the main node sends an UpdataNet message. Upon receiving the UpdataNet message, all nodes in the network update the total number of nodes and views in the blockchain network to complete the node deletion process. Other nodes must reallocate tasks and data to ensure the normal operation of the system.

The node dynamic exit process is illustrated in Fig. 6, where Del represents the node requesting an exit.

2) SCORING MODEL

In the PBFT algorithm, the primary node is determined in order, which may result in an abnormal node being selected as the primary node, thereby affecting the security and stability of the system. In the improved SLI-PBFT consensus algorithm, a scoring model was introduced to evaluate the state of nodes, classify consensus nodes in the network, and select the primary node based on the scoring model. Nodes with higher scores are considered to be more secure and stable, and have a lower probability of executing the view conversion protocol, thereby increasing the security of primary node selection and improving the efficiency of

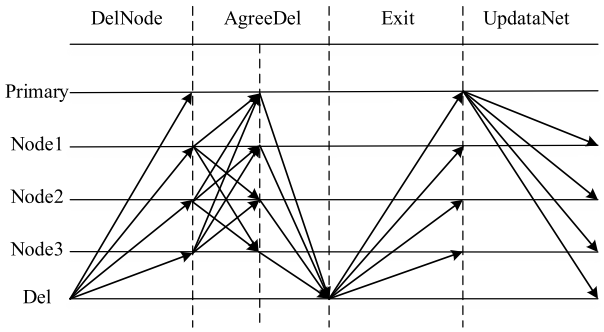

FIGURE 6. Dynamic exit procedure of node.

consensus. By continuously executing the consensus mechanism, nodes that successfully reach consensus accumulate scores and higher scores indicate greater security.

The scoring model proposed in this study sets the node score range to [0, 100], with an initial value of R_{init}. Based on the score size, the nodes participating in the SLI-PBFT consensus are divided into four categories: malicious nodes, ordinary nodes, candidate nodes, and priority nodes (with ranges of [0, R_l), [R_l, R_m), [R_m, R_h), and [R_h, 100], respectively). Malicious nodes do not participate in consensus, do not receive consensus results and are removed. Ordinary nodes do not participate in the consensus process and only receive the consensus result. Candidate nodes can only serve as a sub-node in the consensus and cannot serve as the primary node, whereas priority nodes can become the primary node and participate in the consensus process.

To evaluate the scores of nodes, this study refers to reference [43] and combines SLI improvement methods to measure the scores of nodes based on metrics, such as node activity, historical influence, and historical category.

- *Definition* 1
  Activity is used to evaluate whether the node actively participates in the consensus, and is represented as

$$A(i) = \left[\frac{xR_p}{R} + y\left(1 - \frac{d_{ij}}{d_{max}}\right)\right] * 100 \quad (1)$$

In Equation (1), R represents the total number of consensus processes, R_p represents the number of times node i participates in the consensus, d_{ij} represents the delay of the j-th transaction of node i, and d_{max} represents the maximum allowed delay for transactions. If it exceeds the maximum delay, it indicates a transaction failure. x and y are the coefficients used to balance the weight, where $x+y=1$.

- *Definition* 2
  The historical influence is used to evaluate the impact of a node's historical transaction behavior on the score. The historical influence of node i can be expressed as

$$H(i) = \frac{100}{n}\sum_{i=1}^{m}\mu_i \quad (2)$$

In Equation (2), n represents the total number of transactions in the system, and m represents the number of transactions completed by node i. μ_i represents the participation flag of node i in the transaction: when the transaction is successful, its value is 1 and when the node behaves abnormally, its value becomes -1. With this design, both the promotion effect of successfully completed transactions on nodes and the adverse impact of abnormal node behavior are considered.

- *Definition* 3
  The node historical category was used to evaluate the contribution of nodes in the different categories. If a node is often classified as a priority node, it is considered reliable and has a higher score, which increases the probability that the node will becoming the main node, and reduces the probability of malicious nodes becoming the main node.

$$P(i) = \frac{100}{n}\sum_{i=1}^{R}\lambda_i \quad (3)$$

In Equation (3), R represents the total number of consensus processes, and λ_i represents the node type: if it is a priority node, then the value is 0.5; if it is a candidate node, then the value is 0.4; if it is an ordinary node, then the value is 0.1.

- *Definition* 4
  The formula for calculating the final score of a node is as follows:

$$C(i) = \frac{1}{3 \times 2^z}[\alpha A(i) + \beta H(i) + \gamma P(i)] \quad (4)$$

In Equation (4), α is the weight of the activity, β is the weight of the historical influence, and γ is the weight of the node category value, where $\alpha + \beta + \gamma = 1$. z represents the number of times the node acted maliciously. The scoring model intuitively reflects the performance of the nodes in consensus. If a node has high activity, regularly participates in consensus processes, has low latency, and has a high success rate of transactions, and is often classified as a priority node, then the node score will be high. Conversely, if the node has low activity and a low completion rate of transactions, and is often classified as an ordinary node, then the score will be low. Each time a node acts maliciously, its score is directly halved.

3) SIMPLIFIED CONSENSUS PROTOCOL

For the traditional PBFT algorithm, in the reply phase, all nodes participating in the consensus return responses to the client, and the client judges whether to save the consensus based on the number of received response results. In the context of the SLI application, the purpose of consensus is to ensure the fairness, openness, and validity of people's livelihood data, which requires all parties to jointly maintain it. This study presents an improvement to the reply phase of PBFT in the context of SLI applications, in which the data are broadcast to the entire network to synchronize with nodes that did not participate in the consensus. Consensus nodes supervise the data synchronization process to maintain consistency in the data in the chain. Furthermore, the consistency protocol of the PBFT consensus algorithm two

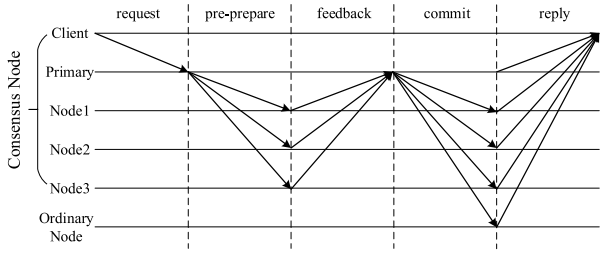

FIGURE 7. Simplified consensus protocol process.

rounds of node communication with a complexity of $O(N^2)$ to prevent Byzantine node interference, which leads to increased communication complexity and decreased communication efficiency. However, we used a the scoring model to select nodes with high scores to participate in consensus. In most cases, the probability of Byzantine nodes is very low. Therefore, in the absence of Byzantine nodes, this study simplifies the consistency protocol to improve the efficiency the of consensus. The simplified consistency protocol execution process is illustrated in Fig. 7.

The specific execution process of the simplified consensus protocol is as follows:

1) Request phase
   Similar to the request phase of the PBFT algorithm, the client sends a request message to the primary node, and the message format is <REQUEST, o, t, c>, where o is the request to execute the state machine, t is the timestamp, and c represents the client number.
2) Pre-prepare phase
   After receiving the request message from the client, the primary node generates a pre-prepared message and broadcasts it to all the consensus nodes. The message format was ≪PRE-PREPARE, o, n, d, g>, w, m>. Where w is the node's score information, which is used for node promotion and demotion processing, and g is the hash calculation result for w.
3) Feedback phase
   After receiving the pre-prepare message sent by the primary node, the consensus node first checks whether the g value in the pre-prepare message is the same as the local g value. If they are different, the local score information s is updated. The consensus node then generates a feedback message and sends it to the primary node. The message format is <FEEDBACK, v, n, d, i>, where i is the number of nodes.
4) Commit phase
   The primary node receives feedback messages from all consensus nodes. If all feedback messages are identical, the primary node generates a commit message and broadcasts it to all nodes in the network. The message format is <COMMIT, v, n, d, a>, where a indicates the confirmed added information and indicates that the primary node has confirmed the addition. After receiving the confirmation message, all the nodes add transaction information to their local memory.
5) Reply phase
   After consensus is completed, the primary node broadcasts the consensus result to all nodes in the network. The nodes that participated in the consensus supervised the broadcast message, and the nodes that did not participate in the consensus synchronized the account book.

4) SLI-PBFT CONSENSUS ALGORITHM IMPLEMENTATION PROCESS

Here is the step-by-step process of the SLI-PBFT algorithm.

1) Initialize the nodes in the current network: Assign initial scores to newly added nodes in the system based on the scoring model and execute the exit process for nodes requesting to leave. The credit scores of the existing nodes in the system were evaluated based on their performance in the previous round. All the nodes in the network were further categorized based on the range of their scores.
2) The client then submitted a transaction request. If no primary nodes are available at that time, the node with the highest score is selected as the primary node from the candidate nodes. If priority nodes are available, the primary node is selected.
3) The primary node receives the request and performs tasks, such as numbering and processing the request message. Then, it executes a simplified consensus protocol. Based on the feedback and messages received, the primary node assesses and compares the statuses of the participating nodes in the network.
4) Determine the presence of Byzantine nodes. During the feedback phase of the simplified consensus protocol, the primary node examines the hash field of the information transmitted by the consensus nodes. If the hash values are consistent, this indicates the absence of Byzantine nodes at that time. The consensus processed proceeds smoothly, and the node scores and consensus information were calculated and recorded. This completes the current consensus round, and the system awaits the next one. However, if inconsistencies are detected in the hash values, the presence of Byzantine nodes in the network is confirmed, and the primary node immediately halts the simplified consensus protocol.
5) The primary node then initiates the execution of the complete PBFT consensus protocol, in which all consensus nodes participate.
6) After consensus is completed, all nodes record the data generated during the consensus process and then calculate and update the scores of each node. If there is a node with a score below the set threshold (R_l), it is excluded from the consensus group and cannot participate in the consensus process.
7) Return to the first step, wait for the next round of consensus.

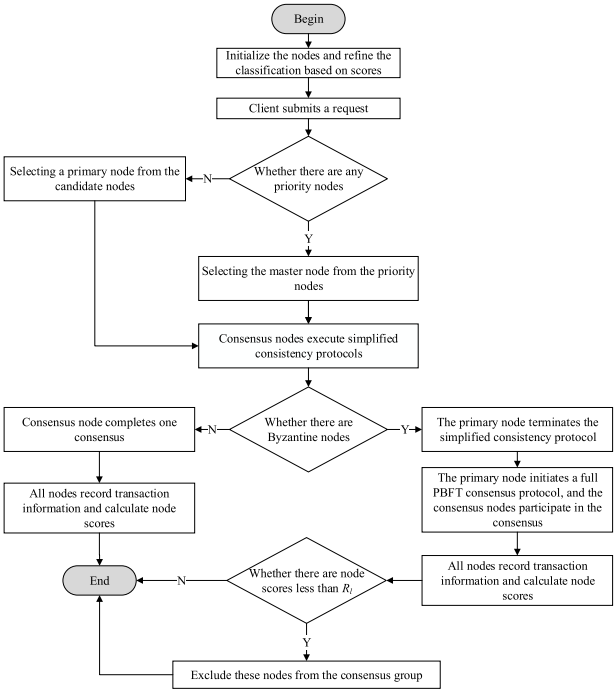

FIGURE 8. SLI-PBFT consensus algorithm process.

The process of the SLI-PBFT consensus algorithm is illustrated in Fig. 8.

5) SUMMARY OF THE SLI-PBFT ALGORITHM

In summary, the dynamic scalability mechanism allows for the addition of more computing resources and participants according to actual needs, and newly authorized nodes can be seamlessly integrate into the existing consensus network. When nodes exit the network, the system can dynamically adjust the composition of the consensus network, thereby enhancing system stability. The flexibility and fault tolerance provided by this mechanism enhance the system resilience. This scoring model incentivizes node participation in the consensus process. Nodes were rewarded with higher scores when they successfully completed consensus tasks or provided valuable contributions. This mechanism encourages active participation by the nodes, motivating them to provide better services and contributions. The simplified consensus protocol was specifically designed for scenarios without Byzantine nodes, reducing unnecessary communication frequency and data transmission between nodes using optimized algorithms and communication mechanisms. This reduces the network load and latency and improves the speed and efficiency of consensus. Nodes can allocate more resources for transaction processing and verification, thereby improving the overall throughput and processing capacity of the system to accommodate SLI application scenarios better.

By combining the dynamic scalability mechanism, scoring model, and simplified consensus protocol, the performance and scalability of the SLI-PBFT consensus algorithm were enhanced, enabling the SLI framework to meet the requirements of dynamism, high performance, and security, effectively addressing the business needs of SLI.

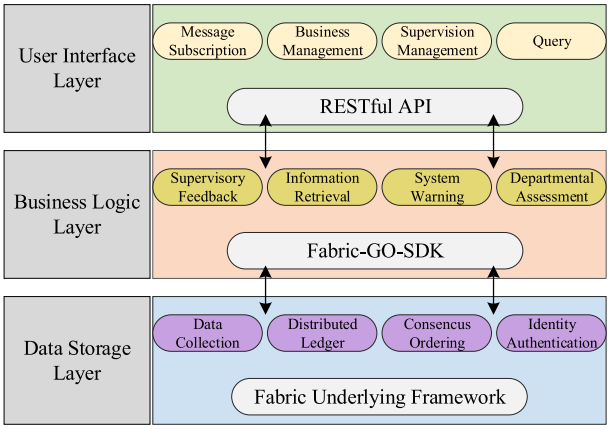

FIGURE 9. Functional layers and interconnections of prototype platform.

V. PROTOTYPE SYSTEM DEVELOPMENT AND IMPLEMENTATION

The developed prototype consists of three layers: a blockchain layer, business logic layer, and user interface layer, as shown in Fig. 9.

The primary objective of the blockchain layer is to provide various blockchain-related services, including data storage, consensus mechanisms, smart contracts, and fabric networks. During the deployment process, the following technologies and tools were employed to build the system. For data storage, multiple state databases provided by HLF, such as CouchDB and LevelDB, are utilized to store platform transaction data and contracts. The selection of these databases was aimed at meeting different data storage and performance requirements. In terms of consensus mechanisms, the proposed SLI-PBFT consensus algorithm was adopted, combining high-throughput, fault tolerance, and low-latency characteristics to ensure consistent transaction endorsement among network nodes. For smart contracts, Go was chosen as the programming language because of its excellent performance and development efficiency. Code editing and debugging were conducted in an integrated development environment called Goland. Regarding the configuration and deployment of the fabric network, emphasis was placed on ensuring the accuracy of network topology and communication among nodes. A Docker is used to containerize each node, thereby simplifying the deployment and management processes. Additionally, the HLF SDK was utilized to configure and manage the network, including tasks such as node settings and channel configuration. The application of these tools and technologies ensures network stability and reliability.

The business logic layer is the core layer of the SLI system and is responsible for implementing the main functionalities and business logic, including identity authentication,

supervision feedback, information retrieval, system alerts, and department assessment. In the implementation process of the business logic layer, the Spring Boot framework is chosen as the foundation, following a typical layered architecture approach that utilizes MyBatis to provide object-oriented data access, map Java objects to database tables, and provide a set of APIs for data manipulation. The service layer defines business logic and functional interfaces to implement the core functionalities of the system. The controller layer receives the user requests and delegates them to the corresponding processing services. In addition, the Fabric Java SDK is integrated into the Spring Boot program, offering an interface for Java applications to interact with the HLF blockchain network. This interface enables connectivity to the fabric network, invocation of smart contracts, transaction-sending and receiving, querying, event-handling, and overall interaction with the blockchain network.

The user interface layer is an integral part of the blockchain-based SLI system, providing users with Restful API interfaces to interact with the system through HTTP requests. This layer encompasses system views, pages, and user interaction components. Through careful design, users can easily perform operations such as data input, querying, analysis, and sharing, while receiving feedback and status information from the system. To build an interactive interface, the popular front-end framework Vue was chosen as the development tool, allowing the division of pages into independent components, each responsible for specific functionality or interface elements. Such component-based structures enhance code maintainability and facilitate component reuse and extension. Additionally, other frontend technologies and tools such as HTML, CSS, and JavaScript were employed to implement interface styling, functional design, and behavior.

In terms of the usage of the prototype system, firstly, after different users log in (taking the administrator as an example), the homepage of the SLI prototype system provides users with an intuitive dashboard that displays real-time information such as transaction volume, block height, smart contracts, and node statistics. These data help increase participants' real-time understanding of blockchain information. To further promote user understanding and participation in the system, instructional materials on system usage were published in system announcements and documents. These materials aim to help citizens better understand blockchain technology, the working principles of the SLI system, and how to use the tools provided by the system to participate in supervision work, thus enabling users to have a clearer understanding of the entire system. Furthermore, the statistics on unresolved and total cases allow citizens to understand the number of ongoing supervision cases and completed cases, as shown in Fig. 10.

Second, the SLI prototype system provides a case query function, allowing users to query case information stored in the blockchain by entering the case details, as shown in Fig. 11. The query results display detailed information about

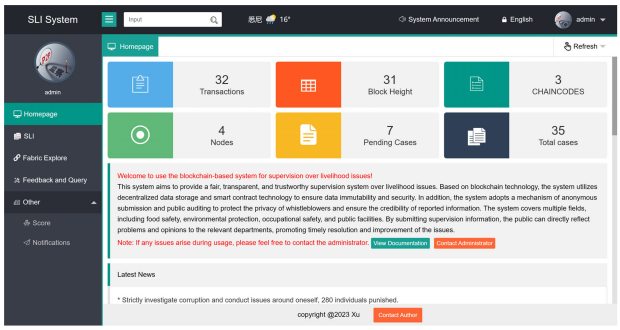

FIGURE 10. Main interface of the system.

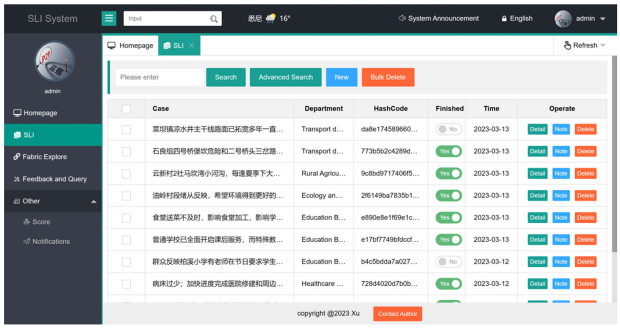

FIGURE 11. Information on the BlockChain.

the case, including the case description, handling department, hash value, processing progress, case timeline, and other information. This function helps citizens and social organizations intuitively perceive their involvement in supervisory work and understand the progress of their specific cases. For ordinary users, the interface only provides the query function and does not include editing, annotation, or deletion capabilities, which are reserved for the relevant departments.

The SLI prototype system offers a range of user interfaces and tools to facilitate interactions between users and citizens participating in the supervision process. These features and tools include real-time data display, case queries, and instructional materials, aimed at assisting citizens in better participation and understanding the blockchain-based SLI system.

VI. EXPERIMENTAL DESIGN AND ANALYSIS

In this section, we describe multiple experiments conducted to analyze and validate proposed the SLI-PBFT consensus algorithm based on the SLI application scenario. To measure performance, we employed several commonly used metrics including throughput, consensus latency, and fault-tolerance security. The measurement tool used was caliper, a widely adopted benchmarking tool that provides performance testing capabilities for blockchain platforms and consensus algorithms. Finally, we compare the performance indicators of the SLI-PBFT, PBFT, and RBFT algorithms. The experimental environment and parameters of the SLI-PBFT algorithm are listed in Table 4 and Table 5, respectively.

TABLE 4. Lab environment.

SOFTWARE	VERSION
CPU	Intel Core i7-9750H 2.60GHz
Memory	16GB RAM
Operating System	Centos7.6
Hyperledger Fabric	2.2

TABLE 5. The parameters of the SLI-PBFT.

PARAMETER	DESCRIBE	VALUE
R_{init}	Initialized score	40
d_{max}	Maximum allowed transaction latency	1000(ms)
x , y	Coefficients in the activity calculation formula	0.5 , 0.5
α , β , γ	Coefficients in the final calculation formula for node scores	0.2 , 0.3 , 0.5
R_l , R_m , R_h	Values for dividing different ranges of node scores	25 , 50 , 75

A. THROUGHPUT ANALYSIS

Throughput typically refers to the number of transactions processed by a system per unit time. This is one of the key metrics for evaluating a consensus algorithm. A higher throughput indicates better performance. In the blockchain field, throughput is often expressed as transactions per second (TPS). Its calculation is shown in Equation (5).

$$TPS = \frac{transcations}{\Delta t} \tag{5}$$

where "*transactions*" refers to the number of transactions processed by the system during the block generation time, and "Δt" refers to the block generation time. Experiments 1 and 2 were conducted to measure throughput.

- *Experiment* 1
  For comparison, we fixed the number of consensus nodes to seven and configured the caliper to initiate transactions at different volumes. The number of consensus transactions completed per second was recorded. To ensure the representativeness of the experimental results, the experiments were repeated multiple times and the average values were used to plot the experimental results, as shown in Fig. 12.

As the number of transaction requests gradually increases within the processing capacity of the nodes, the throughput of each algorithm also increases. Because of the integration of a scoring model into the SLI-PBFT consensus algorithm, which excludes nodes with values below a certain threshold from participating in the consensus process, unnecessary overhead is reduced. Additionally, the SLI-PBFT consensus algorithm simplifies the consistency protocol, resulting in simplified communication between nodes and reduced time, naturally leading to a higher throughput compared to the other two algorithms. However, when the transaction

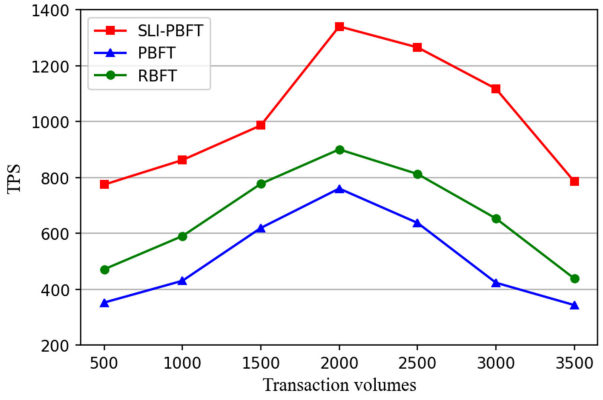

FIGURE 12. Comparison of throughput under different transaction volumes.

volume exceeded 2000, the throughput began to decline, primarily for two reasons. First, as the number of transaction requests increases, nodes are unable to process all requests in a timely manner, resulting in an increased waiting time and reduced throughput. Second, excessive requests lead to higher consumption of computational resources and storage space by the nodes to process transaction data, thereby affecting the throughput. Nevertheless, overall, the throughput of the SLI-PBFT algorithm remained higher than that of the other two algorithms.

- *Experiment* 2
  To observe the changes in the presence of Byzantine nodes, we conducted experiments using Caliper, where a fixed number of 2000 transaction requests were sent while varying the number of nodes and introducing Byzantine nodes. The number of transactions completed per second was recorded, as shown in Fig. 13. Fig. 13(a) represents the scenario without Byzantine nodes, whereas Fig. 13(b) represents the scenario with Byzantine nodes.

From Fig. 13(a), it can be observed that as the number of nodes in the network increased, the throughput of both algorithms decreased. This is because an increase in the number of nodes leads to an increase in the communication among nodes, resulting in longer processing times and reduced throughput. From Fig. 13(b), it can be observed that when Byzantine nodes are present, the throughput of the SLI-PBFT algorithm decreases rapidly. This is because, in such scenarios, the primary node needs to terminate the simplified consistency protocol and switch to an alternative protocol, which requires additional time and impacts the throughput of the algorithm. Overall, the SLI-PBFT algorithm still exhibits a higher throughput than the other two algorithms.

B. CONSENSUS LATENCY ANALYSIS

Consensus latency is an important metric for measuring the speed of consensus algorithm. A lower consensus latency indicates a faster consensus among nodes, resulting in higher system performance and efficiency. In this study, the tested

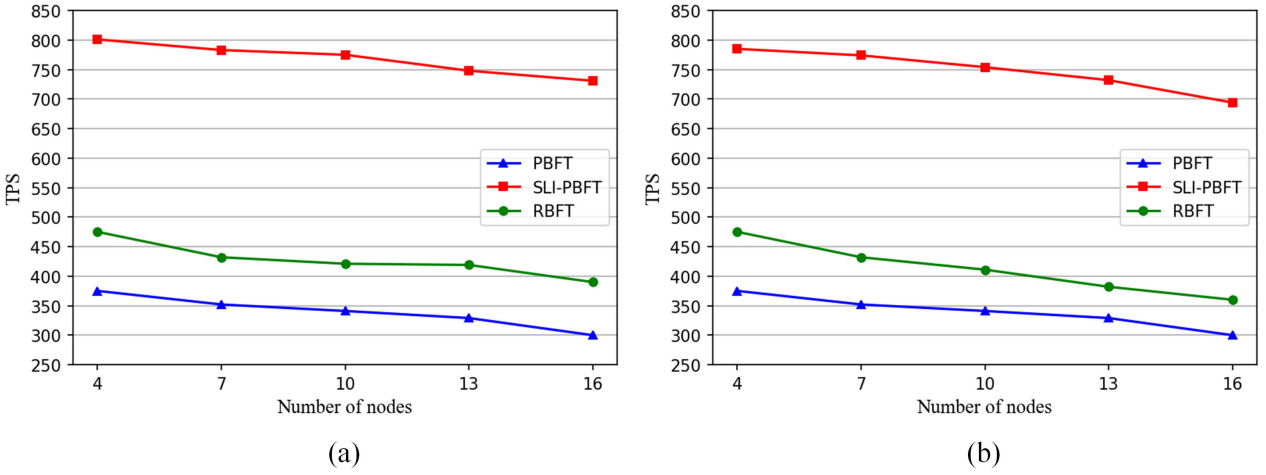

FIGURE 13. The throughput of the two algorithms changed with the number of nodes in the absence and presence of Byzantine nodes.

consensus latency was defined as the time required to initiate a transaction request to achieve consensus and complete a block. Its calculation is shown in Equation (6).

$$T_{cd} = T_{finish} - T_{request} \tag{6}$$

where T_{cd} represents the consensus latency, which indicates the time from when a client request is initiated ($T_{request}$) to the completion of block confirmation (T_{finish}) during the consensus process. Experiments 3 and 4 were designed to record the changes in consensus latency, and multiple measurements were conducted to obtain average values for plotting.

- *Experiment* 3
  We recorded the changes in the consensus latency under different node conditions while also comparing the effects of different block generation times by adjusting the configuration file. Fig. 14 shows the consensus latency of the two algorithms changing with the variation of the node quantity when the block generation time is 5 seconds (Fig. 14(a)), 10 seconds (Fig. 14(b)), 15 seconds (Fig. 14(c)), and 20 seconds (Fig. 14(d)), respectively.

From Fig. 14, it can be observed that the consensus latency increases with both the block generation time and number of nodes. This is because, as the number of nodes increases, the communication volume among nodes increases, resulting in longer processing times. Additionally, it can be observed that when different block generation times are set, latency increases. This is because, with longer block generation times, a certain amount of time must pass before confirmation and execution can occur, leading to an increase in transaction processing latency. It can also be concluded that increasing the block capacity requires a longer waiting time for a sufficient number of transactions to be confirmed and executed. Compared with PBFT and RBFT, the SLI-PBFT consensus algorithm proposed in this study exhibits a significantly lower consensus latency.

- *Experiment* 4
  To examine the impact of Byzantine nodes, we investigated the latency of the SLI-PBFT consensus algorithm, as well as the PBFT and RBFT consensus algorithms, based on the presence or absence of Byzantine nodes. Under the same block generation time (10s), we evaluated the latency of these algorithms. Fig. 15 presents the variations in the consensus latency for both scenarios: without Byzantine nodes (Fig. 15(a)) and with Byzantine nodes (Fig. 15(b)) as the number of nodes changes.

As mentioned in the analysis of the SLI-PBFT consensus algorithm, a simplified consensus protocol was executed when there were no Byzantine nodes. As shown in Fig. 15(a), as the number of nodes increases, the consensus latency of the PBFT algorithm increases rapidly, whereas that of the SLI-PBFT consensus algorithm increases slowly and remains relatively stable. This is because we have simplified the process of the SLI-PBFT consensus algorithm, and in the absence of Byzantine nodes, its time complexity is reduced to O(N), resulting a in slower growth of consensus latency compared to the O(N^2) of the PBFT consensus algorithm. Thus, it can be inferred that the SLI-PBFT algorithm has a significant advantage when there are more nodes. In the presence of Byzantine nodes, as shown in Fig. 15(b), the latency of the PBFT consensus algorithm increased as expected, but the consensus latency of the SLI-PBFT consensus algorithm increased significantly. This is because, in the presence of Byzantine nodes, the primary node suspends the simplified consensus protocol, leading to a significant increase in the consensus latency.

C. FAULT-TOLERANCE SECURITY

Fault-tolerance security is an important evaluation metric for consensus algorithms. In the current complex network environment, ensuring the secure and stable operation of the blockchain is necessary. We designed Experiment Six to

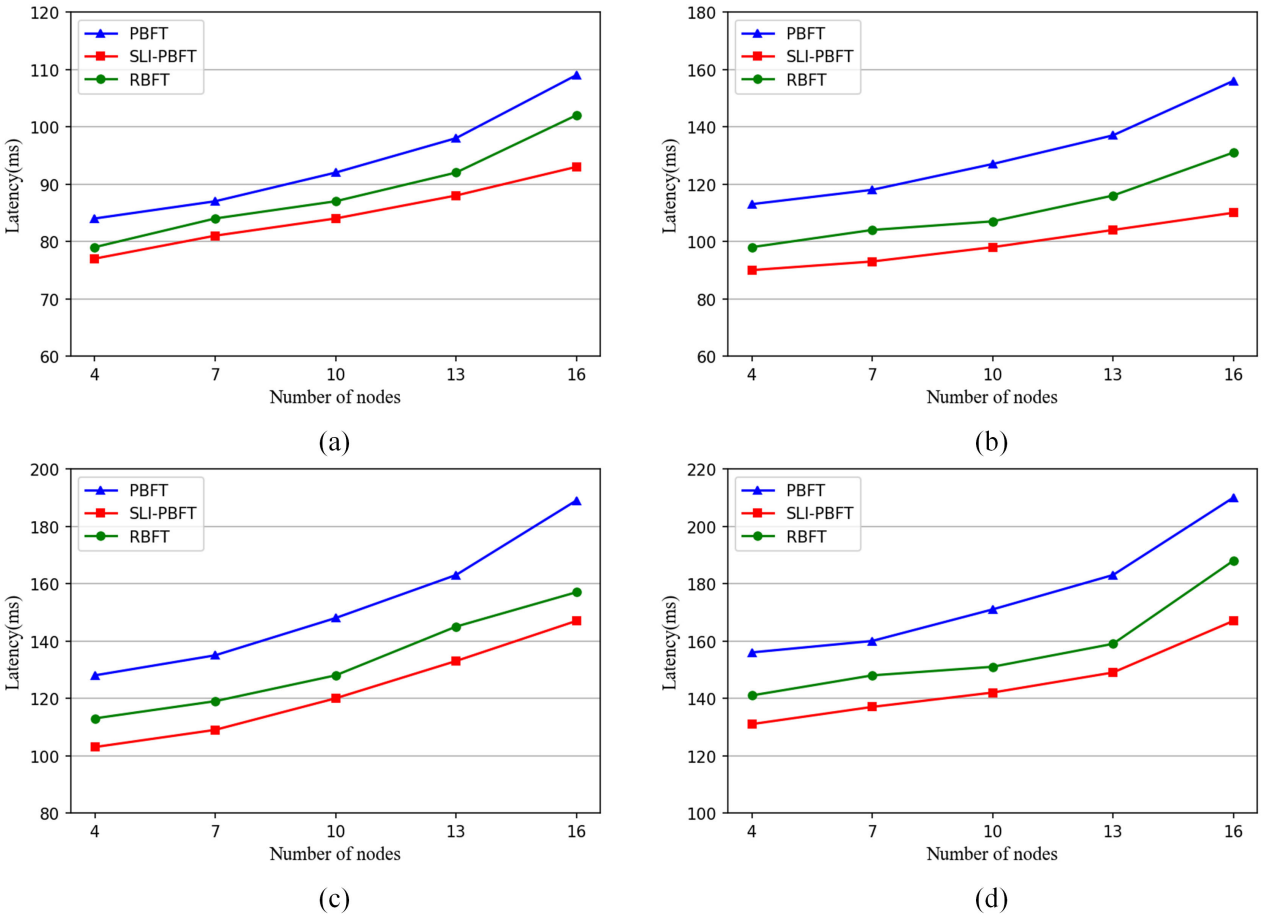

FIGURE 14. Variations in consensus latency with changes in the number of nodes under different block generation times.

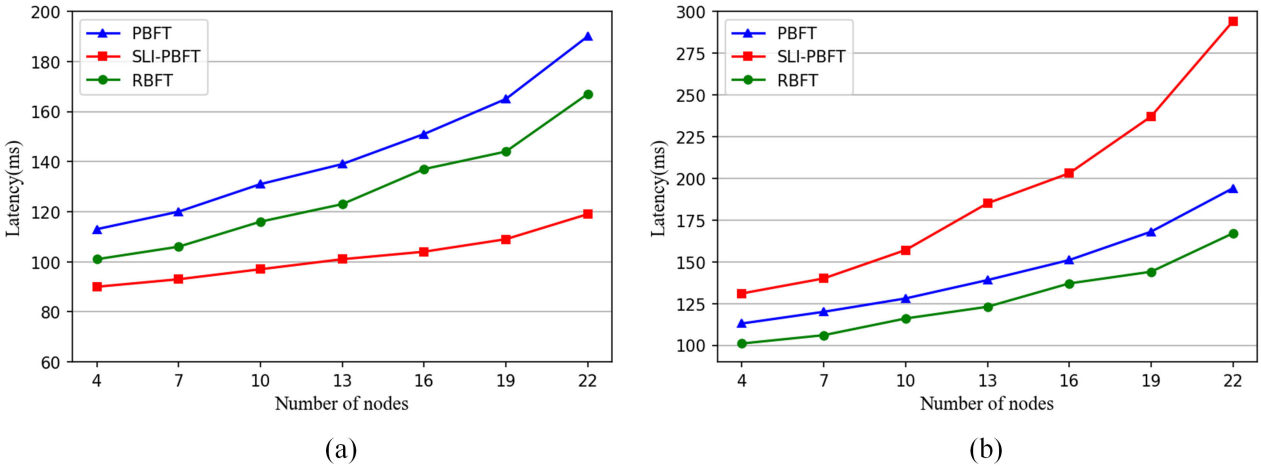

FIGURE 15. The variation of consensus latency with the number of nodes under different Byzantine fault tolerance scenarios.

test the fault-tolerance security of the SLI-PBFT consensus algorithm.

- *Experiment* 5
  We set up 50 nodes, of which 16 were marked as Byzantine nodes. They were assigned the same system parameters as in the experimental runtime environment, and 20 rounds of consensus were conducted. The comparison of Byzantine node quantities in the consensus group as the number of consensus rounds increased is shown in Fig. 16.

From Fig. 16, it can be observed that the number of Byzantine nodes in the PBFT algorithm remains constant at 16 [11].

TABLE 6. Detailed comparison of SLI-PBFT and other PBFT-based algorithms.

Consensus Mechanism	Performance	Security	Latency	Throughput	Dynamic Scalability	Time Complexity
PBFT	Middle	Low	Middle	Middle	No	$O(N^2)$
RBFT	Middle	Middle	Low	Middle	No	$O(N^2)$
P-PBFT [38]	High	Middle	Low	High	No	$O(N^2)$
ST-PBFT [37]	High	Middle	Low	High	No	$O(N)$
SLI-PBFT	High	High	Low	High	Yes	$O(N)$

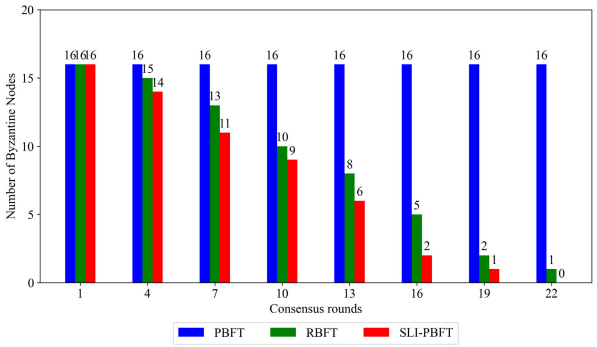

FIGURE 16. The variation of Byzantine node count with increasing consensus rounds.

However, for the RBFT algorithm, the number of Byzantine nodes gradually decreased as the number of consensus rounds increased. For the SLI-PBFT consensus algorithm, after the 22nd consensus round, the number of Byzantine nodes decreased to zero. It can be seen that under long-term operation conditions, the RBFT and SLI-PBFT algorithms exclude Byzantine nodes from the consensus group through reputation values and scoring mechanisms, enhancing the reliability and security of participating consensus nodes. This significantly reduces the probability of Byzantine nodes being selected as primary nodes. Particularly, the SLI-PBFT consensus algorithm, with its scoring model, can rapidly eliminate Byzantine nodes, thereby demonstrating superior efficiency and fault-tolerance security. Thus, it meets the application requirements of SLI business.

However, the SLI-PBFT algorithm has the same tolerance for Byzantine nodes as PBFT, that is $f = (n-1)/3$. Blockchain is a special type of distributed system in which each node has a copy of all transaction information;c if a node in the blockchain is Byzantine, other nodes can verify the transaction information and detect the error. However, when the number of Byzantine nodes in the system exceeds f, it is not possible to guarantee the sufficient participation of honest nodes in the consensus, leading to system paralysis. By contrast, the SLI-PBFT consensus algorithm selects nodes to participate in the consensus based on their scores. These nodes have a reliable score guarantee, and their scores are calculated during each consensus round, excluding nodes with low scores from the consensus group. Therefore, the SLI-PBFT algorithm can reduce Byzantine nodes, thereby enhancing the robustness and improving the system security performance.

Finally, for a detailed comparison among SLI-PBFT, PBFT, RBFT, and the consensus algorithms proposed in references [37], [38], please refer to Table 6.

D. SUMMARY OF SLI-PBFT

The SLI-PBFT consensus algorithm has several advantages over other algorithms. More importantly, SLI-PBFT is well-suited for the application scenarios of SLI. It allows for node join and exit, which aligns with the requirements of the SLI operations. It optimizes the selection process of the primary node, ensuring fairness in node selection, while reducing the risks of malicious behavior and attacks. Furthermore, it simplifies the consensus process, thereby reducing the communication overhead and accelerating the block generation speed within the system's operational limits, thereby enhancing the user experience.

VII. CONCLUSION AND PROSPECTS

First, this study proposes an SLI framework supported by the consortium blockchain technology. In this framework, data uploaded by citizens, social organizations, and relevant government departments are recorded in blocks after participating in the consensus process. Access permissions are controlled through smart contracts, enabling better collection, analysis, aggregation, and notification of information on the blockchain. It also facilitates the evaluation and assessment of the performance of the relevant departments. In addition, to address the issues of high communication complexity, random selection of primary nodes, and limited network scalability in the widely used PBFT consensus algorithm for consortium blockchains, the SLI-PBFT consensus algorithm was proposed as a solution. Finally, a blockchain prototype system based on the HLF open-source architecture is developed to validate the feasibility of the SLI framework. To test the performance of the SLI-PBFT consensus algorithm in this system, tests and comparisons were conducted between the SLI-PBFT, PBFT, and RBFT consensus algorithms, thereby confirming the feasibility of the proposed SLI-PBFT algorithm.

The blockchain-based SLI framework has certain advantages but also has limitations:

1) Blockchain is transparent and public; however, in this framework, certain supervision data require anonymous processing, such as personal privacy information from supervisors. Protecting the privacy of these data is a key challenge in SLI. Future research could focus on enhancing and refining the proposed blockchain-supported SLI conceptual framework to incorporate privacy protection mechanisms. Techniques such as zero-knowledge proofs, secure multiparty computation, acnd differential privacy can be explored to protect sensitive data while maintaining the benefits of transparency and openness for other data.
2) Although blockchain technology has decentralization and immutability characteristics, if the supervision data sources themselves have issues such as missing data, mislabeling, or noise, the SLI framework proposed in this study cannot solve these problems. Therefore, future considerations could involve integrating other artificial intelligence techniques, such as deep learning algorithms, to identify and repair SLI data.
3) The SLI-PBFT consensus algorithm proposed in this study selects a primary node based on scores. If a node has a high score, it is more likely to be selected as the primary node, creating the risk of continuous selection of the same node as the primary node, which could lead to excessive centralization of the system and increase the potential risks of central control and a single point of failure. To mitigate this centralization risk, future research must design suitable mechanisms and algorithms to balance node scores and the process of selecting primary nodes, ensuring the decentralized nature and robustness of the system.

This study has made significant progress in improving the efficiency, fairness, and security of SLI in the field of livelihood issues by proposing an innovative SLI framework. The proposed framework provides a reliable foundation for optimizing government-related SLI initiatives and serves as an important reference for promoting public participation in supervision and improving government decision-making. The research presented in this article offers valuable insights and approaches to enhance the quality and effectiveness of SLI in the field of livelihood issues, demonstrating its practical significance and application value.

REFERENCES

[1] Y. Q. Chen and H. Huang, "Mechanisms, challenges and strategies of big data embedded supervision of livelihood issues at the grassroots: A case study of livelihood supervision platform in Q district of C city," *Social Sci. Res.*, vol. 263, no. 6, pp. 14–24, 2022.

[2] J. Lu, "Exploration of path for improving supervisory capacity in procuratorial work under the background of big data strategy," *Sci. Technol. Law*, vol. 13, no. 1, pp. 48–56, 2023.

[3] R. Wang and X. Ni, "Digital platform driven mass supervision and government response: Taking the supervision information platform in the livelihood field of province a as an example," *Governance Studies*, vol. 39, no. 2, pp. 124–138&160, 2023.

[4] Z. Zeng and Z. Wang, "Digital supervision: A new form of power supervision system in the big data era," *E-Government*, vol. 228, no. 12, pp. 59–68, Dec. 2021.

[5] Z. Xie and F. Fan, "Value, logic, and prospects of big data-driven supervision of livelihood issues: A case study of 'T county livelihood supervision big data platform,"' *Chin. Public Admin.*, vol. 426, no. 12, pp. 125–131, 2020.

[6] L. Zhang, J. Mao, Y. An, T. Zhang, J. Ma, C. Feng, and X. Zhou, "A systematic review of blockchain technology for government information sharing," *CMC-Comput. Mater. Con.*, vol. 74, no. 1, pp. 1161–1181, 2023.

[7] F. Lumineau, W. Wang, and O. Schilke, "Blockchain governance—A new way of organizing collaborations?" *Org. Sci.*, vol. 32, no. 2, pp. 500–521, Mar. 2021.

[8] S. Johar, N. Ahmad, W. Asher, H. Cruickshank, and A. Durrani, "Research and applied perspective to blockchain technology: A comprehensive survey," *Appl. Sci.*, vol. 11, no. 14, p. 6252, Jul. 2021.

[9] M. Touloupou, M. Themistocleous, E. Iosif, and K. Christodoulou, "A systematic literature review toward a blockchain benchmarking framework," *IEEE Access*, vol. 10, pp. 70630–70644, 2022.

[10] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020.

[11] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Design Implement.*, 1999, pp. 173–186.

[12] J. Xue, Y. He, M. Liu, Y. Tang, and H. Xu, "Incentives for corporate environmental information disclosure in China: Public media pressure, local government supervision and interactive effects," *Sustainability*, vol. 13, no. 18, p. 10016, Sep. 2021.

[13] T. Vian, "Anti-corruption, transparency and accountability in health: Concepts, frameworks, and approaches," *Global Health Action*, vol. 13, Feb. 2020, Art. no. 1694744.

[14] W. Zhang and L. Pang, "Multiple collaborative supervision pattern recognition method within social organizations based on data clustering algorithm," *J. Math.*, vol. 2021, pp. 1–12, Dec. 2021.

[15] Y. Hu, W. Wang, and H. Cheng, "Analysis of administrative supervision cases in procuratorial organs under the perspective of big data," *Chin. Procurators*, vol. 390, no. 12, pp. 23–26, 2022.

[16] X. Zhang and W. Tian, "Grid supervision path of platform food safety collaborative governance based on big data," *Int. Trans. Electr. Energy Syst.*, vol. 2022, pp. 1–14, Sep. 2022.

[17] H. Liu, Z. Yu, X. Zhong, and H. Yu, "Network public opinion monitoring system for agriculture products based on big data," *Sci. Program.*, vol. 2021, pp. 1–17, Jun. 2021.

[18] X. Jin and X. Jin, "Application of big data technology in environmental pollution control in energy ecological economic zone," *Int. Trans. Electr. Energy Syst.*, vol. 2022, Aug. 2022, Art. no. 1569905.

[19] B. Cao, Z. Wang, L. Zhang, D. Feng, M. Peng, L. Zhang, and Z. Han, "Blockchain systems, technologies, and applications: A methodology perspective," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 353–385, 1st Quart., 2023.

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, p. 21260, Oct. 2008, doi: 10.2139/ssrn.3440802.

[21] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, and S. Muralidharan, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15.

[23] S. Perera, S. Nanayakkara, M. N. N. Rodrigo, S. Senaratne, and R. Weinand, "Blockchain technology: Is it hype or real in the construction industry?" *J. Ind. Inf. Integr.*, vol. 17, Mar. 2020, Art. no. 100125.

[24] L. Hang and D.-H. Kim, "Optimal blockchain network construction methodology based on analysis of configurable components for enhancing hyperledger fabric performance," *Blockchain, Res. Appl.*, vol. 2, no. 1, Mar. 2021, Art. no. 100009.

[25] Q. Tao, X. Cui, X. Huang, A. M. Leigh, and H. Gu, "Food safety supervision system based on hierarchical multi-domain blockchain network," *IEEE Access*, vol. 7, pp. 51817–51826, 2019.

[26] X. Peng, X. Zhang, X. Wang, J. Xu, H. Li, Z. Zhao, and Z. Qi, "A refined supervision model of rice supply chain based on multi-blockchain," *Foods*, vol. 11, no. 18, p. 2785, Sep. 2022.

[27] Y. Li, K. Ouyang, N. Li, R. Rahmani, H. Yang, and Y. Pei, "A blockchain-assisted intelligent transportation system promoting data services with privacy protection," *Sensors*, vol. 20, no. 9, p. 2483, Apr. 2020.
[28] M. Zhao, W. Liu, and K. He, "Research on data security model of environmental monitoring based on blockchain," *IEEE Access*, vol. 10, pp. 120168–120180, 2022.
[29] B. Zhong, J. Guo, L. Zhang, H. Wu, H. Li, and Y. Wang, "A blockchain-based framework for on-site construction environmental monitoring: Proof of concept," *Building Environ.*, vol. 217, Jun. 2022, Art. no. 109064.
[30] A. H. Ameen, M. A. Mohammed, and A. N. Rashid, "Dimensions of artificial intelligence techniques, blockchain, and cyber security in the Internet of Medical Things: Opportunities, challenges, and future directions," *J. Intell. Syst.*, vol. 32, no. 1, Apr. 2023, Art. no. 20220267.
[31] M. A. Mohammed, A. Lakhan, K. H. Abdulkareem, D. A. Zebari, J. Nedoma, R. Martinek, S. Kadry, and B. Garcia-Zapirain, "Energy-efficient distributed federated learning offloading and scheduling healthcare system in blockchain based networks," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100815.
[32] A. Lakhan, M. A. Mohammed, J. Nedoma, R. Martinek, P. Tiwari, A. Vidyarthi, A. Alkhayyat, and W. Wang, "Federated-learning based privacy preservation and fraud-enabled blockchain IoMT system for healthcare," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 2, pp. 664–672, Feb. 2023.
[33] P. Gaži, A. Kiayias, and D. Zindros, "Proof-of-stake sidechains," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 139–156.
[34] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 172–181, Jan. 2020.
[35] K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-based Byzantine fault-tolerance for consortium blockchain," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Singapore, Dec. 2018, pp. 604–611.
[36] G. Xu and Y. Wang, "Improved PBFT algorithm based on vague sets," *Secur. Commun. Netw.*, vol. 2022, p. 6144664, Mar. 2022.
[37] W. Zhong, W. Feng, M. Huang, and S. Feng, "ST-PBFT: An optimized PBFT consensus algorithm for intellectual property transaction scenarios," *Electronics*, vol. 12, no. 2, p. 325, Jan. 2023.
[38] S. Liu, R. Zhang, C. Liu, and D. Shi, "P-PBFT: An improved blockchain algorithm to support large-scale pharmaceutical traceability," *Comput. Biol. Med.*, vol. 154, Mar. 2023, Art. no. 106590.
[39] P. Gaba, R. S. Raw, M. A. Mohammed, J. Nedoma, and R. Martinek, "Impact of block data components on the performance of blockchain-based VANET implemented on hyperledger fabric," *IEEE Access*, vol. 10, pp. 71003–71018, 2022.
[40] M. Graf, R. Kusters, and D. Rausch, "Accountability in a permissioned blockchain: Formal analysis of hyperledger fabric," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Sep. 2020, pp. 236–255.
[41] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43620–43652, 2021.
[42] X. Fu, H. Wang, and P. Shi, "A survey of blockchain consensus algorithms: Mechanism, design and applications," *Sci. China Inf. Sci.*, vol. 64, no. 2, pp. 1–15, Feb. 2021.
[43] S. Liu, R. Zhang, and C. Liu, "Improvement of PBFT consensus algorithm based on grouping and credit rating," *Comput. Eng.*, pp. 1–13, 2023, doi: 10.19678/j.issn.1000-3428.0066247.

JINYUE XU was born in Sichuan, China. He received the bachelor's degree from the School of Computer Science and Technology, Henan Polytechnic University, China. He is currently pursuing the master's degree with the School of Computer Science and Engineering, Sichuan University of Science and Technology, China. His current research interests include deep learning, blockchain, natural language processing, and their applications in disciplinary inspections and supervision.

CAIJIAN HUA received the B.S. degree in automotive engineering from the South China University of Technology, Guangzhou, China, in 2001, the M.S. degree in computer science and technology from the Southwest University of Science and Technology, Mianyang, China, in 2005, and the Ph.D. degree in measuring and testing technologies and instruments from Sichuan University, Chengdu, China, in 2012.

From 2006 to 2014, he was a Lecturer at the School of Computer Sciences. Since 2015, he has been an Associate Professor with the Software Engineering Department, Sichuan University of Science and Technology. He is the author of two books and more than 20 articles. His research interests include computer vision, machine learning, blockchains and applications, and data analysis and applications. He is a member of the China Computer Federation.

YAN ZHANG received the B.S. degree in computer science and technology from the Sichuan University of Science and Technology, Zigong, China, in 2002, and the M.S. degree in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2010.

From 2003 to 2007, she was an Assistant Lecturer with the School of Computer Science. Since 2008, she has been a Lecturer with the Computer Science and Technology Department, Sichuan University of Science and Technology. Her research interests include enterprise informationalization and applications, the Internet of Things, and applications.