Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **8**, Number **2**, May **2023**

Hyperledger Indy-based Roaming Identity Management System

Yi-Jen Su*

Department of Computer Science and Information Engineering
Shu-Te University, Taiwan
82445 No. 59, Hengshan Rd., Yanchao Dist. Kaohsiung City, Taiwan
iansu@stu.edu.tw

Wen-Chong Hsu

Department of Computer Science and Information Engineering
Shu-Te University, Taiwan
82445 No. 59, Hengshan Rd., Yanchao Dist. Kaohsiung City, Taiwan
s19639102@stu.edu.tw

*Corresponding author: Yi-Jen Su

Received December 8, 2022, revised February 26, 2023, accepted April 24, 2023.

Abstract. *The traditional international roaming service architecture is prone to time delay due to the low efficiency of the service data exchange management process. Users get roaming services through this problem, which makes mobile service providers vulnerable to roaming fraud, and then leads to billions of dollars of economic losses every year.*

This study proposed a mechanism of roaming identity management framework Hyperledger-based Indy blockchain application technology combined with Decentralized Identifiers (DIDs) and Verifiable Credentials Model (VCM) proposed by DID to solve the problems of roaming service data exchange management and user fraud. In the proposed mechanism, users can create their identity for any role, transmit information, control their identity, and have control over disclosing the amount of information to each other. The proposed framework aids in realizing the independent control of Self-Sovereign Identity (SSI) without the authentication or management of a third-party organization. The identity information has the security features of openness, transparency, and high privacy to ensure that the identity and data are not leaked.

In the study, a simulated roaming blockchain network environment was set up to test the roaming identity management mechanism using the Hyperledger Caliper performance testing tool. Compared with the current roaming case based on Ethereum blockchain technology, the transaction processing performance of this study was improved by about three times.

Keywords: Blockchain, Roaming service, Identity management, Hyperledger Indy

1. **Introduction.** The use of smartphones connected to the internet for mobile services has become increasingly popular. Roaming services allow mobile phone users to use the same telecom services, even when leaving the area in which their services were originally applied. This is made possible using the Long-Term Evolution of mobile technology services provided by the mobile internet provider or operator. In essence, this service extends the retail voice and data services of the operator in the user's country to the

visited country via roaming agreements between the operators in the user's country and the visited country.

Groupe Speciale Mobile Association (GSMA), a global organization of mobile network operators with over 750 official members, forecasts that worldwide, 50 billion, or 60% the population, will use mobile internet by 2021 [1]. This steady growth of revenue for mobile service providers (MSPs) brings with it more challenges, such as security issues [2] and delayed data exchange of customer services. Roaming fraud and roaming identity management are the two most concerning issues for MSPs that result in delays in data exchange processes and illegal means to acquire free roaming services, costing MSPs billions of dollars per year [3]. When a user subscribes to a roaming service, their main mobile service is transferred from the Home Public Mobile Network (HPMN) to the Visited Public Mobile Network (VPMN) [4]. Data exchange delays between the two networks create a gap from when the user completes the service and when HPMN receives the service content and report. This can lead to fraudulently obtaining the subscription from HPMN, with HPMN being unable to predict or collect the correct usage billing fee for the user, and this may take up to four hours or longer to be discovered and addressed.

Blockchain technology can address the issues of long delays in roaming data exchange and roaming fraud [5]. It relies on its decentralized ledger, cryptography-enabled data encryption, and consensus mechanism properties to maintain data consistency, allowing for the efficient resolution of roaming fraud in a decentralized framework [6]. Roaming protocols in Blockchain technology are written as smart contracts, and data is stored in the Ethereum blockchain as hashes, utilizing digital wallets in conjunction with the smart contract. The cost of the transaction is paid with Ether to access roaming subscription services and user information. The Ethereum platform can only verify 25 transactions per second (TPS) [7]; nonetheless, data storage in Ethereum takes 10-20s [7], and smart contract programming is open and vulnerable to malicious attacks through loopholes [8]. Creating private chains can improve transaction speed, but suppliers are in a competitive relationship. This solution cannot be implemented in Ethereum if the agreement is limited to only a few suppliers instead of being open to all suppliers [9].

This research proposes the utilization of Hyperledger Indy, a part of the Hyperledger project, to create a roaming identity management mechanism. With distributed ledger blockchain technology, the framework and features of Indy provide a point-to-point identity solution to all users, providers, and exchange machines, in the roaming system and process. Each role can add, modify, delete, and decide how much information of its controlled identity to disclose to the other party, emphasizing self-sovereign identity (SSI) autonomy control without being verified or regulated by a third-party agency. Its identity is transmitted in the decentralized identifiers (DIDs) JSON data format. In the roaming scenario, the Hyperledger Fabric chaincode realizes roles, data, and behavior for writing smart contracts and recording them to the Hyperledger Fabric blockchain node to avoid economic losses due to data exchange delays or roaming fraud.

2. **Related Works.**

2.1. **Blockchain.**

The concept of blockchain was first introduced by Nakamoto in 2008. It combines decentralization, encryption of transaction messages, and immutability of records into a single system. In a blockchain network, nodes are computers that use consensus computing to establish the accounting authority of each node and record every transaction. Each node holds a full ledger of transaction messages, and nodes are connected through a P2P network to form a blockchain. As Blockchain is not controlled by any single organization, it boasts high security, privacy, transparency of transaction

records, irreversibility, and decentralization [10]. Additionally, its data has strong security features such as confidentiality, integrity, authentication, and nonrepudiation.

Distributed Ledger Technology (DLT) leverages a distributed network with a consensus mechanism [11], leveraging public-key cryptography [12] to securely store all messages in an encrypted protocol, to effectively address concerns of a central authority manipulating or altering data on the network.

The first Blockchain 1.0 cryptocurrency was Bitcoin, implemented as a decentralized DLT-based payment system. To initiate a transaction on the Bitcoin blockchain, users need a 27-34-character alphanumeric identifier (Bitcoin Address) and a public and private key to encrypt the currency with a hash value and then send it to the blockchain. The signer of the encrypted message publishes the signed message to the network, and the verifier then uses the signer's public key and the processed hash message to compare and verify the correctness of the transaction message [13].

2.2. **Ethereum.** In 2013, Buterin proposed Ethereum, a Turing-complete blockchain 2.0 platform [14], enabling the public to use blockchain technology to create decentralized applications (DApps) and operate in various application fields. The main core technology of Ethereum is smart contracts: they allow for the automated execution of contract agreements, thereby reducing the need for trusted intermediaries and associated costs and losses due to fraud [15]. The contents of a smart contract typically include a transaction hash, status, block number, address, value, gas, and the contract program.

2.3. **Hyperledger.** In December 2015, the Linux Foundation announced the open-source project [16], bringing together over 250 leading companies from the financial, IoT, and technology sectors to develop a distributed ledger based on blockchain. The purpose of Hyperledger is to provide a secure, enterprise-grade distributed ledger framework and codebase for global enterprise blockchain deployments, with the main goal of enabling blockchain and distributed ledgers to share and collaborate across industries. Hyperledger offers various tools, libraries, and frameworks for building distributed ledger applications, such as smart contracts and decentralized apps.

2.4. **Hyperledger Fabric.** Hyperledger Fabric is a distributed ledger platform with a modular architecture that enables its blockchain to be highly encrypted, easily deployed, and conveniently scalable [17]. It is designed as a private and permissioned blockchain system and supports plug-and-play modular components. Its core functions are to provide a distributed ledger solution for enterprise blockchain applications, comprising Membership Services, Blockchain Transactions, and Chaincode Services.

- Membership services: The Membership Services Provider manages the identities of all nodes within the system, including clients, Peer Nodes, and Ordering Service Nodes (OSNs). It authenticates and authorizes credentials for authentication and supervision. All nodes in Fabric are authenticated through encrypted messages with digital signatures, ensuring secure access.
- Blockchain transactions: Ledger offers two main services: Blockchain and State. Blockchain is a chain of data blocks that provides consensus mechanisms, distributed ledgers, and Channel mechanisms to record the history of transactions. The State involves a key-value mapping of the current date to the ledger, managed and maintained by the Peer Transaction Manager with versioned keys. LevelDB and Apache CouchDB are utilized to query and operate the implementation.
- Chaincode services: Fabric's smart contract, known as Chaincode, can be written in programming languages such as Go, Java, and Node.js. This program manages the business logic of network members. Unlike Ethereum, FabricChain separates

the program and underlying framework. These programs do not need updating or upgradation, and new programs can be migrated to convert discrete logic into actual data. Once the Chaincode is running in a secure Docker container and activated, it uses gRPC to connect to the Chaincode Peer Node.

2.5. **Hyperledger Indy.** Hyperledger Indy is a tool and library developed by Evernym and donated to the Sovrin Foundation. It facilitates the development of digital identities based on blockchain or other distributed ledgers to address identity authentication issues with blockchain technology. Hyperledger Indy also provides a user authentication scheme for the Hyperledger blockchain ecosystem [18]. Through the Verifiable Credentials Model (VCM), digital identities can be achieved in a decentralized, self-sovereign, and independent manner. Verifiable credentials are structured in an encrypted manner, with four key attributes: issuer, recipient, tamper-proof claims, and non-revoked status.

Hyperledger Indy enables identity owners to independently control their data and relationships and construct a portion of identity owner transactions related to its structure. Based on open standards and secure mechanisms of public key cryptography, it can be interoperated with other distributed ledgers; trust is the main feature of the decentralized identity system. Indy provides an accessible source to support user-controlled related ID verifiable claims and also provides a revocation mode to handle the situation when these claims are no longer correct. Verifiable Claims are the key part of this platform, enabling the exchange of related transactions. The main features of this platform include a decentralized, distributed ledger designed specifically for digital identities, featuring an anti-correlative design, DID as the sole global ID that can be resolved without any centralized resolution agency via a distributed ledger, and a 1-to-1 secure relationship when creating identity. In addition, Zero Knowledge Proofs are used to prove that part or all the data is true without revealing any information, including the identity of the proofer.

2.6. **Hyperledger Ursa.** Hyperledger Ursa is a secure, shared cryptographic library [19] for the Hyperledger blockchain framework and enables developers to create and manage cryptographic keys, sign, and verify digital signatures, and more. Ursa is written in Rust and has interfaces for Go, Python, and Java; it is designed to be modular and extensible, for example used in Hyperledger Fabric, Hyperledger Indy, and Hyperledger Sawtooth.

As the Hyperledger project matures, complex processes are needed to implement encryption processing for various Hyperledger projects. Rather than each project having to independently implement its encryption protocols, it is better to cooperate in sharing the encryption service, which includes: (1) Avoiding duplication – this cryptological library allows projects to share encryption implementations and thus avoid unnecessary duplication and extra work. (2) Improved security – by storing most, or all, of the encryption code in a single unit, the security analysis of the Hyperledger cryptography component is simplified. (3) Expert evaluation – centralizing all encryption code in one unit allows for centralized expert review and reduces the possibility of dangerous security vulnerabilities. (4) Cross-platform interoperability – if two projects use the same cryptology library, cross-platform interoperability is facilitated, as both sides of the encryption authentication involve the same encryption protocols.

2.7. **Decentralized Identifiers.** Currently, centralized IDs are managed in a centralized framework to collect or store data, with administrators controlling or limiting user access or use of their data. For data consistency and management, centralized management is very convenient, but there are also problems of excessive central authority and data leakage. These institutions decide their stay or withdrawal and can revoke at any time. Users can be authenticated by institutions only under certain circumstances, and authentication

will no longer be valid with the failure or disappearance of an organizational mechanism, causing a possible leak of unnecessary information. In various authentication scenarios, third parties may be maliciously copy and re-proclaim using fraudulent or inducement methods, resulting in "identity theft".

DIDs [20] are a new form of digital identity that enables decentralized authentication. In the Web 3.0 era, returning control of personal identity information to the user is becoming increasingly popular, with only limited data being exposed externally for use, ensuring secure and convenient data transmission. DIDs can be detached from centralized registries, identity providers, and certificate-issuing organizations. The controller of a DID can prove its control without needing permission from anyone, and the DID provides a Uniform Resource Identifier (URL) for the subject associated with the DID document, allowing trusted mutual operations with the subject.

Each DID document can express its encrypted data, authentication methods, or services, and the service mechanism of the DID can enable its controller to prove the trustable interactions associated with the DID subject. If the subject is a trusted resource such as a data model, the DID can provide a method to return itself. As an individual or organization, many types of Globally Unique Identifiers are used in various situations, such as communication addresses (e.g., phone numbers, email addresses, and social media usernames), ID numbers (e.g., passports, driver's licenses, and insurance cards), and product labels (e.g., serial numbers, barcodes, and RFID). Uniform Resource Identifiers are used for resources on the web, and each web page has a globally unique URL that can be queried in the browser.

3. **Research Method.** The international roaming subscription service has three main players: users, Home Public Mobile Network (HPMN), which provides user identity information and stores data, and Visiting Public Mobile Network (VPMN), which provides roaming service to the country. In the traditional roaming system architecture, both MSPs must first sign a roaming agreement. When the user wishes to subscribe to a roaming service, the VPMN sends a subscription request. Upon receiving the request, the VPMN confirms the identity information of the user and Call Detail Records (CDRs) with the HPMN. If the information is accurate, the roaming service is provided. After finalizing the roaming agreement and subscription, the VPMN sends its roaming agreement, CDRs, user information, and Transfer Account Procedure to the Data Clearing House (DCH) for verification and transmission back to the HPMN for confirmation. Subsequently, the service fee is paid to the VPMN based on the roaming agreement, and the user can then use the roaming service. The drawbacks of the traditional roaming architecture include:

- The DCH-managed data exchange has low information transmission efficiency between service providers, and users can obtain roaming subscriptions through roaming fraud.
- DCH is a centralized system institution, and the power of the management mechanism is too centralized; thus, its data and information are prone to be leaked by people with ulterior motives or become the target of hackers.
- The services between providers for accessing user authentication, authorization, and billing are legally different in terms of national agreement regulations, and the users and providers can easily have consumer disputes.

This study proposed a Hyperledger Indy blockchain technology-based solution to effectively address the delay in information exchange between HPMN and VPMN caused by fraudulent activities during international roaming services, resulting in economic losses. The user requests service exchange and access between the protocol, data, and identity between HPMN and VPMN to establish identity and access data in Hyperledger Indy,

transmitting and storing records in DID format to the blockchain to be verified by other nodes. The operation schematic diagram is shown in Figure 1. This proposed solution has three advantages:

- Storing roaming service-related data in the blockchain using chaincode can effectively improve the efficiency of the data exchange process to prevent a user from roaming fraud.
- Using the Hyperledger Indy blockchain framework to implement DCH to manage the identity of suppliers and their users and the blockchain feature to avoid the problem of centralization of rights and data being attacked.
- Writing the various national agreements into chaincode effectively solves the consumer disputes caused by the regulations of various countries' laws.

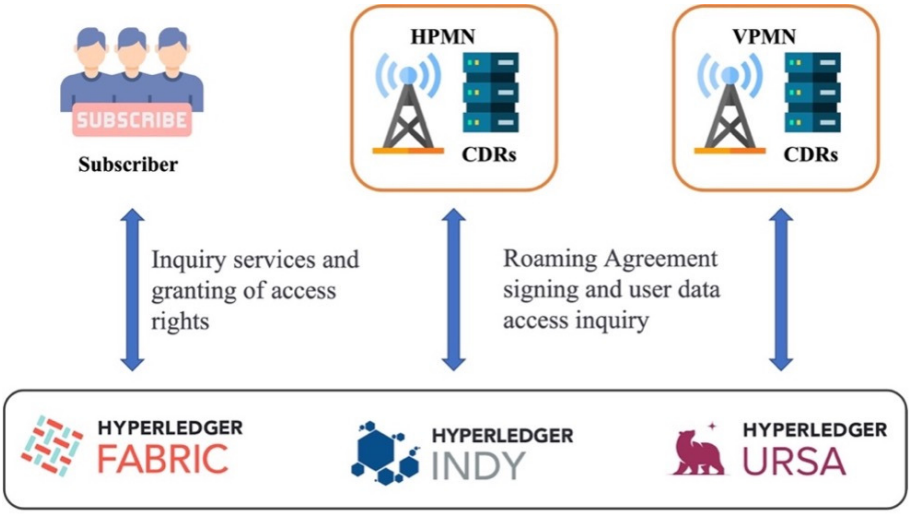

FIGURE 1. Hyperledger Indy roaming system architecture

This research proposed an identity management mechanism for international roaming service scenarios using the W3C's Verifiable Credentials Model (VCM) [21]. Compared to paper certificates, VCM utilizes four attribute evaluations that can only be verified with cryptographic algorithms, and are thus difficult to forge. When the verifier receives the credential from the holder, the blockchain-verifiable registry message is used for the encryption calculation of the four attributes, further making them difficult to forge. The four attributes of the credential model include: (1) Issuer: responsible for verification and audit, then issues the credential after verification to the requester; (2) Holder: controls the user's credentials; (3) Verifier: checks and verifies the requester's credibility; and (4) Registry: stores all digital identity users and data.

In the VCM operating architecture, users first generate and review credentials for requestors from the issuer. The issuer plays the role of a company, industry, government, event, or organization, which creates verifiable certificates by requesting claims and then sends the verified certificates to the holder. The holder holds various certificates to establish connections with others or things. Lastly, the verifier verifies the certificates and records all events in the registry. In this study, the functions designed for VCM operation were divided into three methods: creating certificates, linking identities, and verifying identities, respectively corresponding to the three qualities of VCM mode: Issuer, Holder, and Verifier. The registry attribute stores all events in the blockchain in JSON data format.

The detailed procedure of the user identity creation function is shown in Algorithm 1. First, the correctness of the user ID is determined (lines 1-3), then the DID-related document information and government identity proof are created (lines 4-6), and all proofs and documents are packaged and sent to the user (Line 7), and finally, the status result of the creation is returned to the user (lines 8-9).

Algorithm 1 User Identity Creation

Input:
Wallet wId, userData $udata$, Timestamp $time$, EndpointDid Eid
Output:
$status$;
1: Id←stroe.getID(wId)
2: uId←uuid(wId)
3: **if** (Id==uId) **then**
4: $schema$←schema.method(Id, $udata$, $time$, Eid)
5: $proof$←getGovIdCredProof(Id)
6: $credDefRequest$←buildCredDefRequest($schema$, Eid, $proof$)
7: submitRequest($credDefRequest$)
8: $status$←$success$
9: **end if**
10: **return** $status$;

Algorithm 2 outlines the process for linking identities using the function. When a user initiates a connection using their created identity, they first request the identity of the other party. Before sending a package of information, the accuracy of the user's ID and identity certificate is verified (lines 1-5), including a nonce for a one-time authentication protocol to prevent replay attacks, relevant identity details, and a record of the connection time (lines 6-8) to the other party. The process outcome is then returned (line 9).

Algorithm 2 Identity Linking Request

Input:
Wallet wId, userData $udata$, Timestamp $time$, theirEndpointDid $tEid$, submitRequest $credDef$, requestNonce $noce$
Output:
$result$;
1: Id←stroe.getID(wId)
2: uId←uuid(wId)
3: $Pairwise$←createPairwise(Id, $tEid$, uId)
4: $createCredDef$←createCredentialsDef($credDef$)
5: **if** ($Pairwise$ && $createCredDef$) **then**
6: $connRep$←connectionResponse(Id, uId, $createCredDef$, $noce$)
7: msg←getMessage($connRep$, $udata$, $time$)
8: $result$←sendMsg($tEid$, msg)
9: **end if**
10: **return** $result$;

The steps for responding to an identity-linking request using the function are described in Algorithm 3. The responder first checks the authenticity of the identity information received (lines 1-5) and then confirms if the nonces from both parties match (lines 6-7).

Algorithm 3 Response to Identity Linking Request

Input:
Wallet wId, userData $udata$, theirEndpointDid $tEid$, getMessage msg, submitRequest $credDef$

Output:
$result$;

1: Id←stroe.getID(wId)
2: uId←uuid(wId)
3: $Pairwise$←createPairwise(Id, $tEid$, uId)
4: $createCredDef$←createCredentialsDef($credDef$)
5: **if** ($Pairwise$ && $createCredDef$) **then**
6: $relationship$←indy.store.pendingRelationship.getAll()
7: **if** ($relationship.noce$==$msg.noce$) **then**
8: $result$←sendAcknowledgement($relationship.Id$, $tEid$, $msg.data$)
9: **end if**
10: **end if**
11: **return** $result$;

Upon successful verification of the identity and data, the connection with the other party is established and the outcome is returned (line 8).

In the blockchain, the identity verification function involves other nodes or participants to verify the participant. The process, as outlined in Algorithm 4, begins by confirming the validity of the verification method and the certificate (lines 1-5). The data of authentication of both parties are then compared to ensure they match and are true (line 6). The verification result is then returned (line 7).

Algorithm 4 Identity Verification

Input:
Wallet wId, theirEndpointDid $tEid$, sendAcknowledgement $sAcknow$, aendMessage msg, submitRequest $credDef$

Output:
$status$;

1: Id←stroe.getID(wId)
2: uId←uuid(wId)
3: $Pairwise$←createPairwise(Id, $tEid$, uId)
4: $createCredDef$←createCredentialsDef($credDef$)
5: **if** ($Pairwise$ && $createCredDef$) **then**
6: **if** ($sAcknow$ && msg) **then**
7: $status$←$success$
8: **end if**
9: **end if**
10: **return** $status$;

In this study, we proposed a roaming identity management system using Hyperledger Indy. The system utilizes VCM to create identities so that the identity holder can self-manage. Each node has an agent to store and manage multiple identities. The identity holder can control the disclosure of information. All transactions and identity information are recorded on the blockchain. The process for implementing the roaming identity management system is outlined in four steps (Figure 2) presented here.

1. All roaming service providers (HPMN and VPMN) sign agreements for roaming identity regulations and roaming cooperation. These agreements are encrypted and transferred in JSON format, incorporating DID, and recorded on the Indy blockchain network node.
2. Before subscribing to roaming services, user Alice first verifies her identity by storing her government-established identity in an Agent and requesting her ID and other information from the government. This data or information is encrypted and transferred in JSON format, incorporating DID, for verification before subscribing to roaming services.
3. Alice can then proceed to query the desired roaming services. Alice and VPMN establish an identity relationship, in which Alice discloses certain information from the government's ID and information to VPMN. The user can query service offerings by the VPMN, and VPMN then verifies the user's subscription to roaming services.
4. Finally, VPMN queries the user's information from HPMN and verifies the accuracy of the information disclosed by Alice. If accurate, the roaming service for Alice is activated, and the service rate is sent to HPMN for roaming service billing.

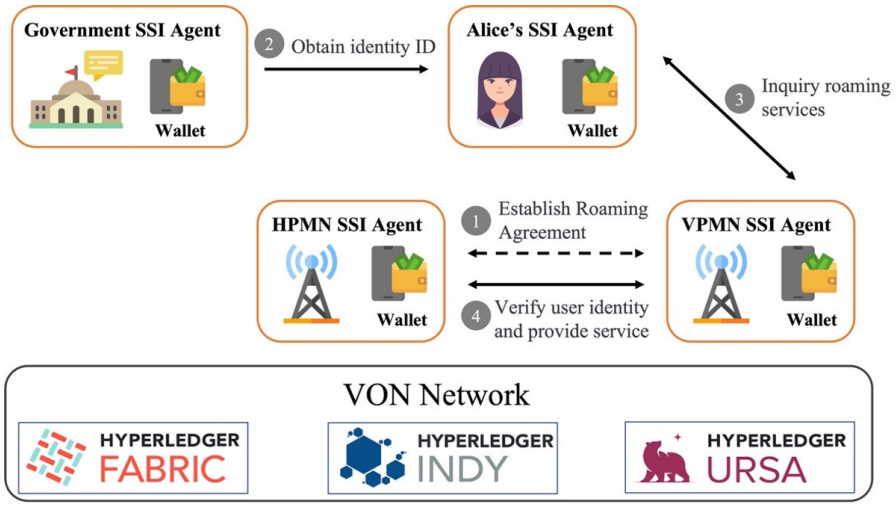

FIGURE 2. Hyperledger Indy roaming system architecture

4. **Experimental Result.** The environment proposed in this study for implementing a roaming identity management system using Hyperledger Indy was mainly configured with an Intel(R) Core(TM) i7-9700 CPU and DDR4-2666 1333MHz with 16GB of memory. The operating system used was Ubuntu 20.04. The software specifications used in the architecture system are listed in Table 1. A general blockchain performance testing framework called Hyperledger Caliper is used to evaluate the performance of the system under different load conditions, measure the relevant data changes, and evaluate the system behavior with the increase in the number of nodes and transaction requests.

This study utilized a Hyperledger Fabric-based test environment for roaming services, as depicted in Figure 3. A Channel comprising two organizations, Org1 and Org2, was created. Org1 comprises four peer nodes representing the government, VPMN, HPMN, and the user's Agent. Org2 includes three OSNs that order transactions verified by each node. The aim is to establish identities and access data between the government and users, order transactions for the access of the user to suppliers, and order cooperation agreements and transaction data between roaming service providers, VPMN, and HPMN.

TABLE 1. A list of software used in the experimental environment

Software	Version
Ubuntu	20.04
Hyperledger Fabric	2.1
Hyperledger Indy	1.16.0
Hyperledger Ursa	0.3.6
Node.js	10.20.0
Docker	19.0.14

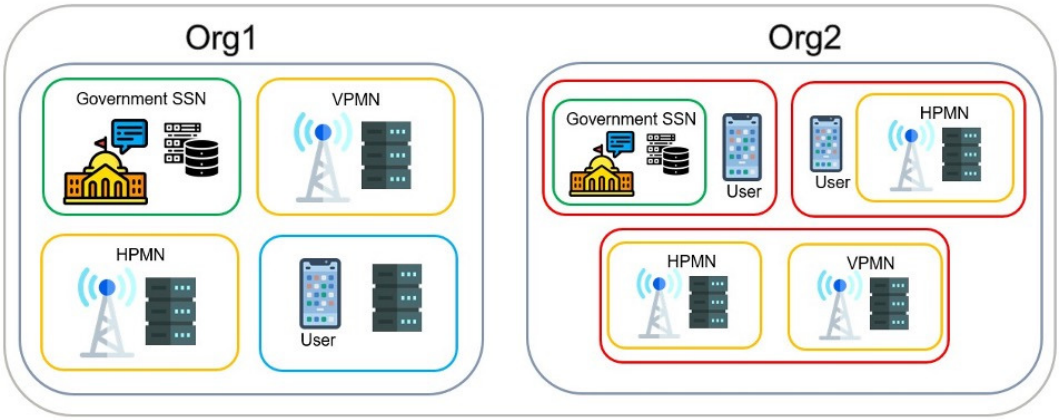

FIGURE 3. Example of Hyperledger Fabric roaming service test environment

Hyperledger Caliper, a blockchain performance testing tool, has a key feature that enables users to conduct performance tests on specific blockchain applications [22]. The monitored metrics include TPS, transaction latency, and resource consumption. Using this tool, the data transmission volume of the roaming scenario was evaluated in this study. The performance of each Chaincode function in the four steps of the roaming identity management system was tested in a blockchain operating environment. Two hundred transactions were sent per second for 30s with varying numbers of nodes. The results are illustrated in Figure 4. When the number of clients began at five and gradually increased, its throughput also increased to 15 clients, reaching a throughput of approximately 182 TPS and around 11000 transactions per min. As the number of clients increased from 20 to 30, its performance decreased to 70 TPS. Moreover, as the consensus protocol in the blockchain, which verifies transactions and stores the same transaction events, increased with the number of clients, the time spent managing and verifying the same identity also increased.

Max Latency(s) is the time delay after a successful transaction deployment, as shown in Figure 5. As the number of clients increases, the processing of transaction requests, verification and deployment, and the recording of the same transactions stored in the blockchain nodes become more complex. In addition, the nodes in the consensus processing also become more complex, resulting in an increase in the performance delay time. The maximum delay time significantly increases when there are 20 to 30 nodes.

This application uses Ethereum blockchain technology to address roaming fraud and data transmission issues, writes supplier and user access supplier events into smart contracts and stores them on the Ethereum network. Performance tests are conducted using CPU and Virtual Terminal configurations, with each Virtual Terminal participating in blockchain mining and acting as a node on the blockchain. The configuration of one CPU

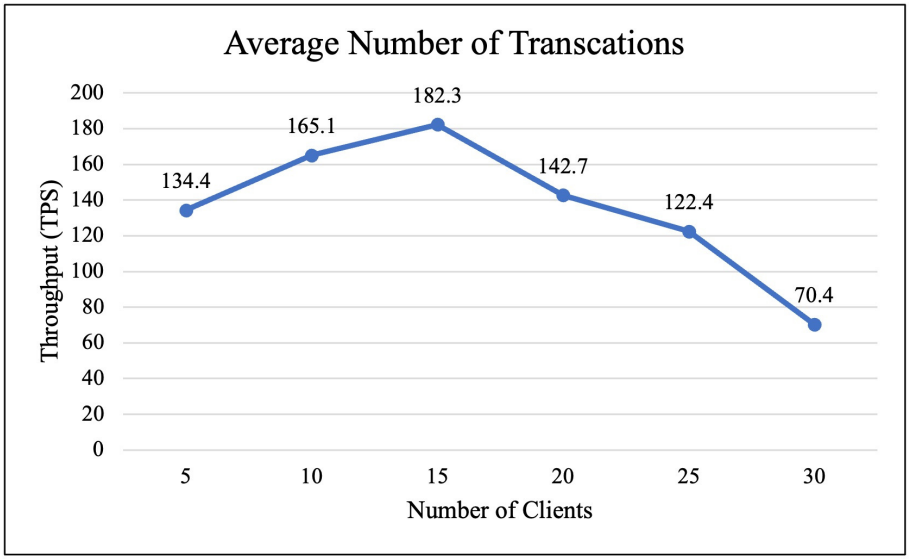

FIGURE 4. Throughput (TPS) Testing

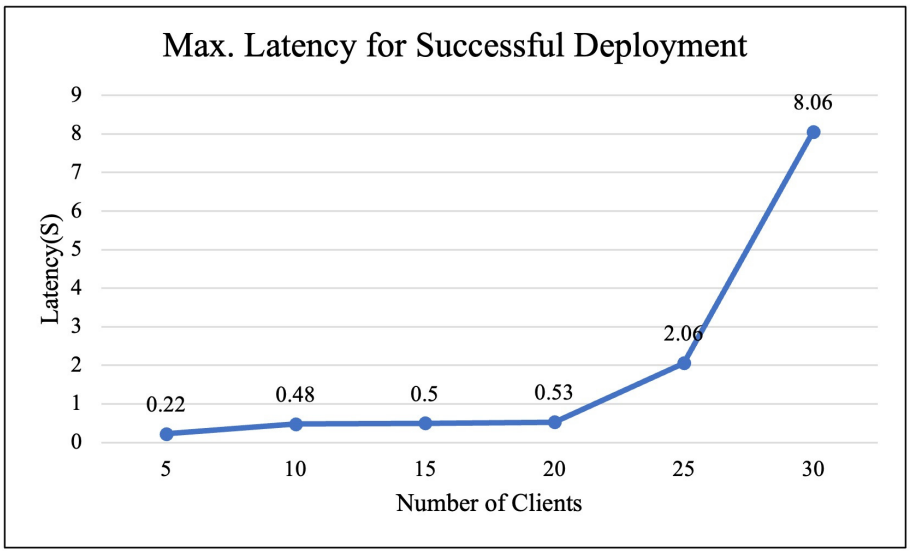

FIGURE 5. The maximum delay time for successful deployment of the trading department

and eight Virtual Terminals demonstrated the best performance and was able to process transactions at an average of 54 TPS, as shown in Figure 6. Further, using the same configuration, the performance test was able to process transactions at an average of around 150 TPS, which is about three times more than the Ethereum method.

5. **Conclusions.** The efficiency of traditional roaming system architecture is low in managing roaming transmission and user fraud problems and can be addressed by utilizing Ethereum blockchain technology. Roaming service agreements and user personal data are written into smart contracts and stored on the Ethereum blockchain platform. Digital wallet spending cannot be maliciously or arbitrarily used, while Ethereum virtual machines do not have the risk of logical bomb problems and can control each function of the contract more effectively. However, since smart contract design is relatively open-ended, designers may be more susceptible to malicious attacks looking for loopholes in roaming

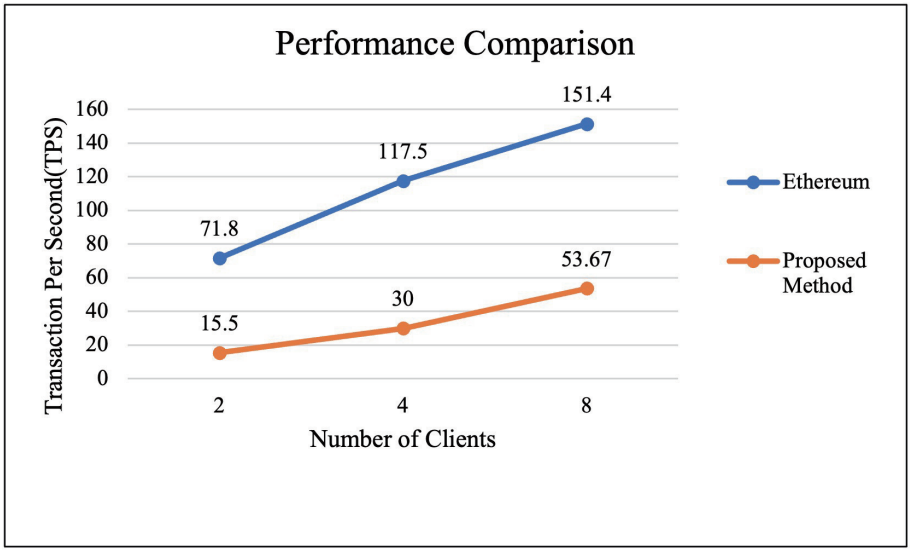

FIGURE 6. Comparing the performance of Ethereum and the proposed solution

service contracts caused by a lack of design experience or familiarity with smart contract language, or may inadvertently leak user personal identity information, resulting in economic losses. Compared to the designed system architecture and its use in this study, the efficiency of verifying transactions on Ethereum nodes was relatively low.

This study proposed a roaming identity management framework and process to address issues of roaming fraud and data exchange management efficiency, without the need for supervision and control from a third-party organization. The proposed framework resolves the issues associated with centralized frameworks and allows users to freely control their own identity to achieve Self-Sovereign Identity (SSI) autonomy. Herein, the use of Hyperledger Indy blockchain technology and framework tools was suggested to address the problems of roaming services in DCH and low efficiency in data exchange and roaming fraud. The proposed roaming identity management process was realized by utilizing a module based on VCM, thus enabling the system to manage roaming identities, allowing users to freely create and connect identities with any provider or other user in the roaming scene, and control and manage the created identities. The contributions of this paper include: 1. The Hyperledger Indy blockchain mechanism was utilized and the roaming identity management process established in this study, the traditional roaming data exchange efficiency was improved, and blockchain characteristics were utilized to prevent users from roaming fraud to the provider. 2. In the roaming identity management process, users can create identities on their own with any provider or other users and can flexibly use and manage their identities without the control and management of a third-party organization. 3. The mechanism and tools proposed and used in this study could improve the transaction processing performance for roaming problems by about three times, compared to the solution provided by the Ethereum blockchain.

REFERENCES

[1] P. Belton, "GSMA: 1 billion more mobile Internet users by 2025," *Light Reading*, 2021.
[2] T.-Y Wu, Q. Meng, L. Yang, X. Guo, and S. Kumari, "A provably secure lightweight authentication protocol in mobile edge computing environments," *The Journal of Supercomputing,* vol. 78, pp. 13893–13914, 2022.

[3] C.T. Nguyen, D.N. Nguyen, D.T. Hoang, H.A. Pham, N.H. Tuong, Y. Xiao, and E. Dutkiewicz, "BlockRoam: Blockchain-Based Roaming Management System for Future Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 3880–3894, 2022.

[4] G. Macia-Fernandez, P. Garcia-Teodoro, and J. Diaz-Verdejo, "Fraud in roaming scenarios: an overview," *IEEE Wireless Communications*, vol. 16, no. 6, pp. 88–94, 2009.

[5] J. Shawe-Taylor, K. Howker, and P. Burge, "Detection of fraud in mobile telecommunications," *Information Security Technical Report* vol. 4, no. 1, pp. 16–28, 1999.

[6] J. Chen, H. Xiao, M. Hu, and C.-M. Chen, "A blockchain-based signature exchange protocol for metaverse," *Future Generation Computer Systems,* vol. 142, pp. 237–247, 2023.

[7] S. Rouhani, and R. Deters, "Performance analysis of Ethereum transactions in private blockchain," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS).* IEEE, 2017, pp. 70–74.

[8] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts ," in *International Conference on Principles of Security and Trust.* 2017, pp. 164–186.

[9] M. Valenta, and P. Sandner, "Comparison of Ethereum, Hyperledger Fabric and Corda," *Frankfurt School Blockchain Center* vol. 8, pp. 1–8, 2017.

[10] Q. Zhu, S.W. Loke, R. Trujillo-Rasua, F. Jiang, and Y. Xiang, "Applications of Distributed Ledger Technologies to the Internet of Things: A Survey," *ACM Computing Surveys* vol. 52, issue 6, pp. 1-34, 2020.

[11] Z. Zheng, S. Xie, H.N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services* vol. 14, no. 4, pp. 352-375, 2018.

[12] S.S. Al-Riyami, and K.G. Paterson, "Certificateless Public Key Cryptography," in *International Conference on the Theory and Application of Cryptology and Information Security.* 2003, pp. 452–473.

[13] C.-M. Chen, X. Deng, S. Kumar, S. Kumari, and SK Islam, "Blockchain-based medical data sharing schedule guaranteeing security of individual entities," *Journal of Ambient Intelligence and Humanized Computing,* pp. 1–10, 2021.

[14] D. Vujicic, D. Jagodic, and S. Randic, "Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview," in *2018 17th International Symposium Infoteh-jahorina (infoteh).* 2018, pp. 1–6.

[15] L. Luu, D.H. Chu, H. Olickel, P. Saxena and A. Hobor, "Making Smart Contracts Smarter," in *The 2016 ACM SIGSAC Conference on Computer and Communications Security.* 2016, pp. 254–269.

[16] C. Cachin, "Architecture of the Hyperledger Blockchain Fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers.* vol. 310, no. 4, 2016, pp. 1–4.

[17] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, and S. Chatterjee, "Performance Characterization of Hyperledger Fabric," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT).* 2018, pp. 65–74.

[18] S.Y. Lim, P.T. Fotsing, A. Almasri, O. Musa, M.L.M. Kiah, T.F. Ang, and R. Ismail, "Blockchain technology the identity management and authentication service disruptor: a survey," *International Journal on Advanced Science, Engineering and Information Technology* vol. 8, no. 4-2, pp. 1735–1745, 2018.

[19] W. Li, H. Guo, M. Nejad, and C.C. Shen, "Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach," *IEEE Access,* vol. 8, pp. 181733–181743, 2020.

[20] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen, "Decentralized Identifiers (DIDs) v1.0–core architecture, data model, and representations," *W3C Community Group,* 2022.

[21] Z.A. Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-Ledger-based authentication with decentralized identifiers and verifiable credentials," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS).* 2020, pp. 71–78.

[22] H. Sukhwani, N. Wang, K.S. Trivedi, and A. Rindos, "Performance modeling of Hyperledger Fabric (Permissioned Blockchain Network)," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)* IEEE, 2018, pp. 1–8.