electronics

MDPI

Article

Interoperable Multi-Blockchain Platform Based on Integrated REST APIs for Reliable Tourism Management

Linchao Zhang [1], Lei Hang [2], Wenquan Jin [3] and Dohyeun Kim [1,*]

[1] Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; zhanglinchao@jejunu.ac.kr
[2] Tianhua College, Shanghai Normal University, Shanghai 201815, China; hanglei@jejunu.ac.kr
[3] Big Data Research Center, Jeju National University, Jeju 63243, Korea; wenquan.jin@jejunu.ac.kr
* Correspondence: kimdh@jejunu.ac.kr; Tel.: +82-64-754-3658

Abstract: The tourism industry can significantly benefit from the blockchain since its implementation can build trust among stakeholders and improve customer satisfaction. However, most of the existing tourism-specified blockchain platforms are single-chains that provide business support for enterprises without guaranteeing transaction information privacy. Besides, these platforms are specified to a single use case and lack interoperability with other platforms to support heterogenous tourism services. This paper aims to address this issue by introducing a multi-chain architecture that utilizes multiple blockchains to enhance processing capability and provide various business services for the tourism industry. The proposed multi-chain architecture improves the interoperability between the activities in different chains by providing functional requirements in practical applications and supports the inter-ledger application. In addition, the private blockchain will be made available to allow users to access the network through central authorization. It also increases the transaction processing capability by distributing multiple tasks across the chains for large-scale applications. To demonstrate the usability and efficiency of the developed approach, a case study on hotel booking is conducted using the blockchain frameworks Winding Tree and Hyperledger Fabric. A comprehensive evaluation experiment is conducted, and the results show the significance of the proposed system.

Keywords: blockchain; multi-chain; tourism; router; hash

Citation: Zhang, L.; Hang, L.; Jin, W.; Kim, D. Interoperable Multi-Blockchain Platform Based on Integrated REST APIs for Reliable Tourism Management. *Electronics* **2021**, *10*, 2990. https://doi.org/10.3390/electronics10232990

Academic Editor: Juan M. Corchado

Received: 18 October 2021
Accepted: 27 November 2021
Published: 1 December 2021

1. Introduction

The tourism industry has a high impact on economic development and job opportunity creations all over the world. In 2018, the tourism sector's total contribution had accounted for 10.4% of global Gross Domestic Product (GDP) and 10% of total employment [1]. With the advance of the Internet, technical innovation kept coming out in the tourism industry and migrated conventional tourism services from offline to online. Consequently, online marketplaces like Airbnb and Uber act as brokers to receive commissions from each booking and offer customers lodging, homestays, and tourism experiences. The tourism industry points to multiple industries since it needs to combine financial services, communication technology, and business knowledge to construct innovative and cost-effective service platforms [2].

Many recent blockchain studies have pointed out the emergence of blockchain applications in large-scale business cases, such as the tourism industry [3–5]. Blockchain [6] was proposed by Satoshi Nakamoto and is also known as decentralized ledger technology. Each network member owns a copy of a shared ledger where all content is tamper-proof through a digital signature. In the beginning, blockchain technology was used primarily for financial-related services such as banking and insurance due to its decentralized nature to guarantee the security and efficiency of financial transactions [7]. However, as the development of the technology becomes more robust, blockchain is not limited to financial services, and many use cases have been gradually explored in various application areas, such as Internet of Things (IoT) [8], healthcare [9], and sharing economy applications [10].

Most IoT systems are developed based on the centralized architecture that brings various assailable problems in security and privacy [11]. The blockchain enables the IoT system to be aware of the inappropriate manipulation of data [12]. To ensure the safety of drugs, preventing counterfeits from entering the authentic supply chain is essential [13]. The blockchain-based system can prevent drugs from passing through different complex distributed networks based on the blockchain network's secure and scalable data sharing characteristics [14]. The distributed governance model can be built on blockchain for decentralized and distribution organizations [15]. Blockchain technology has a slow but growing influence on the tourism industry in several research positions [16].

Many tourism and hospitality companies have realized the potential benefits of blockchain and have applied the technology at their workplace. For example, blockchain technology can be used to track hotel guests by instantly updating right from when a guest leaves home for the airport to upon arrival at the hotel. The guest's authorization is required to access the whereabouts info, and stakeholders can determine the information to share with hotels or other participants within the network. Besides, business relationships of hotels and travel agencies can be automated by smart contracts. Anytime a transaction occurs, the smart contract enforces the obligation on specified resources in terms of predefined. Once a transaction is confirmed, the payments can be processed immediately according to the agreement content. The use of smart contracts can facilitate payment and optimize room sales through better collaboration between hotels and travel agencies. Many companies in the tourism industry have investigated lots of efforts on blockchain's bandwagon effect to create better and more productive user experiences. Travelport [17], Populstay [18], and Travala [19] are some of the recent blockchain implementations that have been put to use already in the tourism industry.

As far as we know, most of the existing blockchain-based tourism platforms are built on a single-chain. Such a structure cannot meet the requirements of the growing tourism market, as it does not provide practical options when application services need to operate multiple closely-connected blockchain ledgers according to the business logic of the services. When handling a large volume of transactions, a single-chain can lead to transaction delays and network crashes. As one of the most popular blockchain platforms, Ethereum can only process a dozen transactions in 1 s, and it takes about 15 s to process a transaction. Applications built on single-chains are inefficient for enterprise applications, generating millions of transactions in 1 s. There is a large amount of business and privacy information from markets and customers, especially in the tourism industry [20]. As discussed by Gartner in their report [21] on blockchain platforms, interoperability is one of the emerging issues across blockchains. For example, a user may want to execute payment transactions on one blockchain over another based on which blockchain has low network congestion at that time.

This article proposes a multi-chain architecture that provides a secure and efficient solution for the tourism industry to solve the above problems. The main advantages of the proposed solution can be summarized as follows:

- Improve interoperability: Single-chain networks cannot meet functional requirements in practical applications and are not suitable for an inter-ledger application. The proposed multi-chain architecture enables the interoperability between the activities in different chains.
- Provide data privacy: Single-chain is usually a public blockchain network where any member can participate without central authorization. The multi-chain architecture supports the private blockchain that only allows authorized users to access or perform operations on the blockchain.
- Increase transaction processing capability: The single-chain architecture has limited performance and cannot meet the needs of large-scale applications. The multi-chain architecture contains multiple chains in which the various tasks are distributed appropriately to improve processing efficiency.

The contributions provided by this paper can be summarized in a three-fold structure. First, we propose a multi-chain-based architecture to enhance the transaction processing ability for tourism services. As the name implies, it contains multiple chains that are independent of each other. Each chain performs its business logic and stores the related data separately. Second, a case study of a hotel reservation is implemented to demonstrate the usability and efficiency of the designed solution, containing a public chain for room management and a private chain for trading. Winding Tree [22] is a public chain that allows the interaction between customers with service providers such as airlines, hotels, and tour guides.

Meanwhile, a private chain based on Hyperledger Fabric [23] is used to process hotel orders. External applications can call and operate the specified blockchain through various APIs provided by the corresponding blockchain network. Third, a comprehensive evaluation experiment is conducted with multiple performance indices. A detailed benchmark analysis compares the proposed system with existing studies to demonstrate the significance of the developed solution.

The remainder of this paper is structured as follows: Section 2 reviews some recent blockchain implementations in the tourism industry. Section 3 presents the designed multi-chain architecture and describes the business process that occurs in the system. Section 4 elaborates on the case study implementation for the hotel reservation. Section 4.3 represents the implementation results through various screenshots. Section 4.4 evaluates the performance of the proposed approach. Section 5 attests to the significance of the proposed system through a comprehensive benchmark analysis. Section 6 summarizes the whole paper and points out the future research direction.

2. Related Work

The tourism industry is an information compound, which lies in diverse service providers sharing data. For instance, Online Travel Agents (OTA) have to post customer details to hotels and airlines, and the booking data should only be transparent and accessible to associated providers. Blockchain allows it to access and store crucial information straightforwardly and reliably because it is stored in a distributed ledger shared across the network [24–26]. The adoption of blockchain in the tourism industry provides a seamless experience in the way travelers can directly trade with service providers to book hotel rooms and tickets without third-party intervention. Blockchain has raised significant interest in the tourism industry, and many major companies have incorporated blockchain technology in their services [27]. As a network-based system, blockchain is used to develop secure, intelligent, and transparent distributed ledgers in the tourism industry through new tools such as smart contracts, decentralized applications, and cryptocurrencies. Ozdemir et al. [28] introduce blockchain basics criteria including models, platforms, type of consensus, cryptocurrency, smart contract, and tokens for the tourism solutions that can be considered to develop blockchain-based distributed applications in the tourism industry. At the same time, the characteristics and advantages and disadvantages of related tourism blockchains are compared in Table 1.

Table 1. Compare the technology and characteristics with the existing tourism blockchain.

Type \ Chainname	Multi-Chain	Rezchain	ZatGo	Travelchain
Technology	Combination of the main chain and sub-chain	Share hotel inventory and data information	Use the alliance chain to build a business travel platform and a bidding platform	Public open-source blockchain under the management of the EcoSystem's users
Transaction	Self-issued token	No token	Blockchain payment unified platform (ZUP Token)	TravelToken
Disadvantage	High throughput, low latency	Low throughput	High latency, low throughput	Low latency, low throughput

Viachaslau et al. [4] presented the knowledge of blockchain technology from a business perspective to emphasize the challenges of commercial prompter uptake. Nam et al. [29] introduced the key characteristics of blockchain technology and how this technology would evolve and affect the tourism industry. They identified three categories by analyzing recent tourism DApps: reducing costs, adopting cryptocurrencies, and developing eco-systems. Tripio [30] is another decentralized tourism marketplace using blockchain technology. It aims to provide direct linking between consumers and travel service providers within the blockchain network. New incentive and credit mechanisms are introduced to reduce commissions and rewards consumers with high credit ratings. DeskBell [31] aims to create a standard informational system for business clients and tourism and hotel industry users based on flexible monetization mechanisms. The system also allows all participants to distribute and exchange services, offers, and events. ZatGo [32] introduces blockchain technology and token payment scenario mode. The business logic of ZatGo is specified by the smart contract, which can perform various operations on the distributed ledger, including digital identity authentication, payment, risk control, and credit information. Users can query all recorded data in real-time through client access and be rewarded if they contribute to the state.

Travel Chain [33] is a decentralized data exchange platform for the travel market where users enter their personal information and receive rewards for it. All users involved in the exchange of information have equal access to control the personal information they enter. BloHosT [34] is a blockchain-based framework that provides tourists with a single wallet to initiate payments with various stakeholders in a straightforward manner. Once the ledger confirms the data, it cannot be tampered with at will. Besides, it provides spot recommendations to prospective travelers according to rating scores from the experience of previous travelers. Other applications like Webjet [35], FlightDelay [36], and Cool Cousins [37], as outlined in [38], are also used extensively. All these applications, to the best of our knowledge, are built on a single-chain. These applications are not in full service yet, as blockchain in tourism is still in its infancy. This paper addresses these issues by introducing a multi-chain architecture to enable the inner integration between different blockchain implementations to provide miscellaneous tourism services.

The distributed ledger technology allows users to have identical copies in the blockchain network, enabling transparency, equity, and accountability in transactions [39]. Also, the collaboration of IoT and blockchain enables significant transformations across several industries for providing new business models [40]. Wood et al. [20] proposed the Polkadot framework that provides meaningful improvements in scalability, isolation, developability, governance, and applicability through multiple chains in the blockchain network. Kan et al. [41] proposed a component-based framework for exchanging information across an arbitrary blockchain system by presenting an inter-blockchain connection model to manage routings and transferring messages in private multiple blockchain systems. Hwang et al. [42] proposed InfiniteChain, a multi-chain architecture, to solve the problems of inadequate transaction bandwidth, excessive data volumes, and the lack of privacy protection in conventional blockchains.

Consortium chain, which is composed of multiple private chains. Usually, a blockchain that numerous institutions jointly manage and the number of nodes is limited. This leads to incomplete decentralization. As long as most institutions (nodes) reach a consensus, the blockchain data can be changed. The multi-chain architecture can guarantee that user access rights are restricted, and that data cannot be modified. We describe the difference between multi-chain and consortium chains from five perspectives:

- Design goals

The blockchain system is a very typical distributed system. The multi-chain adopts a public chain + private chain architecture. The public chain (main-chain) is responsible for public data, completely decentralized, and the private chain (sub-chain) is responsible for accounting, wholly privatized. The transaction method needs to be cross-linked. The chain operation completes the transaction. The consortium chain is mainly a blockchain that

multiple institutions jointly manage. Each organization or institution contains one or more nodes of the identical blockchain. To the extent of regional centralization, the multi-chain and alliance chain is semi-centralized, but the underlying technical architecture of the two is different. The alliance chain is a collection of multiple private chains and adopts a multi-center technical architecture. When a new node is added, it needs to be verified and reviewed. The sub-main has only one center, all operations are controlled by the authority of the center (query transactions), and there is no need to consider the addition and exit of new nodes.

- Access method

According to the division of functional modules, multi-chain has different restrictions on participants. Main-chain does not have any restrictions on participants. Account addresses are generated through non-conversion encryption algorithms and hash algorithms, allowing anonymous participation in chain activities. Successfully authenticated encrypted CA digital certificates are issued (the encryption method will be explained in detail below). The private chain (sub-chain) uses the CA for identity identification and identification to achieve the permission control access method. The consortium chain usually has a fixed number of nodes in a committee composed of start-up members of the entire consortium and enters the nodes through multiple votes.

- Consensus Algorithm

The blockchain consensus algorithm has two meanings; one is data consensus, and the other is business consensus. Data consensus indicates what kind of software and hardware algorithms are used between nodes to reach the agreement of the ledgers between nodes. This mainly refers to the consistency of the transaction sequence because whether it is a consortium chain, a public chain, or a private chain, the transaction has the digital signature of the initiator. It is almost impossible to be tampered with and at most discarded. Business consensus refers to which parties should endorse and guarantee the business meaning represented by the data on the chain to ensure the authenticity of the business data. The public chain (main-chain) uses data consensus in the multi-chain architecture and completes data broadcasting through the DPOS consensus algorithm. The main disadvantage of the DPOS algorithm is that the accounting nodes are relatively reduced, blocking transaction speed. The main disadvantage of using Block Producer (BP) is due to the small number of BPs, losing some decentralization. The private chain (sub-chain) uses the PBFT consensus algorithm. The advantages are high efficiency, high fault tolerance, no tokens, energy-saving, and environmental protection. The disadvantage is that PBFT is a partially centralized network and is prone to forks. The consensus algorithm of the alliance chain is based on different business scenarios and different degrees of decentralization. Most of the suitable consensus algorithms are selected (RAFT, Kafka). The following Table 2 briefly introduces some consensus algorithms.

- Performance

In the multi-chain architecture, the main-chain and sub-chain architectures selected mainly use the complete decentralization of the public chain and the efficient processing capabilities of the private chain. Combining the characteristics of decentralization and high performance, the business is distributed on different blockchains according to other functions. Cross-chain operations complete the maximum business processing capacity. The alliance chain is configured and structured for different scenarios, and its performance is very high in simple data checks and sequential write performance. If a correlation between transactions and the order significantly impacts the execution results, transaction performance will be reduced considerably.

Table 2. Consensus Algorithm application scenarios and comparison.

Consensus Algorithm	Representative Scene	Algorithm Description	Whether to Tolerate Malicious Nodes	Number of Nodes Participating in a Consensus
PoW	Bitcoin, Ethereum	Whoever contributes (high probability) listens to whom	Yes	Unknown
PoS	Ethereum (NG)	Whoever contributes (high probability) listens to whom	Yes	Unknown
DPoS	EOS	Whoever among the agents has more assets (high probability) listens to whom	Yes	Unknown
PBFT	BCOS, TrustSQL	Propose first before voting	Yes	Known
Paxos	Distributed DB	Propose first before voting	No	Known
RAFT	R3 Corda, Fabric	Election of Leader, Leader is responsible	No	Known
Kafka	Fabric	First in, first out queue	No	Known

3. Proposed Multi-Chain Architecture

3.1. System Architecture

The network performance improves as the main chain can decompose a complex task and assign single tasks to different sub-chains compared to a difficult task in a single-chain. The interworking application layers serve as an intermediate to exchange data between the main chain and other sub-chains. Besides, it verifies the return value from the sub-chain and decides whether to pass the return value to the main chain. For example, when a user sends a login request to the sub-chain using a client, the sub-chain performs the received request and responds to the application layer client. Meanwhile, the application layer checks the validity of the return value and sends the verified value to the main chain. In this way, the data is encrypted and exchanged between the main chain and sub-chains using the interworking application. The information is always encrypted and transmitted through the application layer to verify the data and synchronize the data to ensure security.

Figure 1 represents the proposed multi-chain architecture to address the issues of the single-chain blockchain in actual business. This architecture is designed in the main chain centered with multiple sub-chains diverged around. The main chain is generally a public network, which stores non-sensitive information. Each sub-chain is a private network specified to a particular business use case, running independently from each other. The number of sub-chains can be extended infinitely according to the functional requirements of the business, thus ensuring the scalability of the multi-chain architecture.

The cross-chain communication between multiple blockchain networks is an essential factor in the proposed architecture. The interoperable application layer allows important parameters and synchronization information to be shared between various chains. As shown in Figure 2, the proposed multi-chain architecture consists of five layers: application layer, intercommunication layer, middleware layer, blockchain layer, and basic platform layer.

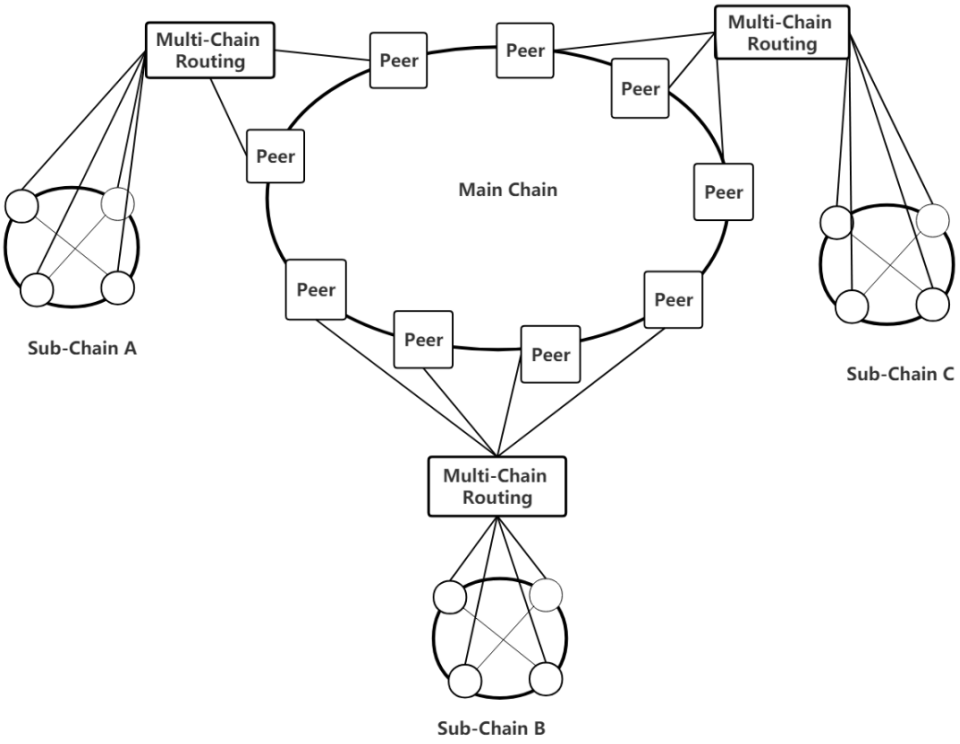

Figure 1. Overview of the multi-chain architecture.

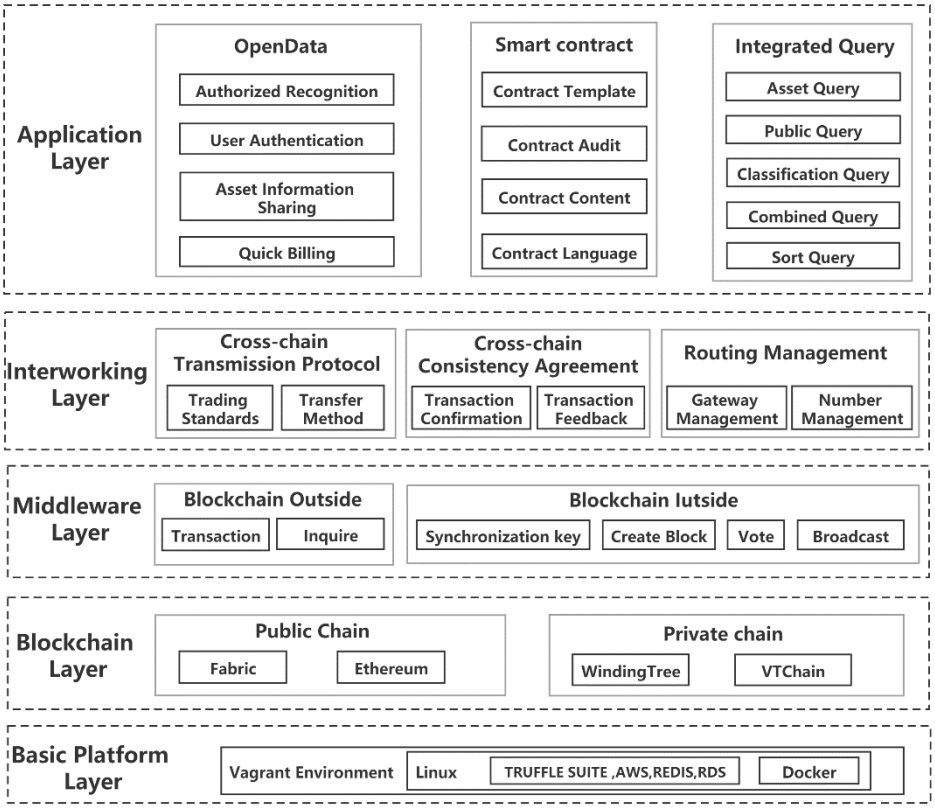

Figure 2. The layered architecture of the multi-chain.

The application layer contains an open data module, an intelligent contract module, and an integrated query module. The available data model includes two sub-modules:

(1) Consent to authorization sub-module used by third parties to make data requests, and users grant permission to third parties to access user data; (2) Authentication sub-module used by data custodians user identity authentication. Therefore, the above modules control the user login identity verification and authorization identity verification, and at the same time communicate relevant asset information to the customer, so that the customer can quickly perform asset transactions and complete the transaction steps. Smart contracts and transaction rules are written into innovative contract modules to control transaction assets, verification of transaction authorization, and complete transaction information passed in Open Data. At the same time, the integrated query provides an asset query. It offers related public information queries (hotel room information, ticket information) and business environment, classified query, combined query, and sort query.

The interworking layer includes the cross-chain transmission protocol used for different transmission standards, the cross-chain consistency protocol for transaction confirmation and transaction feedback, and the routing management for gateway management and numbering management. The cross-chain protocol mainly solves the problem of data consistency in the data exchange process between multiple chains. The transaction interface is used to define a unified transaction standard so that the data consistency is met after the cross-chain transmission. Data transfer is done via the IBC protocol [43] but requires complete verification by a consensus mechanism. The cross-chain consistency protocol guarantees the completion of the data exchange between the two chains during the transaction. The process is divided into three parts: local transactions, cross-chain transactions, and confirmed transactions. When the A chain transfers the transaction to the B chain, the local trade in the routing is first carried out locally. The routing sends the transaction across the chain to the B chain. The B chain also routes the local transaction. After the verification is completed and synchronized, the transaction feedback is generated. A chain accepts transaction feedback to complete cross-chain transmission. In the verification process, the transmission is performed through route management, and the accuracy of route transmission is ensured through the management of gateways and route numbers. This process ensures that the API functional modules interoperate between the main chain and the sub-chain, and realize the module functions (described in detail in Section 4.1, active layer) to transmit data safely and efficiently. Pillai et al. [44] propose communicating with multiple entities' blockchain systems in a distributed fashion (without an intermediary) while maintaining the property of trust and integrity built by individual blockchains. The article effectively solves the problem of cross-chain interoperability using transactions.

The middleware layer is used for transactions and inquiries outside the blockchain and synchronization key, creating blocks, voting, and broadcasting functions inside the blockchain. The essence of blockchain transactions is the network of value transfer. In the process of value transfer, the address information is used to determine the value belongs to. That is, a certain amount of the value is stored in an address (transaction data). If value transfer occurs, the process of transferring value from one address to another will occur. In the transaction of the blockchain system, the address is used to represent the initiator and receiver of the transaction, and the address is represented by the lowercase letter a. A user can have multiple addresses, such as $a1, a2, \ldots an \in Ui$, indicating that user Ui owns $a1, a2, \ldots, an$. The transaction defined in this article includes a transfer transaction similar to the real one and the data that the uppermost application layer business logic sends to the blockchain system to be stored. Therefore, each block will keep a transaction list in the blockchain system according to a specific transaction format. During the transmission process, the security of the data is guaranteed, and the data is hashed and encrypted to generate a string for transmission. When querying, search the first block to obtain query information and then perform a consensus algorithm to ensure the accuracy of the information in the first block.

Inside the blockchain, to adapt to various transaction types in the production environment, if a transaction is represented by T, the proposed approach adds several fields on the original basis of the blockchain. The source address is an address constant representing

the originator of the value or data. The hash algorithm encrypts the address for ordinary transactions or transactions that create smart contracts represented by *Tf*. The destination address is an address constant representing the recipient of the value or data. The hash algorithm encrypts the address.

Ordinary transactions or transactions are created by using smart contracts. This field is represented by the character *Tt*. The timestamp of the transaction initiation, this field is represented by the symbol *Ts*. A 128-bit positive binary integer represents the serial number corresponding to the exchange. The first 64 bits represent the timestamp sent by the transaction, and the middle 32 bits represent the block initiated by the trade. The last 32 bits of the chain system I.D. indicates the number under this blockchain timestamp under timestamp, counting from 0. This field is represented by the symbol *Tn*. The ack number is used to confirm the transaction; the remaining transaction types are empty, used to confirm a particular serial number transaction. Coded as a 128-bit positive binary integer, this field is represented by the symbol *Ta*. The type constant to which the transaction belongs is coded as an 8-bit positive binary integer, and represented by the character *Tp*. Transaction information is used to store transaction-related data. They are designed according to the characteristics of different blockchain systems; there is no fixed size, and the size is not limited. This field is represented by the symbol Td. The corresponding information is used for transaction signatures. Variables v, r, and w are represented by *Tv*, *Tr*, and *Tw*, respectively. Therefore, a transaction T can be expressed as $T = (Tf,Tt,Ts,Tn,Ta,Tp,Td,Tv,Tr,Tw)$.

The blockchain layer supports various blockchain implementations, including Hyperledger Fabric, Ethereum, WindingTree, VTchain, etc. In this article, the public chain uses Fabric to control the shared information and permissions. The private chain uses WindingTree to trade and control the current tourism-related businesses on the market.

The primary platform layer is used to configure the operating environment, using the Vagrant environment to manage virtual machines (VMware, AWS). Its main advantage is providing a configurable, portable and reusable software environment, such as a Linux virtual machine. Truffle Suite Development Kit (Truffle, Ganache, Drizzle) is installed in the virtual machine and the scalable database RDS is used for storing non-core data.

3.2. System Interaction Diagram

In the proposed multi-chain architecture, cross-chain transaction transmission is encapsulated, parsed, and forwarded by the interworking layer. The blockchain system connected to the entire network only needs to implement the interface provided by this layer. The transaction is converted into a standard transaction; it can be connected to a homogeneous and heterogeneous blockchain system. Transactions are transferred across chains on a peer-to-peer basis without using any equipment provided by a third party, ensuring the privacy of the transfer. The interworking layer can prevent cheating; meanwhile, each node can be extended to a cluster to provide external services, including cross-chain transaction success, a cross-chain transaction failure, and cross-chain transaction timeout retransmission.

In cross-chain communication, there is an atomic consistency problem in the communication between different chains and the execution of smart contracts. The following solutions are proposed:

- Difficulties of cross-chain communication

Cross-chain consistency: Cross-chain transactions can only succeed or fail at the same time when cross-chain technology realizes asset transfer and exchange between chains, ensuring that the ledger information of both parties in the inter-chain transactions is updated synchronously and the consistency of cross-chain transactions is maintained.

Transaction verification issues: Cross-chain transaction verification mainly includes two aspects: to confirm that the transaction is executed and successfully written into the blockchain ledger, and that both parties to the cross-chain transaction can verify the legitimacy and validity of the transaction during the cross-chain transaction. However, in the blockchain system, to ensure the absolute reliability of the information, most of the blockchain system is a definite and closed system environment, which makes the data

interaction inside and outside the chain very difficult, thereby increasing verification and other aspects. The difficulty lies in assessing the legality and validity of transactions in a chain. At present, the "block header + SPV" model is a common cross-chain transaction verification mechanism [45].

This paper uses the hash timelock contract method for cross-chain communication to solve the cross-chain consistency problem. This method does not require a trusted notary to complete the exchange of assets between chains without using hash locks and time locks. In the implementation process, the initiator first randomly selects the secret value as the hash decryption key, then hashes the private value and sends the obtained hash value as the public key of the hash lock to the responder, initiator, and response. The participants lock their digital assets in the smart contract by the hash value and set their time lock (usually the time lock of the initiator is greater than the time lock of the responder), such as: A random chain number (unlocking secret key), S hash function (locked public key), $h = hash(S)$, Time lock T_1, Token A Hash lock (locked) h, Failure timeout T_1. If both parties provide the secret value within the specified time, the asset locked in the contract will be exchanged successfully. Otherwise, if either party cannot provide the personal value (hash decryption key) within the specified time, the other party will recover the assets locked in the contract. In the process, the hash lock and time lock methods ensure the consistency of cross-chain transactions (simultaneous success of simultaneous failure).

As shown in Figure 3:

(1) The A chain generates the value S, and at the same time calculates the corresponding hash value h, and passes h to the B chain through the network.
(2) A time lock is set on the A chain, and locks Token A in the smart contract of the A chain through the hash value h.
(3) The B chain sets a time lock, and at the same time uses the h passed from the A chain to lock Token B in the B chain's smart contract.
(4) A chain provides S (unlocked secret value) to B chain within the time range of T_2, while B chain transfers the locked Token B to A chain and obtains S simultaneously. If the time expires, the cross-chain fails, and both parties retrieve the assets in the smart contract.
(5) The B chain provides S (unlocking secret value) to the A chain within the time range of T_1, and the A chain transfers the locked Token A to the B chain. If the timeout expires, the cross-chain will fail, and the two parties will retrieve the assets in the smart contract.
(6) Any chain that does not provide S within the time range specified by the other party's time lock will cause the entire cross-chain asset exchange to fail.

- Difficulties of smart contract cross-chain data interaction

As shown in Figure 4, the information exchange between main-chain and sub-chain smart contracts uses hash time locks to ensure the atomicity of cross-chain transactions. In essence, a smart contract is to achieve a specific condition and execute a function. In this paper, the asynchronous mode is used for data interaction in data cross-chain transfer. In the asynchronous call scenario, a complete process requires three transactions: first, a transaction T_{x1} is sent to the A chain, and when the A chain code is executed to the cross-chain process, an event will be issued, and a callback function will be declared. After subscribing to the event, the middleware (application) initiates the transaction T_{x2} called to the B chain. After the T_{x2} is successfully on the chain, the middleware (application) triggers the callback function to call T_{x3} to complete the entire smart contract cross-chain interaction (data interaction). When using asynchronous mode for data interaction, the fragmentation of business logic is enhanced, and the number of interactions is high. We adopted a modular design in the development process to reduce the number of interactions.

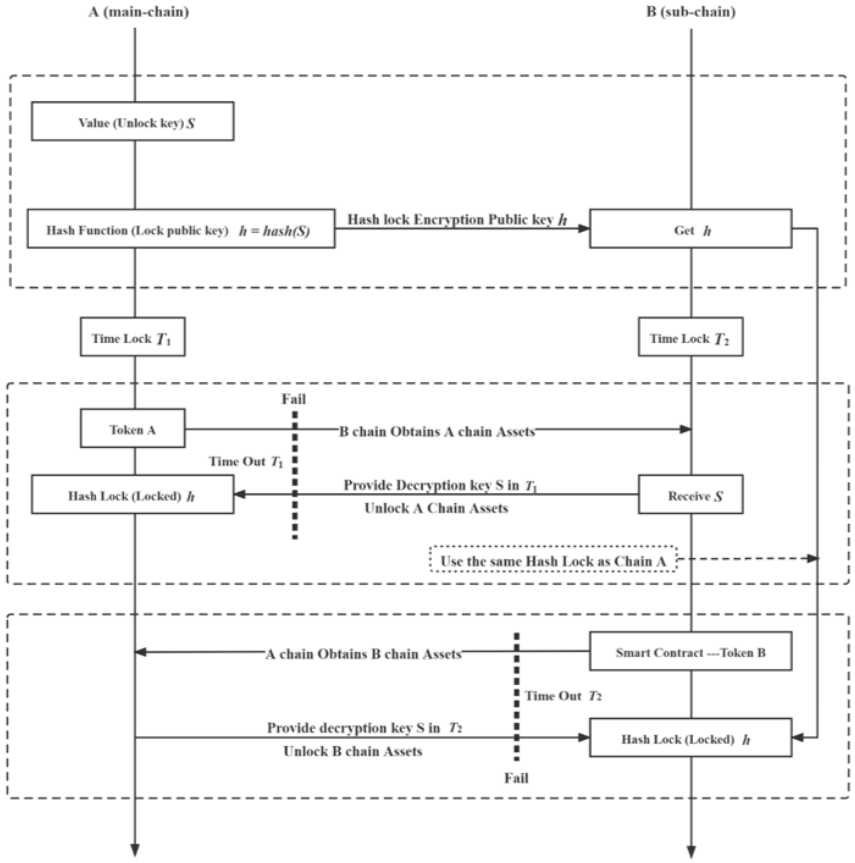

Figure 3. The principle of hash lock and hash time in cross-chain communication.

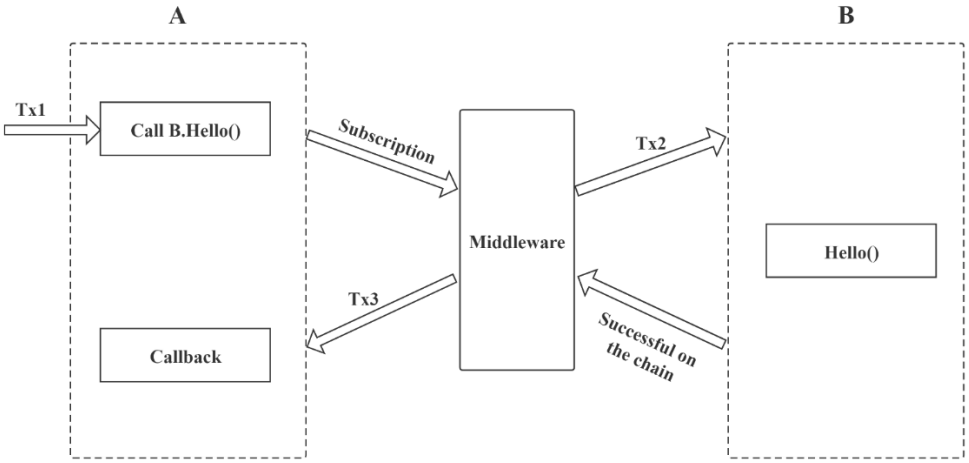

Figure 4. Cross-chain smart contract data information interaction process.

Figure 5 depicts the communication process between the public and private chains through the interworking application. Suppose the address A1 in the public chain system S1 initiates a cross-chain transaction to the address A2 in the private chain system S2. After receiving the cross-chain transaction T', the public chain system S1 enters the pre-preparation stage, locks the relevant assets, and transfers the assets to the temporary address At1 of the interworking application I1. Thus, there is a TRANSFER (A1, At1, value) value transfer function, which means to transfer the asset with value in the address A1 to the address At1, and then send the transaction T' to the interworking application I1. After I1 receives the transaction T', it uses the PACKAGE function to package the local cross-chain transaction

and forwards it to the private chain system S2. It puts the transaction T' into the local transaction cache. After passing the internal consensus, it enters the pre-submission stage and sends the confirmation transaction via the interworking application I1. After interworking application I1 receives the confirmation transaction, it forwards the confirmation information to the public chain system S1 and puts it into its transaction cache, waiting for the internal consensus of the public chain system S1. At this point, after the blockchain system passes the internal consensus, the transaction can be fully submitted.

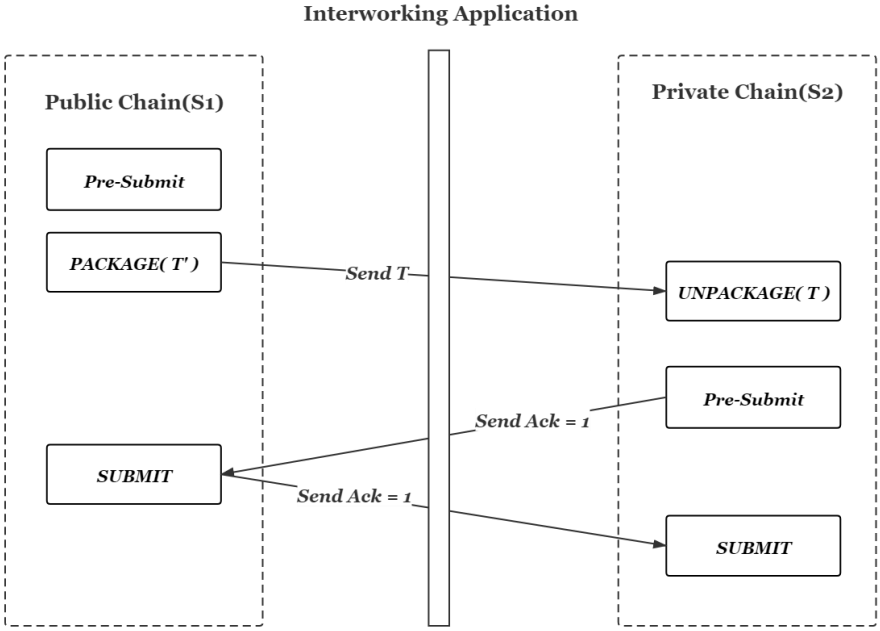

Figure 5. Main chain and sub-chain communication process through interworking application.

3.3. Execution Process of the Proposed System

Figure 6 gives a sample to illustrate various service execution processes that take place in the multi-chain architecture. In the beginning, the client verifies the information from the sub-chain. This step is usually used to authenticate the user at login and confirm the information from the sub-chain. After that, the application queries and searches the business information on the main chain. The sub-chain confirms the legitimacy of the information, and the application receives a confirmation message. The application can perform various business operations on the main chain by invoking associated transactions. The main chain executes the transaction, and the execution results are sent back to the application. The application transmits the execution results to the sub-chain, and these values are recorded accordingly. The trade transaction is executed, and the data transmitted by the application is used for consensus on the main chain. The main chain records part of the transaction information, and the rest is returned as a specific value to trigger the transaction on the sub-chain. After the transaction is recorded on the sub-chain, the result is returned. The application, in turn, sends the return value to the main chain. The transaction process is complete, and the main chain returns the success ack to inform the application that the transaction was successful. This example describes how a transaction is executed in multi-chain architecture. This architecture can solve the lack of privacy control across multiple blockchain networks, different in practical application environments. Moreover, the proposed approach is designed in a modular architecture that supports the extension of varying blockchain implementations concerning business requirements.

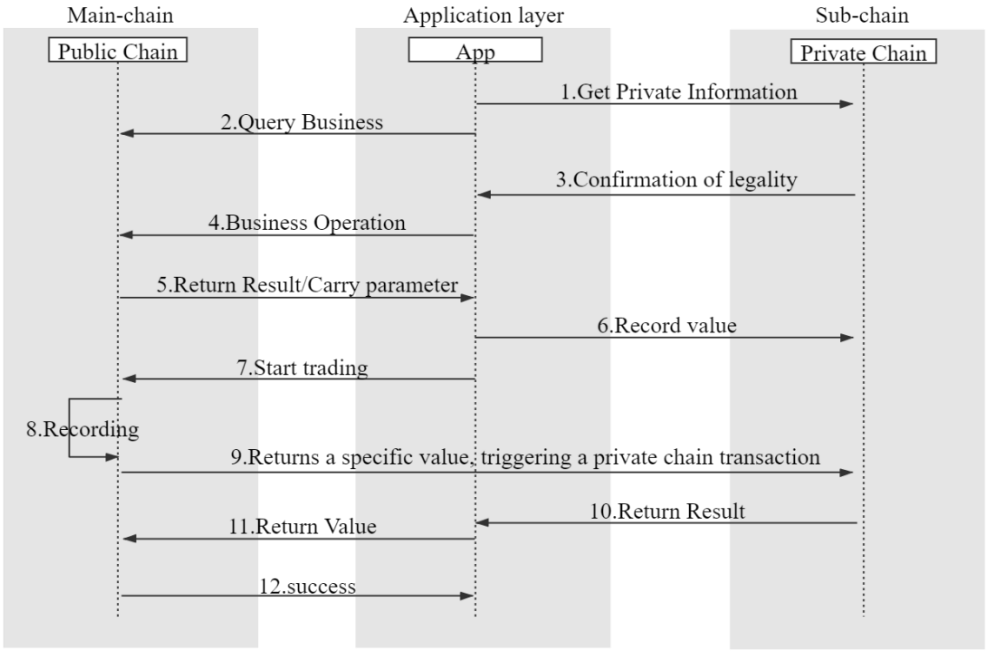

Figure 6. Service execution process in the multi-chain.

4. Case Study Implementation

4.1. Prototype Architecture

As shown in Figure 7, the tourism service prototype built on top of the proposed architecture consists of three main components: an interworking application, the main chain on WindingTree, and a sub-chain on Hyperledger Fabric. The interworking application provides a set of high-level REST APIs to interact with each chain. The main chain is implemented on the Winding Tree network, which stores URLs, indexes, and inventory content in the distributed database. The foundation of the main chain lies in a set of Ethereum smart contracts, which serve as the entry point and directory of hotels. Each hotel links a JSON document stored in off-chain storage such as Swarm and IPFS. The Hyperledger Fabric-based sub-chain is used to deal with the order, including payment and refund. The Certificate Authority (CA) owns user IDs and authenticates clients who enroll in the network. Transactions start with client applications that act on behalf of users to submit transaction proposals to peers through the software development kit (SDK). The Fabric network comprises an orderer and multiple peers that can be either endorser peers or committer peers. The orderer creates a shared communication channel between clients and peers, packages signed transactions into a block, and delivers them to all committer peers. Endorser peers simulate and sign transaction proposals, and respond granting or denying approval while committer peers validate transaction results before writing new blocks to the ledger. At the same time, for this article, using a database in the public chain to store some unimportant data off-chain and storing all information on the chain will be expensive. Off-chain storage dramatically improves the concurrency of the system architecture and processes data through multiple channels.

Figure 8 presents the interaction between end-users and blockchain networks in detail. The proposed system provides various REST APIs for access to platform data in runtime environments. The end-user sends a request to the business network by calling the API, then the network performs corresponding operations and returns the execution result to the client. Read API returns the related data from the off-chain storage. Write API is used to add hotel or airline data to the Winding Tree network. Search API retrieves the data stored in the Winding Tree network and returns the inquiry result to users. The booking manager handles the process related to booking, and it is tied to both the Winding Tree and Fabric network since it performs operations on these two networks. In case of booking a

room, the booking manager initializes a new booking into the Fabric network. The booking process is divided into confirmation of booking information (inventory, price, etc.) and order creation (including payment processors), and the main chain (WindingTree network) is used to update inventory information accordingly. Figure 6 Steps 3 and 4 return the booking result to the booking manager, including generation of reservation information and update of the customer. In the fourth step of Figure 6, Post "Value" sends the generated reservation information to the Order Manager to develop a complete order and pay the fee in the sub-chain (Fabric network), record the order information, and return the order information result (successful or failed request). In this process, the final order time is generated after the order process to ensure the correctness of the order information. Only the sub-chain returns the order result to reduce system resource overhead and improve system performance. The order manager can perform various order-related functions such as adding, updating, and deleting the order.

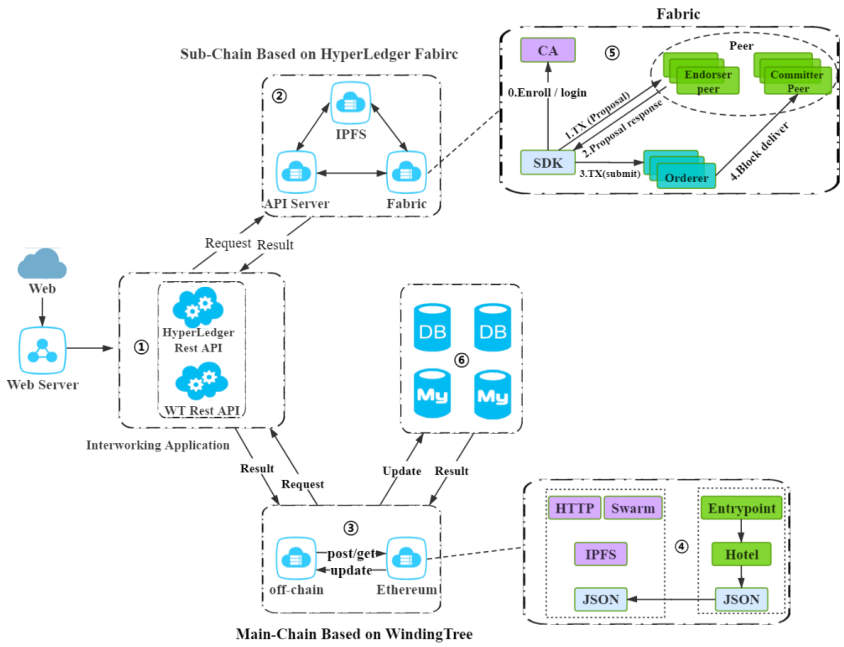

Figure 7. Overview of the multi-chain architecture.

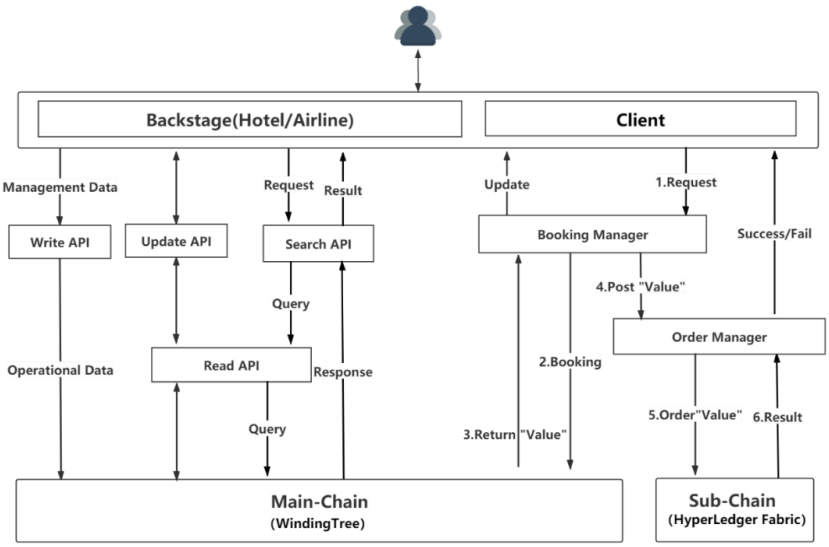

Figure 8. Interworking process based on multi-chain for secure tourism service.

In the transaction process described above, the problem of privacy loss in transactions is solved. Users access public data (hotels, airline tickets) on the WindingTree network (main-chain) and use anonymity and encryption to access and obtain public data in a shared environment. Public data obtained is packaged into HyperLedger Fabric orders and generated on the network (sub-chain). Logging into the HperLedger network requires access control lists (ACL) [46] and membership service provider (MSP) [47] mechanisms. Hyperleder Fabric-CA is a sub-project of Hyperleder Fabric, and its function mainly provides certificate generation and management. MSP is just an interface, and Fabric-CA is an implementation of the MSP interface. The CA issues PKI (private key) certificates and Tcert-Public Key to users to ensure identity information. All the user's private information is encrypted and stored in the MSP organization, and only the PKI certificate (secret key) can be accessed and modified. The encrypted user information is always used in the transaction process. In the above process, the Identity Mixer mechanism and data privacy are analyzed in Hyperleder Fabric. The analysis is as follows:

- Fabric's zero-knowledge identity certificate

As shown in Figure 9, The CA uses idemix to protect user privacy when verifying users. It is an encryption protocol suite. The underlying signature of the Identity Mixer system allows to effectively prove the ownership of the signature and the corresponding attribute without displaying the signature and (selected) attribute value itself. It has robust identity verification and privacy protection functions, such as anonymity, transactions without disclosing the identity of the trader, and unlinkability, that is, the ability to send multiple transactions with a single identity without revealing that the transaction is sent through the same identity. In CA, we apply for idemix credentials for user users. If idemix credentials are used to execute trades, organization information will not be exposed. Idemix can allow verifier authentication without a CA's involvement, selectively disclose those attributes required by verifiers, and do not need to be linked to their transactions. Even if a user sends multiple transactions, it cannot reveal that they are from the same user.

- Data privacy

As shown in Figure 10, when a user generates transaction data in the network, according to the business scenario of this article, the data generated by the transaction is realized by a combination of symmetric encryption and public-key encryption. The encryption process is as follows: 1. Generate a symmetric encryption key (AES-GMN) for the transaction data of users A and B, and save the encryption key in the CA center; 2. Use a symmetric encryption key to encrypt data; 3. Use the public key certificate (Tcert-Public Key) authorized by the CA center to encrypt the symmetric key (AES-GMN) to generate signed encrypted transaction data (after the encryption is completed, it will be sent to the node to wait for the chain); 4. Only users or administrators authorized by the CA with a public key certificate (Tcert-Public Key) can decrypt data when accessing the encrypted data. The entire data encryption process guarantees data privacy.

As shown in Figure 11, the proposed system consists of four layers: network layer, functional layer, API router layer, and the application layer. The network layer provides the running environment for the blockchain infrastructure. The Winding Tree network is deployed on Infura, offering instant, scalable API access to Ethereum and distributed file systems. The REST API abstracts the services provided by the blockchain network into web services for convenient access. The functional layer specifies a set of service modules to handle the business logic of the proposed system. The API router layer is a routing gateway that controls the network load by selecting a path for traffic between the client and the blockchain network. The NGINX reverse proxy is an intermediary proxy that takes the client request to the server and delivers the response to the client. Server load balancing technology is used to distribute incoming network traffic through the Zuul cluster efficiently.

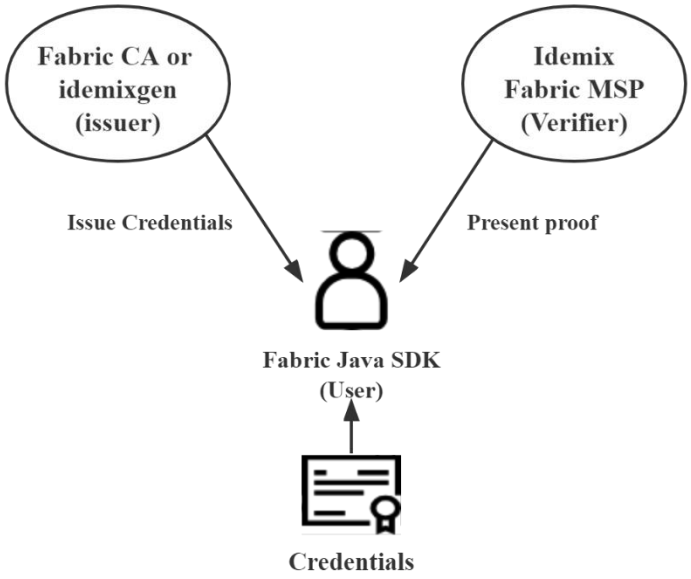

Figure 9. CA authentication process.

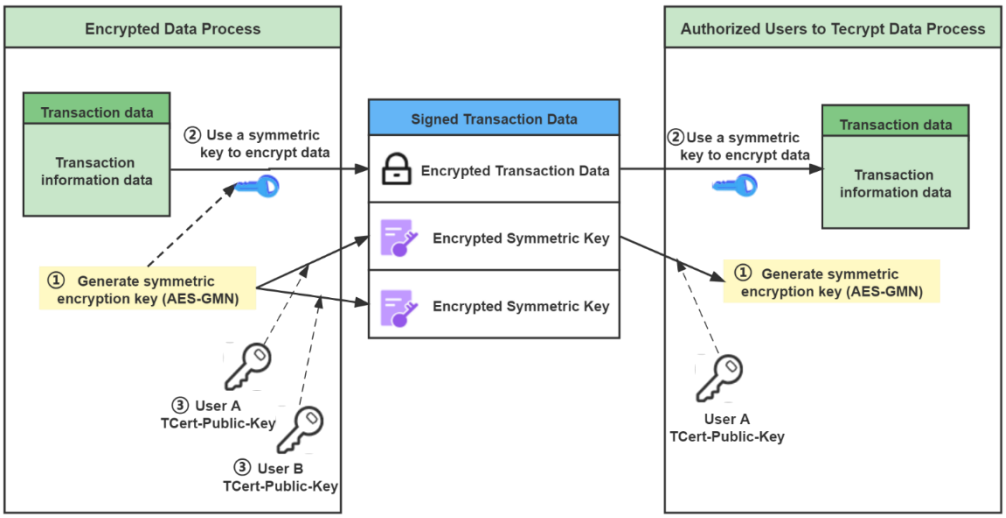

Figure 10. Data information encryption and decryption process.

4.2. Smart Contract Implementation

Figure 12 presents a sample data model of the organization content stored in the off-chain storage. The content of the data model is in JSON, comprised of two schemas: legal entity and hotel. A legal entity represents the info of the hotel owner, including name, address, and contract data. As the entry point of the public chain, the EntryPoint model contains three business directories: hotels, airlines, and OTA. For example, the hotel directory contains all registered hotel information, and each record contains hotel business information (room, price, hotel). This article uses JSON format data to process this information. As shown in Figure 13, the hotel schema represents the metadata related to the hotel, such as name, website, and location.

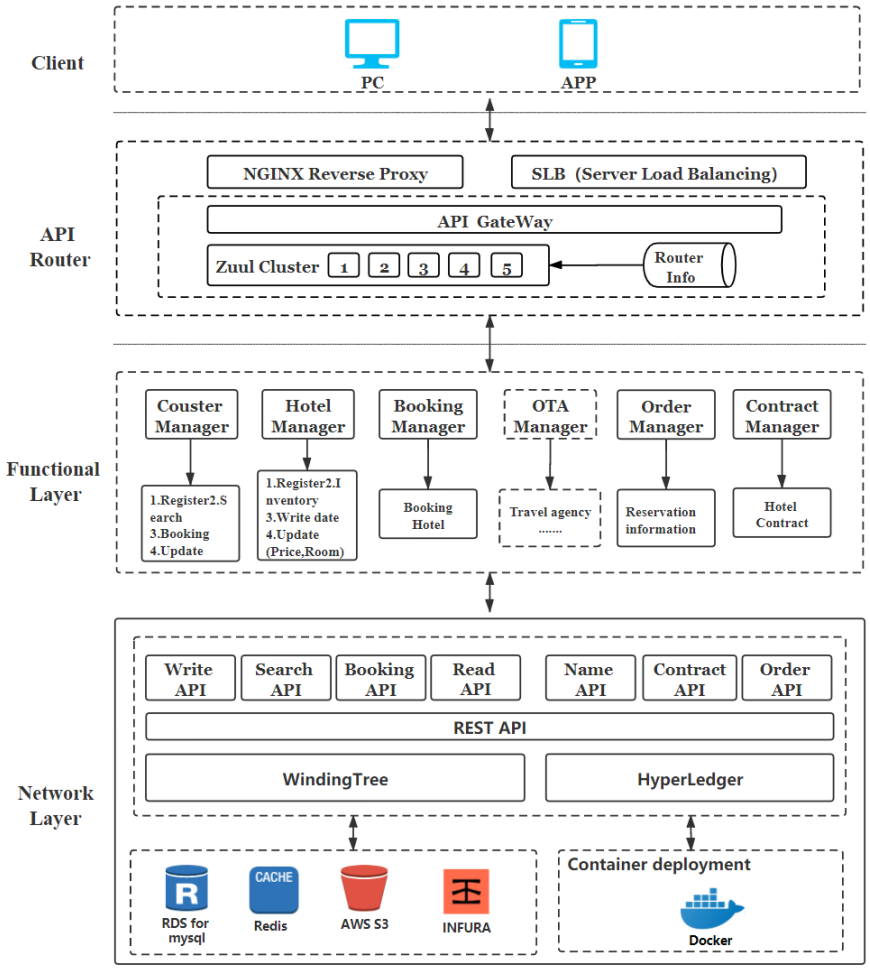

Figure 11. Configuration diagram of the prototype for the tourism service.

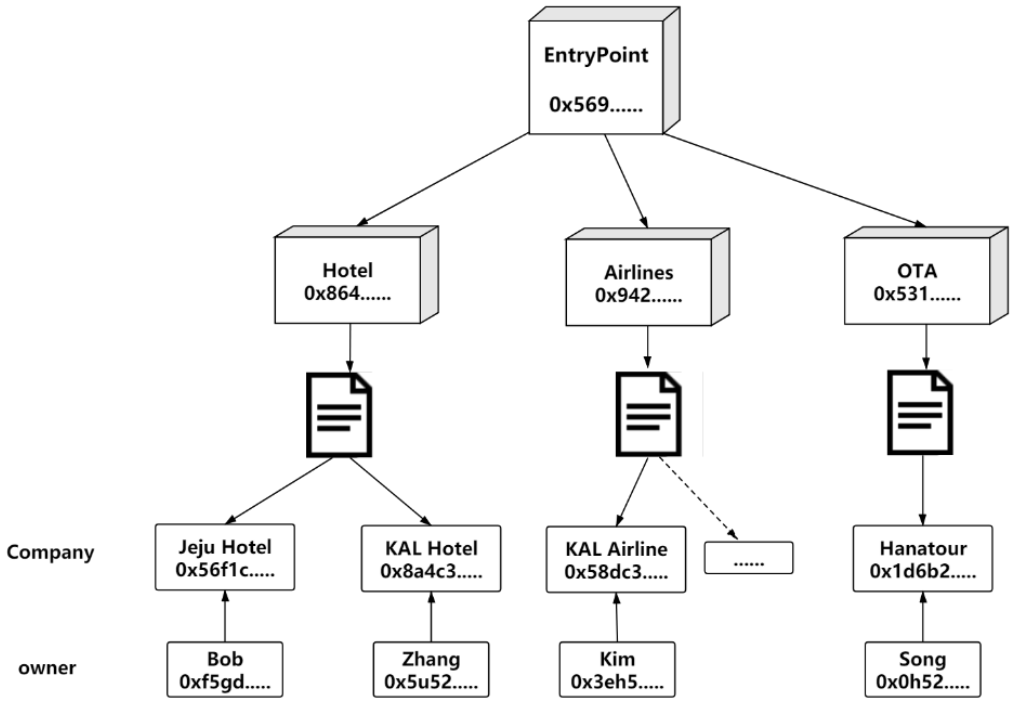

Figure 12. Smart contract architecture in Winding Tree Network.

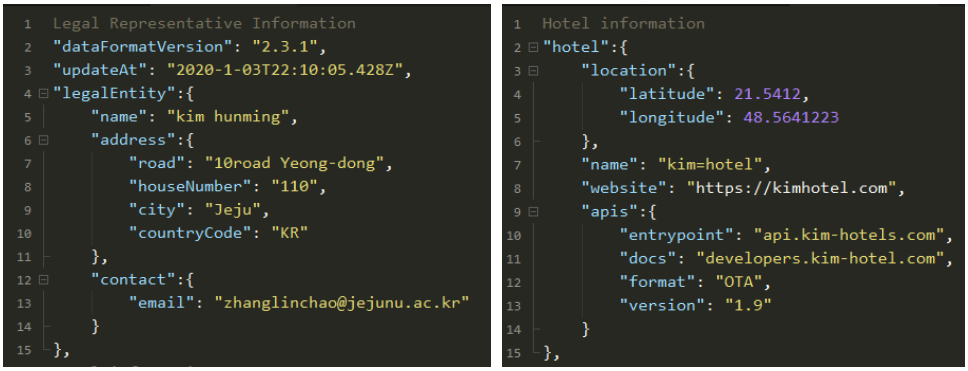

Figure 13. Sample organization content in JSON.

The smart contract of the Hyperledger Fabric network specifies the business network definition, which contains participants, assets, and transactions. Participants are members of a business network and can own assets or initiate transactions. Assets represent any property that can range from the tangible to the intangible. Transactions define the mechanism by which participants interact with assets. In this case study, the participant groups consist of customers, hotels, and regulators, as shown in Figure 14. These participants have different roles and authorities; for example, the regulator is the network manager who has full access to all network resources. Each participant contains a unique ID that the blockchain network can identify.

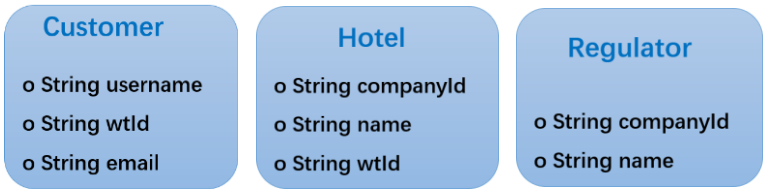

Figure 14. Participant definition in Hyperledger Fabric smart contract.

The details of the contract and contract assets are shown in Figure 15. It is worth noting that contract status is used to represent the status of a particular contract used throughout the business network. Four states are defined to describe the order's lifecycle: PLACED, CHECKIN, CHECKOUT, and END. For example, CHECKIN represents the guest who has arrived at the hotel. When the customer places the order, a contract asset is created automatically by the smart contract. The contract contains the order ID, reservation date, order status, room information, contact information, booking preferences, etc. The contract status indicates the status; for example, the group will be changed from active to termination when the customer checks out.

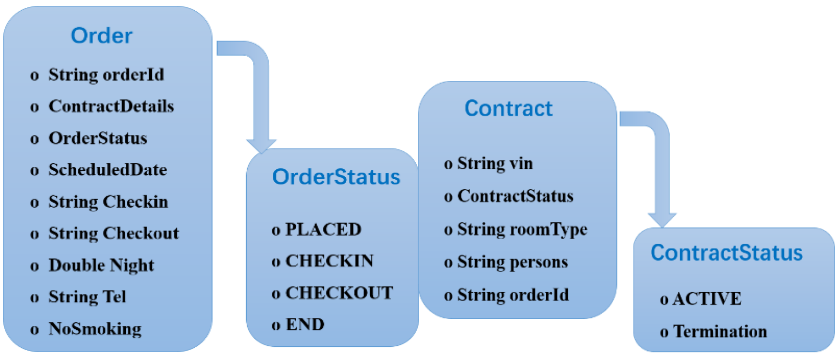

Figure 15. Order asset definition and contract asset definition in Hyperledger Fabric smart contract.

Figure 16 shows the use of a blockchain to explain the internal structure and driving relationship of the block. According to the smart contracts and the order transactions, the transaction information is stored in the two chains. Each block consists of two parts: a block header and a block body. The length of the block header is 80 bytes, and the version number is stored in the block header. The hash address of the previous block, the Merkle root, the block creation timestamp, the block workload difficulty factor, and the random number are also stored. There are mainly two types of transactions stored in the block body: coinbase transactions and ordinary transactions. The first transaction in a block is defined as a coinbase transaction, which the system rewards miners. The ordinary transaction records are submitted in the blockchain network. In the transaction order, the blocks are connected end to end. Information is stored in the main-chain and provides public and open information access APIs, including customer data, hotel information, reservation inquiries, and OTA service centers. In the sub-chain, the order transaction is completed, the order information is packaged, encrypted, and stored in the block, and the smart contract controls the core transaction and transfers assets between blocks. Data is stored in a restricted-access environment, and each access needs to be verified by the authorization verification center in Hyperledger.

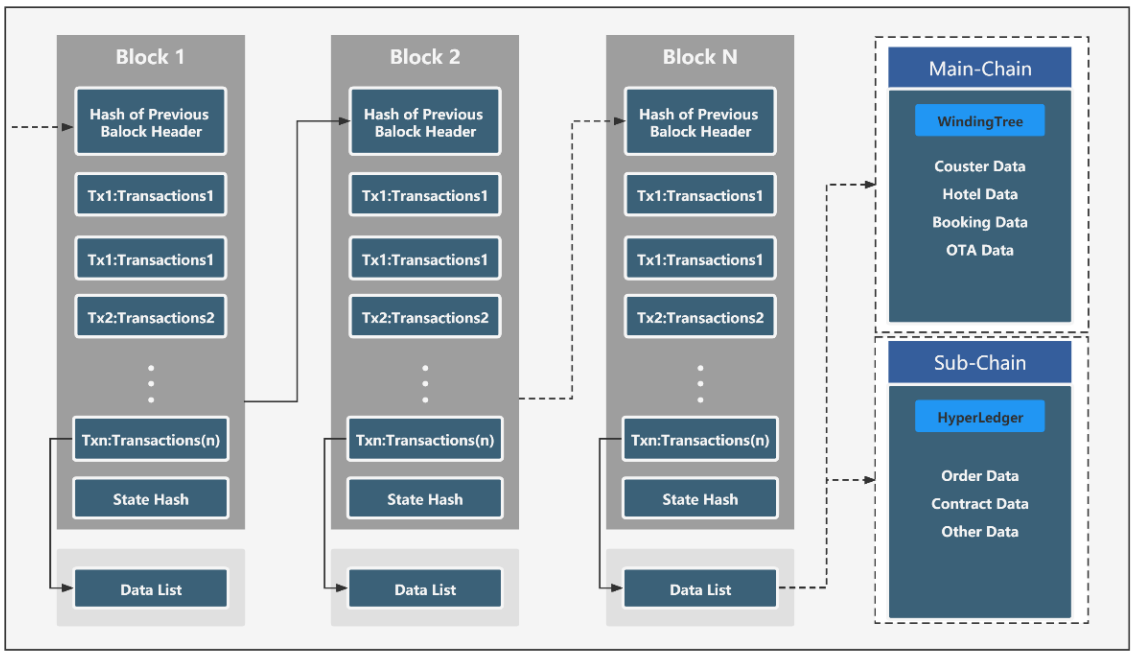

Figure 16. Block transaction information and structure.

4.3. Prototype Service Execution

As shown in Figure 17, the smart contract of Winding Tree has several layers: entry point, segment directory, and ORG.ID. The entry point provides a list of all segment directories with a unique Ethereum address to be identified by the platform. The segment directory is a collection of organizations, either hotels, airlines, or OTAs. An associated ORG represents each organization ID that the owner creates—each ORG. ID. contains a URI that points to the location in the off-chain storage where the organization information is stored. A hash is generated based on the contents to prevent the content from being tampered with.

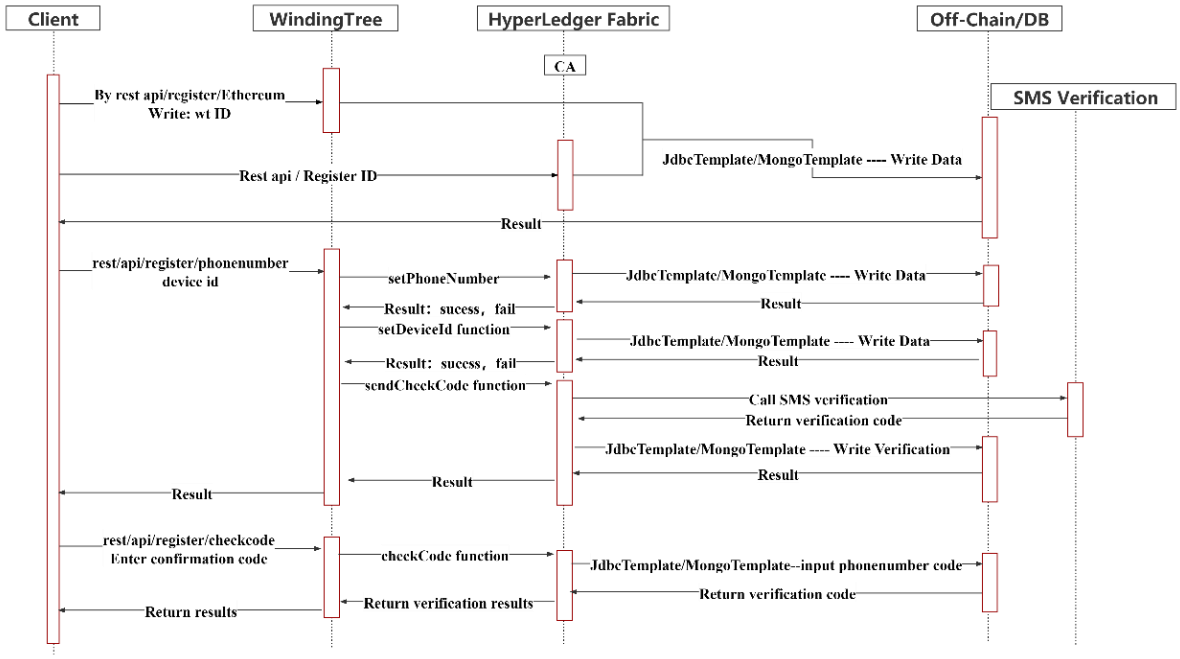

Figure 17. Execution process of user registration.

The end-user must register its identity in the system before accessing or manipulating resources in the blockchain network. Figure 4 illustrates various operations of user registration in the proposed system. This operation starts with certification from the client by invoking Contract API to check if the user info exists in the Winding Tree network. Meanwhile, another request is sent to the CA of the Hyperledger Fabric network. The CA generates a certificate (public key, enroll ID), ensures it is private, and sends it back to the user. This system also supports real-name authentication by sending a verification code to the user's smartphone. After verification, the user info will be stored in the off-chain storage.

As shown in Figure 18, the authorized user identified by the CA can query the hotel data stored in the Winding Tree network through the SearchAPI. The network returns the inquiry result to the user client. Afterward, the user can book a selected room, and the client initializes another request to perform the booking process. The Winding Tree network changes the inventory accordingly and returns the price of the room. The user can place the order by submitting a transaction to the endorser peer of the Hyperledger Fabric network. The transaction is signed by these endorser peers and sent to the consensus manager. The consensus manager arranges the transaction into a block and sends the block information to all committer peers. These committer peers verify the transaction before being added to the ledger. Lastly, a notification is issued to inform the client whether the transaction is executed or not.

Figure 19 describes the workflow associated with the booking process that occurs over the network. First, the customer requests the Winding Tree network to obtain information about the available hotel. Then, the customer initializes the booking request by calling the booking API. Meanwhile, another request is sent to the Hyperledger Fabric, and a new order asset is created in the ledger. The order asset creates a contract asset that is signed by the order ID. A notification is then sent back to the hotel to confirm the order on the blockchain network. Once the order process is complete, the hotel can update the inventory information stored in the Winding Tree network and change the order status accordingly. Finally, the Hyperledger Fabric network issues an event to inform the customer that the booking process has been completed.

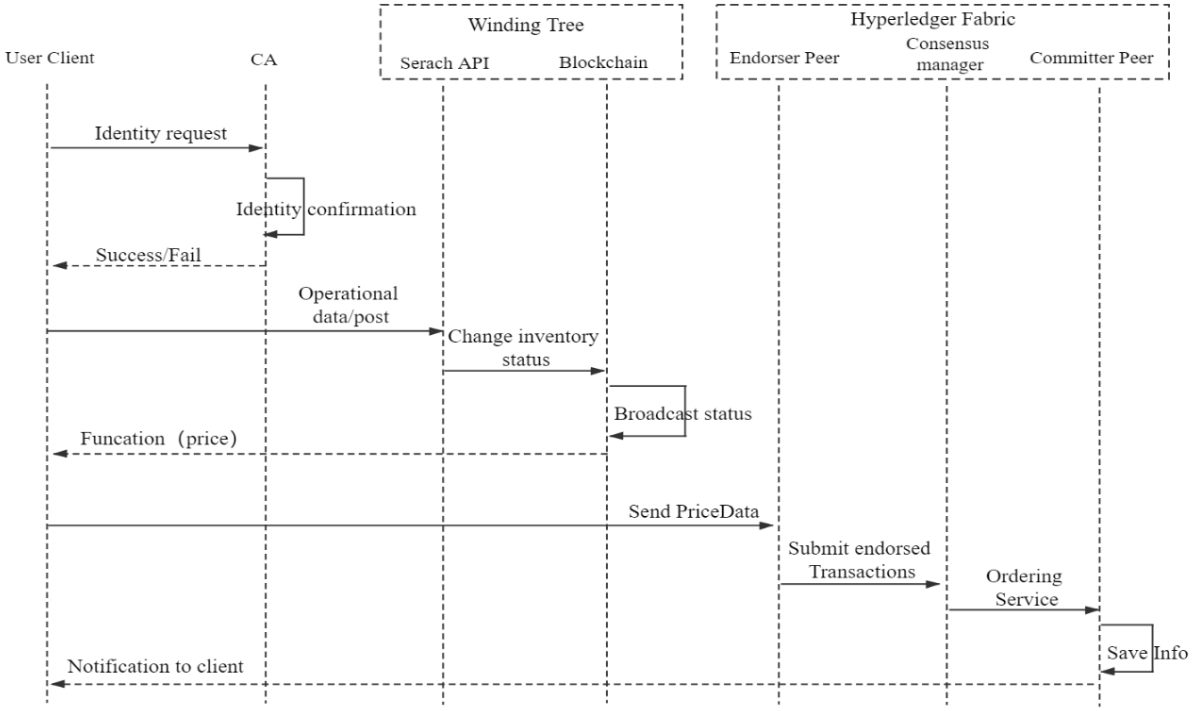

Figure 18. The transaction process of the proposed system.

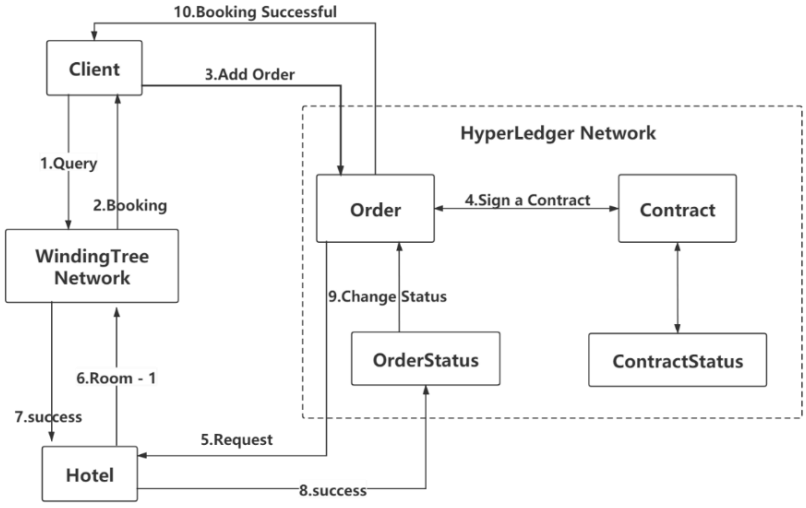

Figure 19. Booking process workflow in the proposed system.

4.3.1. Implementation Results—Backstage Management

This section presents the implementation results of the case study via different screenshots.

Figure 20 shows the screenshot of the ReadAPI, which can read data from the Winding Tree network. It provides several built-in RestAPIs to retrieve information about the hotel, such as room type, rate plan, and availability. It also supports conditional statements in the query, which allow the user to retrieve the information of a specific hotel through a specific ID, as well as query, read, and write information about the chain through the RestAPI while using HyperLedger to control permissions and ensure data security.

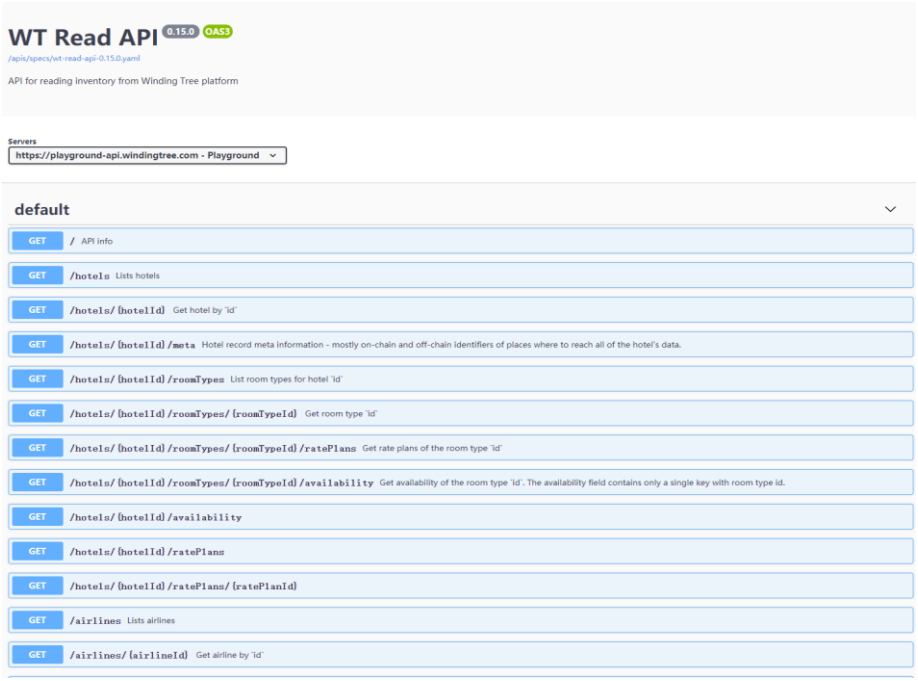

Figure 20. Screenshot of ReadAPI.

Figure 21 presents the screenshot of the participant dashboard on the web. This dashboard enables the admin to view the participant info after logging into the system. It also provides various entries through which the admin can update or delete the info of a selected participant. In the actual applications, the user (the organization) registers and changes the private information on the application, stores it in the blockchain, and gets an encryption certificate (private key). No organization can view the user's (company's) private information; only private key verification is allowed and later modified.

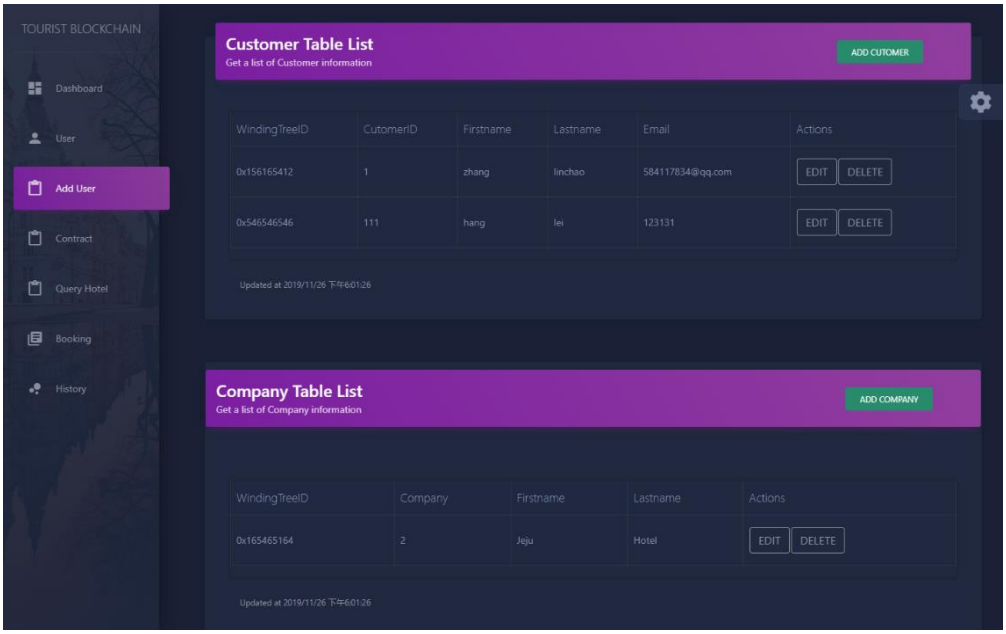

Figure 21. Screenshot of the participant dashboard.

Figure 22 presents the screenshot of the contract dashboard on the web. The admin can read, create, update, and delete the contract asset via the client application. Each

contract contains an order ID, which is the reference to the associated order asset. It also provides other information, including room type, check-in time, check-out time, and order status. The contract dashboard provides two entries for adding and updating the order asset to facilitate the operation.

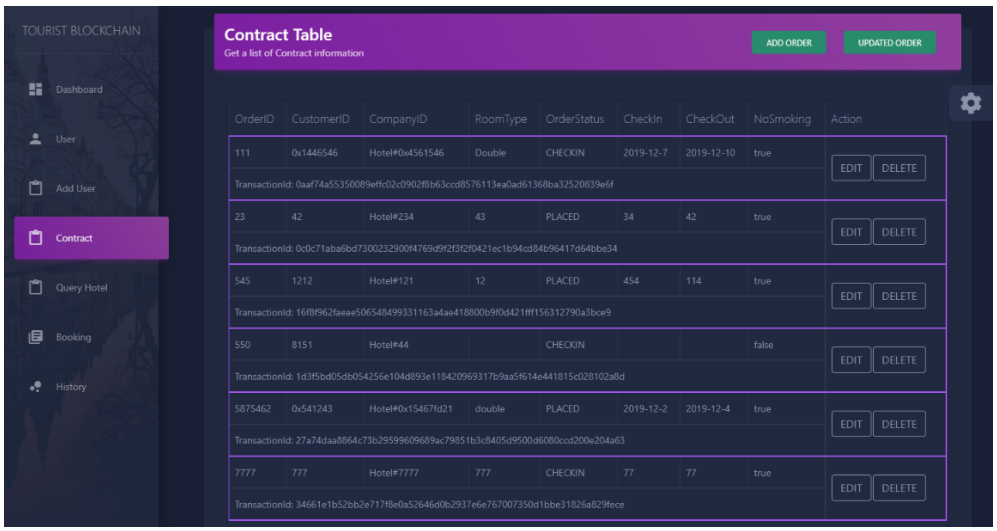

Figure 22. Screenshot of the contract dashboard.

Figure 23 presents the screenshot of the transaction record dashboard through which the admin can audit the transaction history. Each transaction record includes a transaction ID., timestamp, type, and the participant who submits the transaction. The transaction log and transaction order information are recorded on the blockchain, and unexpected hotel problems occur. The immutable nature of the blockchain makes transaction records strong evidence.

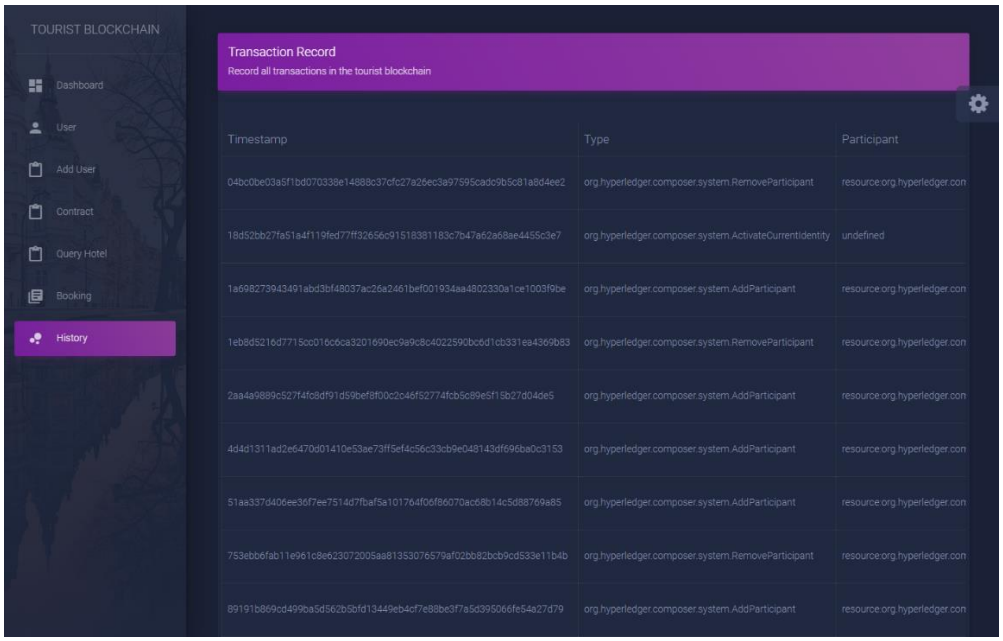

Figure 23. Screenshot of the transaction record dashboard.

4.3.2. Implementation Results—Client

Figure 24 shows using location information to get the current location to find nearby hotels. It can display all hotel information in the network within the specified area. The

report includes the hotel's name and address. The user clicks to show the hotel to display detailed information. The user can quickly book the hotel. They can then use the map to show that the hotel is a friendly UI service and function for the booking service. The user clicks the icon to get hotel information, query hotel room information, and make hotel reservations.

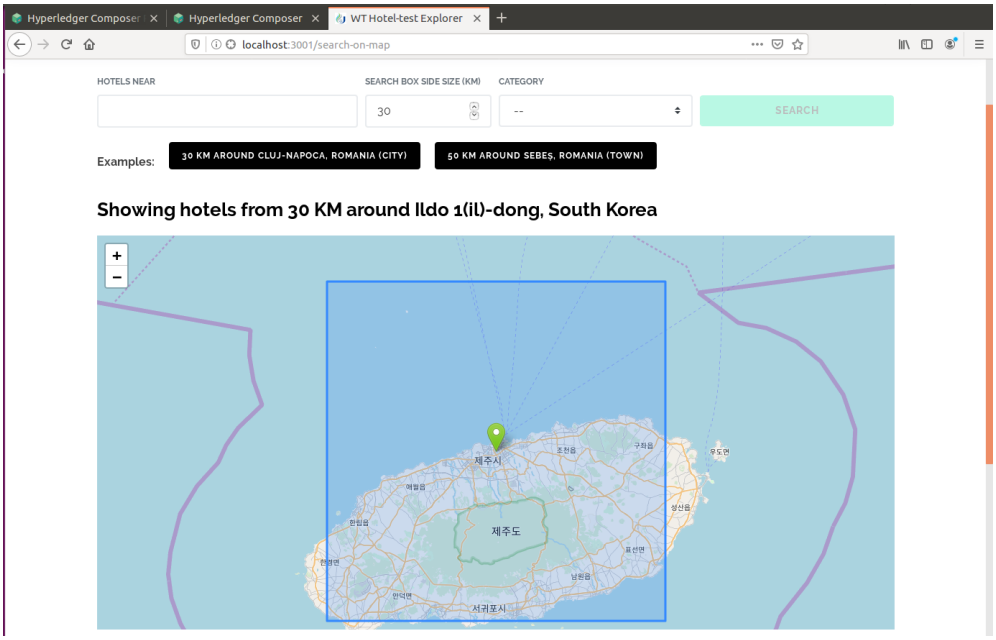

Figure 24. Network Client Map Search Implementation.

Figure 25 shows this article uses a browser to implement client functions. The client sets the start date and end date and then calls ReadAPI to display qualified hotel information and obtain hotel information (pictures, introductions, prices) from the main-chain. Moreover, the client develops more functions, such as condition filtering, region selection, and preference settings.

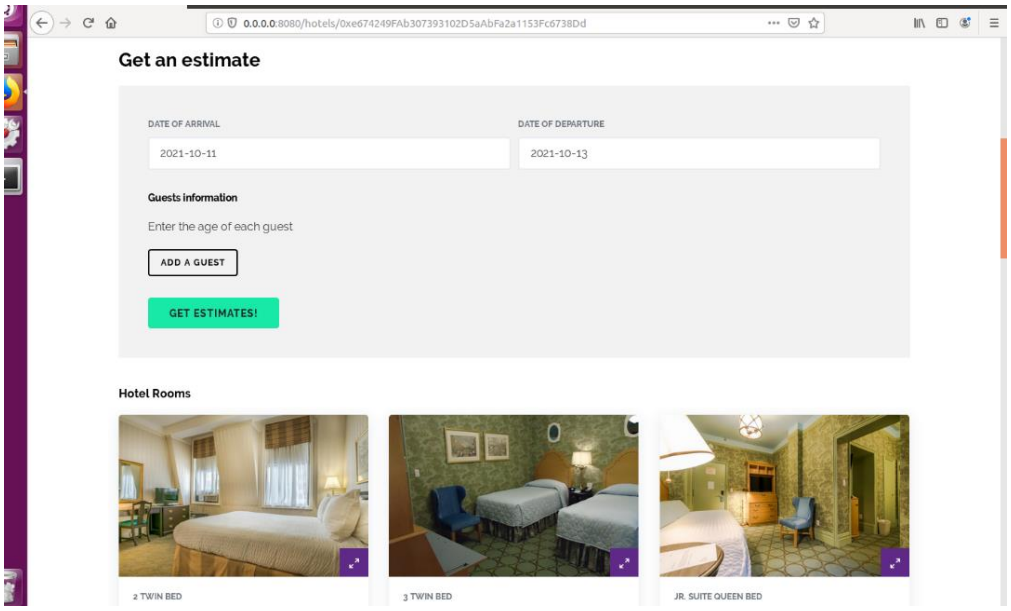

Figure 25. The client searches for hotel information.

Figure 26 shows the hotel information containing the room's introduction and the corresponding label, and the required hotel can be accurately found by filtering the title. The client uses asynchronous data loading to monitor the number of rooms and hotel room inventory information. All room information is loaded from main-chain. If the room information is changed, the modification information can be queried and recorded on the blockchain.

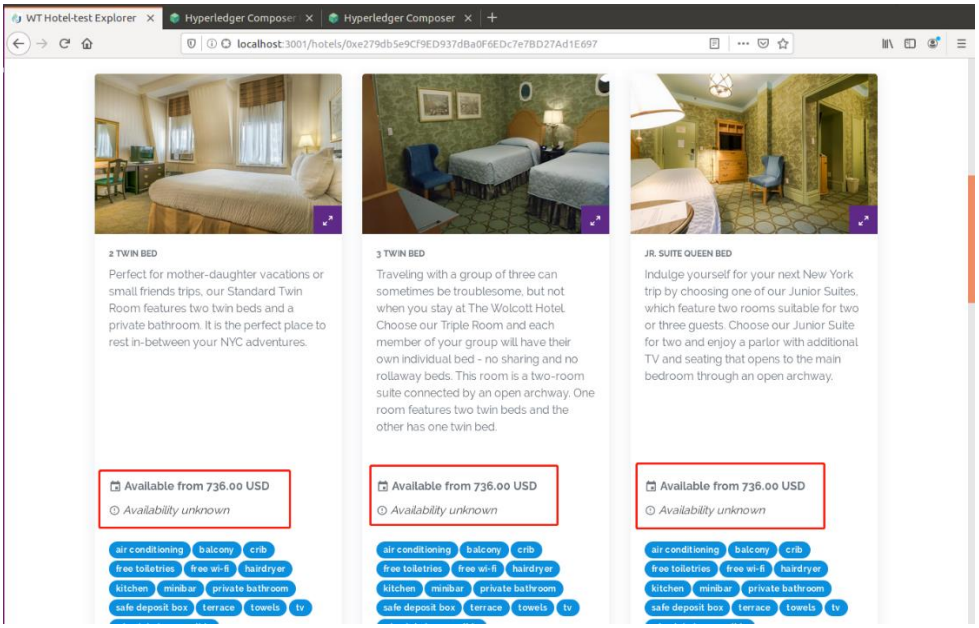

Figure 26. The client search results.

4.4. Performance Evaluation

This section evaluates the system performance of our case study. We use Hyperledger Caliper [48], an open-source blockchain benchmark tool designed to measure the performance of different blockchain implementations. To demonstrate the efficiency and significance of the designed solution, an experiment was conducted to compare the arrangement between the underlying single-chain architecture and the proposed multi-chain architecture. We explored the differences between these two architectures by evaluating the average network latency and transaction throughput. The network latency represents the amount of time taken for a transaction to be executed within the network. More precisely, the network latency consists of the time from the point that the transaction is submitted and the time for broadcasting and validating the trade since the consensus occurs. The transaction throughput represents the number of valid transactions committed by the blockchain in a given time, and the unit is transactions per second (tps). In this experiment, we modified the script provided by Hyperledger Caliper to target one function of our case study application. This function is used to create the order since the user client most continually calls it. We conducted 10 rounds of evaluation experiments in different send rates, ranging from 100 tps to 500 tps. The single-chain architecture performed the first five rounds, while the proposed multi-chain architecture performed five rounds. For reducing the underlying impact of network congestion and overload, multiple rounds of experiments were conducted. Average transaction throughput and network latency in different rounds of the experiment are presented in Figures 24 and 25, respectively.

It is obvious to see from the performance evaluation results that with the increase in send rate, the transaction throughput in a single-chain architecture decreases linearly. However, As shown in Figures 27 and 28, For the multi-chain architecture, it increases significantly. The proposed multi-chain architecture has much higher throughout than single-chain architecture in all of the rounds. The maximum throughput of the single-chain

architecture was observed around 48 tps at a send rate of 100 tps. For the multi-chain architecture, it was observed around 476 tps at a send rate of 500 tps. As shown in Figures 29 and 30, For the network latency evaluation, it is evident that the average latency in single-chain architecture increases significantly with the growing send rate. By contrast, the increase in average latency in multi-chain architecture is negligible that can even be ignored.

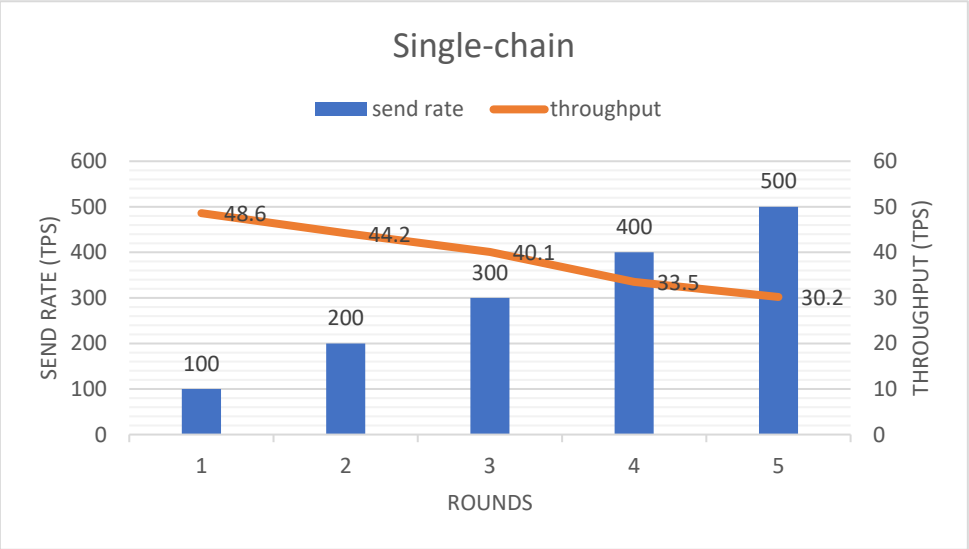

Figure 27. Transaction throughput evaluation of single-chain architecture.

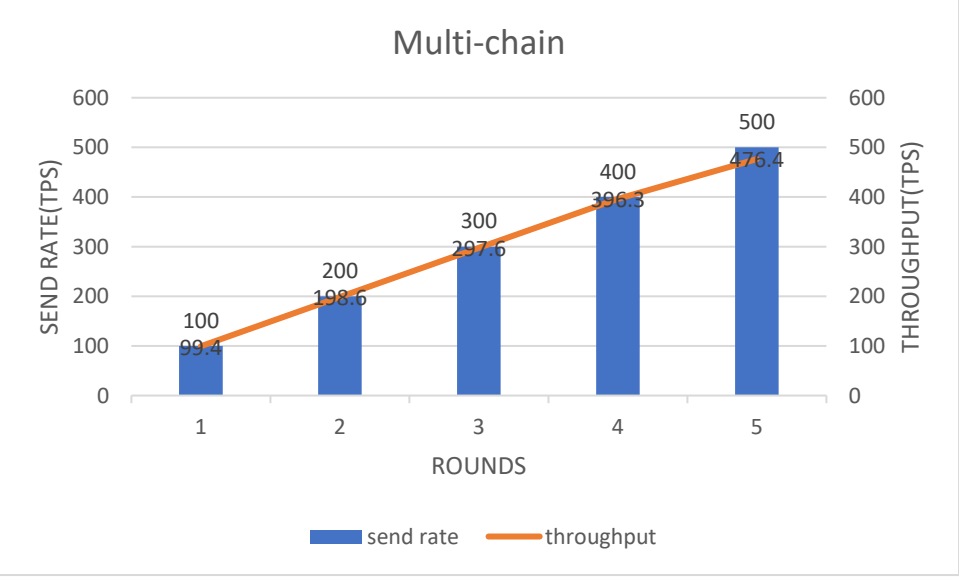

Figure 28. Transaction throughput evaluation of multi-chain architecture.

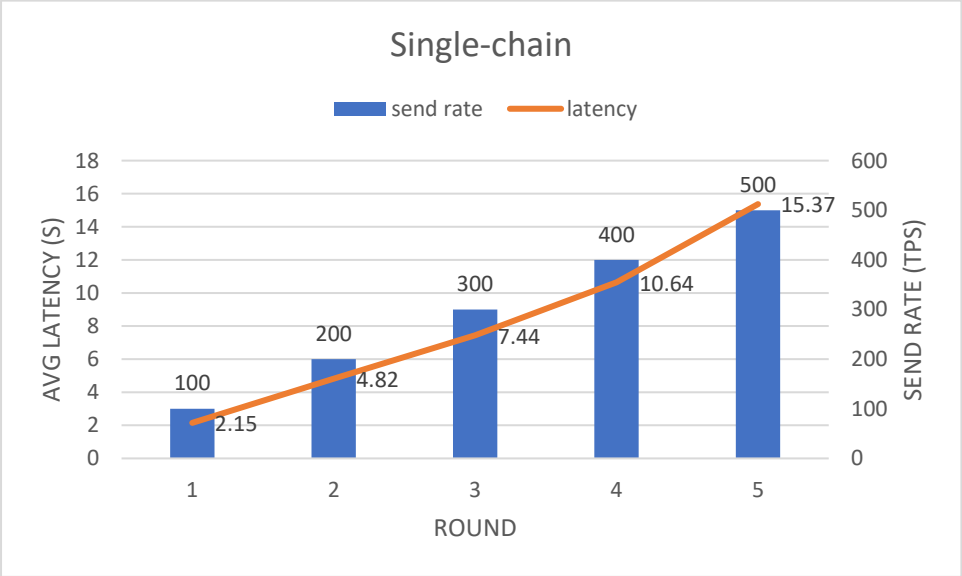

Figure 29. Network latency evaluation of single-chain architecture.

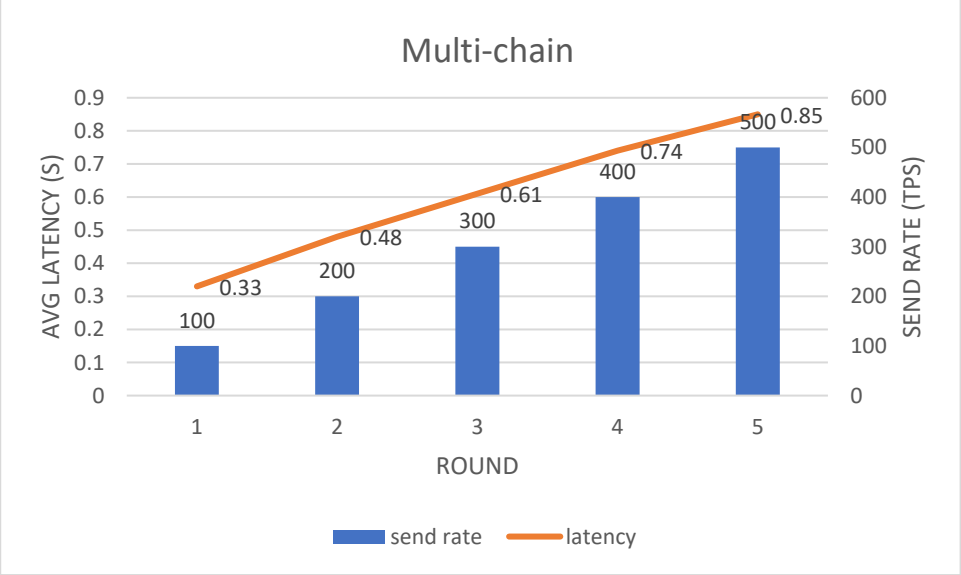

Figure 30. Network latency evaluation of multi-chain architecture.

Pongnumkul et al. [49] tested the throughput and latency performance of Ethereum and HyperLedger in the paper. Performance comparison is performed by configuring virtual machines in the same environment. Figure 31 shows the comparison of transaction throughput. After the first two Ethereum, Hyperledger Fabric, and multi-chain increase transaction volume, the throughput performance peaks, and the version is blocked, resulting in performance degradation.

Figure 32 shows network latency performance test results with increased transaction volume. Among the three types of network latency, multi-chain latency has a linear relationship with transaction volume. Ethereum and Hyperledger Fabric reached the maximum transaction volume in the third test, and the latency increased immediately after the decrease.

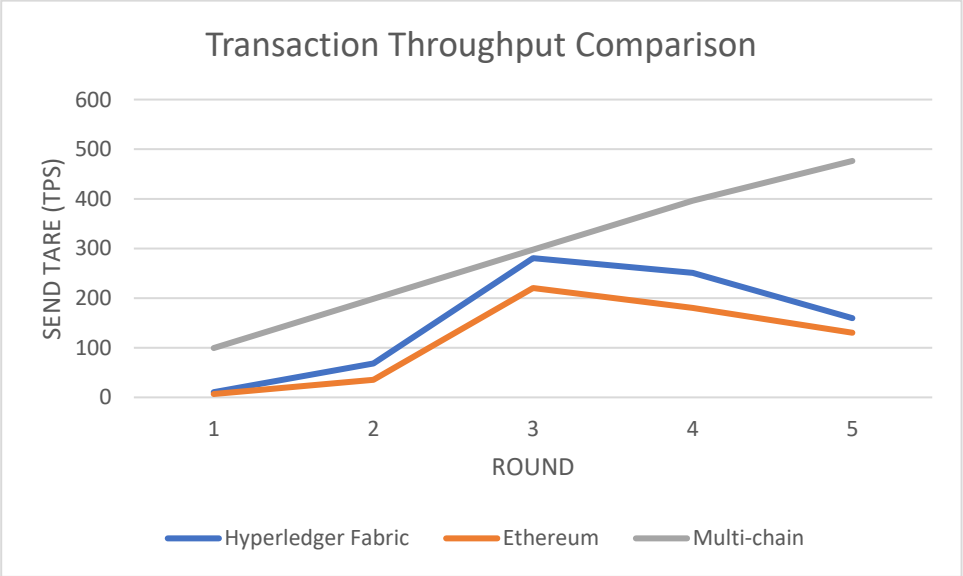

Figure 31. Comparison of three types of network transaction throughput performance.

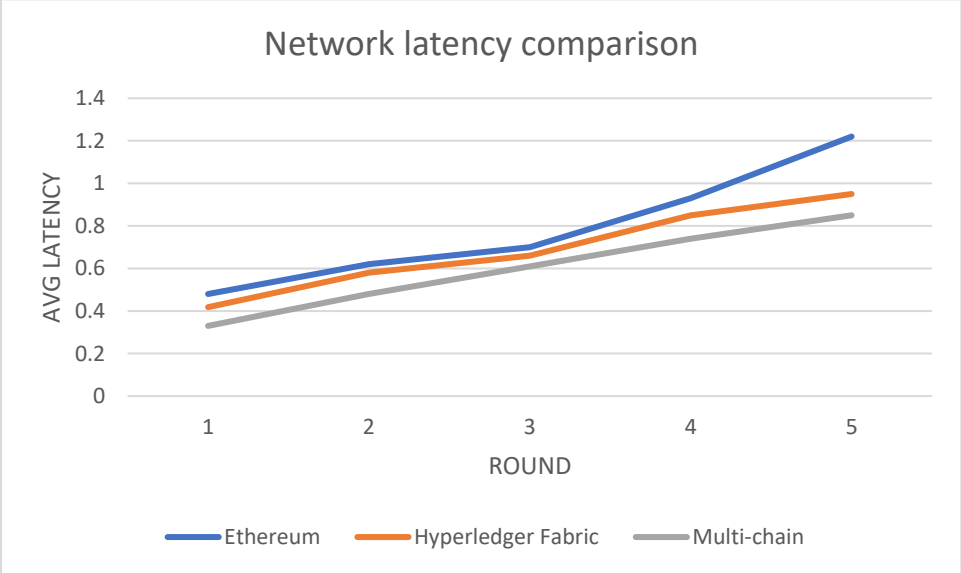

Figure 32. Comparison of three types of network transaction latency performance.

5. Comparison and Significance

This section carries out a detailed benchmark study to compare the proposed system with some recent research reviewed in the related work section. The following properties that play an essential role in analyzing these platforms are considered in this study, and the evaluation results are illustrated in Table 3.

It is evident to see from the table that a majority of blockchain-based tourism platforms are single-chain implemented on permission-less network infrastructures, where identities of all network participants are anonymous and without trust. This would raise security and privacy concerns since the smart contracts themselves can be modified by anyone. The transaction data can be interrupted if someone has enough computing power to control most of the network. Consequently, these systems utilize their encrypted tokens to incent mining and give miners who validate the transaction in the blockchain. However, the usage of tokens can significantly increase transaction costs and reduce transaction processing speed. Moreover, token-based blockchain platforms have limitations in interoperating with

other platforms if the token used in these systems is not unified. The proposed system is built on a combined infrastructure. The non-sensitive data such as hotel info is stored in the permission-less Ethereum network, and the sensitive data like order is preserved in the permissioned Hyperledger Fabric network. Besides, the payment process is performed in the Hyperledger Fabric network, eliminating the risk of the smart contract being modified by malicious users and improving the transaction processing capability as there is no native token.

Table 3. Comparative study of the proposed system with existing systems.

Name	Type	Number of Chains	Native Cryptocurrency	Infrastructure	Smart Contract	Efficiency	Support Client
Tripio	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
Webjet	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
ZatGo	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
Travelchain	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
Deskbell	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
Flightdelay	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
Cool Cousins	Permissionless	Single	Yes	Ethereum	Yes	Low	Yes
Proposed Platform	Permissionless/ Permissioned	Multiple	No	Ethereum/ Hyperledger Fabric	Yes	High	Yes

This paper presents a case study of hotel reservations in the tourism industry to justify the feasibility and efficiency of the designed solution. However, this system supports a modular architecture that can be further extended to meet the demand of various business cases such as home rentals, tourism currency, and travel recommendations. For example, home rentals will significantly benefit from the results of this work as blockchain technology can directly bridge the gap between today's house owners and tourism without third-party commissions. Homeowners can post the rental information to the blockchain, and all operations are entirely transparent and accessible to authorized users.

6. Conclusions

The adoption of blockchain technology has shown a new growth potential to trigger a revolution across the tourism industry but is still at an early stage. This paper proposes a multi-chain blockchain architecture to enhance the transaction processing capability and provide various tourism-related services. The network can be further extended to adopt other blockchain implementations. The proposed multi-chain architecture includes a public chain and a private chain for different data types regarding sensitivity and privacy. The public chain keeps non-sensitive data such as property information, while the private chain keeps sensitive data such as user information and orders. A hotel booking case study was implemented to demonstrate the usability and functionality of the developed solution. The public chain is implemented on Winding Tree, and the private chain is based on Hyperledger Fabric. The Winding Tree network supports direct interaction between different service providers and service sellers. Service providers can post information about their services, and these activities are available to all the participants. The Hyperledger Fabric network is responsible for serving the payment service and recording the orders. The significance of the designed multi-chain architecture is elaborated through a comprehensive evaluation experiment and a benchmark analysis by comparing the proposed system with some current studies. We will refine the proposed architecture to connect with more blockchain platforms and additional support for different tourism services in future work.

Author Contributions: L.Z., L.H., W.J. and D.K. designed the overall system. L.Z., L.H. and W.J. implemented the overall system and performed experiments. L.Z., L.H., W.J. and D.K. wrote this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Energy Cloud R&D Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2019M3F2A1073387), and this work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00188, Open source development and standardization for AI enabled IoT platforms and interworking). Any correspondence related to this paper should be addressed to Dohyeun Kim.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. WTTC: World Travel and Tourism Council, Global Economic Impact and Trends. 2019. Available online: https://www.wttc.org/economic-impact/country-analysis/ (accessed on 18 February 2019).
2. Colombo, E.; Bagiio, R. Tourism distribution channels. In *Knowledge Transfer to and within Tourism (Bridging Tourism Theory and Practice)*; Scott, N., De Martino, M., Van Niekerk, M., Eds.; Emerald Publishing Limited: Bingley, UK, 2017; Volume 8, pp. 289–301.
3. Pilkington, M. Can Blockchain Technology Help Promote New Tourism Destinations? The Example of Medical Tourism in Moldova. Available online: https://ssrn.com/abstract=2984479 (accessed on 18 February 2019).
4. Erceg, A.; Sekuloska, J.D.; Kelić, I. Blockchain in the Tourism Industry—A Review of the Situation in Croatia and Macedonia. *Informatics* **2020**, *7*, 5. [CrossRef]
5. Kwok, A.O.J.; Koh, S.G.M. Is blockchain technology a watershed for tourism development? *Curr. Issues Tour.* **2019**, *22*, 2447–2452. [CrossRef]
6. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 18 February 2019).
7. Peters, G.W.; Panayi, E. Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money. In *Banking Beyond Banks and Money*; Springer: Cham, Switzerland, 2016; pp. 239–278.
8. Hang, L.; Kim, D.-H. Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity. *Sensors* **2019**, *19*, 2228. [CrossRef] [PubMed]
9. Hang, L.; Choi, E.; Kim, D.-H. A Novel EMR Integrity Management Based on a Medical Blockchain Platform in Hospital. *Electronics* **2019**, *8*, 467. [CrossRef]
10. Hang, L.; Kim, D.-H. SLA-Based Sharing Economy Service with Smart Contract for Resource Integrity in the Internet of Things. *Appl. Sci.* **2019**, *9*, 3602. [CrossRef]
11. Hang, L.; Kim, D.-H. Reliable Task Management Based on a Smart Contract for Runtime Verification of Sensing and Actuating Tasks in IoT Environments. *Sensors* **2020**, *20*, 1207. [CrossRef] [PubMed]
12. Hang, L.; Ullah, I.; Kim, D.-H. A secure fish farm platform based on blockchain for agriculture data integrity. *Comput. Electron. Agric.* **2020**, *170*, 105251. [CrossRef]
13. Jamil, F.; Hang, L.; Kim, K.; Kim, D. A Novel Medical Blockchain Model for Drug Supply Chain Integrity Management in a Smart Hospital. *Electronics* **2019**, *8*, 505. [CrossRef]
14. Peng, Z.; Jules, W.; Schmidt, D.C.; Gunther, L.; Rosenbloom, S.T. FHIRChain: Applying Blockchain to securely and scalably share clinical data. Comput. *Struct. Biotechnol. J.* **2018**, *16*, 267–278.
15. Pazaitis, A.; De Filippi, P.; Kostakis, V. Blockchain and value systems in the sharing economy: The illustrative case of Backfeed. *Technol. Forecast. Soc. Chang.* **2017**, *125*, 105–115. [CrossRef]
16. Önder, I.; Treiblmaier, H. Blockchain and tourism: Three research propositions. *Ann. Tour. Res.* **2018**, *72*, 180–182. [CrossRef]
17. Blockchain and Distributed Ledger Technology at Travelport. Available online: https://www.travelport.com/sites/default/files/travelport-blockchain-whitepaper.pdf (accessed on 24 February 2020).
18. Populstay. Available online: https://www.populstay.com/Populstay_Whitepaper_EN.pdf (accessed on 24 February 2020).
19. Travala. Available online: https://www.travala.com/ (accessed on 24 February 2020).
20. Wood, G. Polkadot: Vision for a Heterogeneous Multi-Chain Framework. Available online: https://polkadot.network/PolkaDotPaper.pdf (accessed on 18 February 2019).
21. Coll Vendors in Blockchain Platforms. Available online: https://www.gartner.com/en/documents/3734117 (accessed on 1 February 2021).
22. WindingTree. Available online: https://windingtree.com/ (accessed on 21 February 2020).
23. Cachin, C. Architecture of the Hyperledger Blockchain Fabric. Available online: https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf (accessed on 21 February 2020).
24. Alnemari, A.; Arodi, S.; Sosa, V.R.; Pandey, S.; Romanowski, C.; Raj, R.; Mishra, S. Protecting Infrastructure Data via Enhanced Access Control, Blockchain and Differential Privacy. Available online: https://link.springer.com/chapter/10.1007/978-3-030-04537-1_7 (accessed on 1 February 2020).
25. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef] [PubMed]
26. Pop, C.; Cioara, T.; Antal, M.; Anghel, I.; Salomie, I.; Bertoncini, M. Blockchain Based Decentralized Management of Demand Response Programs in Smart Energy Grids. *Sensors* **2018**, *18*, 162. [CrossRef] [PubMed]

27. Calvaresi, D.; Leis, M.; Dubovitskaya, A.; Schegg, R.; Schumacher, M. Trust in tourism via blockchain technology: Results from a systematic review. In Proceedings of the Information and Communication Technologies in Tourism 2019, Nicosia, Cyprus, 30 January–1 February 2019; pp. 304–317.
28. Ozdemir, A.I.; Ar, I.M.; Erol, I. Assessment of blockchain applications in travel and tourism industry. *Qual. Quant.* **2019**, *54*, 1549–1563. [CrossRef]
29. Filimonau, V.; Naumova, E. The blockchain technology and the scope of its application in hos-pitality operations. *Int. J. Hosp. Manag.* **2020**, *87*, 102383. [CrossRef]
30. Nam, K.; Dutt, C.S.; Chathoth, P.; Khan, M.S. Blockchain technology for smart city and smart tourism: Latest trends and challenges. *Asia Pac. J. Tour. Res.* **2019**, *26*, 454–468. [CrossRef]
31. Tripio. Available online: http://trip.io/en/ (accessed on 21 February 2020).
32. Webjet. Available online: https://www.rezchain.com/ (accessed on 25 February 2020).
33. ZatGo. Available online: http://www.zatgo.net/ (accessed on 21 February 2020).
34. Travelchain. Available online: https://travelchain.io/ (accessed on 21 February 2020).
35. Bodkhe, U.; Bhattacharya, P.; Tanwar, S.; Tyagi, S.; Kumar, N.; Obaidat, M.S. BloHosT: Blockchain Enabled Smart Tourism and Hospitality Management. In Proceedings of the 2019 International Conference on Computer, Information and Telecommunication Systems (CITS), Beijing, China, 28–30 August 2019; pp. 1–5. [CrossRef]
36. Deskbell Chain. Available online: https://deskbell.io/ (accessed on 25 February 2020).
37. Flightdelay. Available online: https://fdd.etherisc.com/ (accessed on 25 February 2020).
38. Cool Cousins. Available online: https://www.coolcousin.com/ (accessed on 25 February 2020).
39. Al-Saqaf, W.; Seidler, N. Blockchain technology for social impact: Opportunities and challenges ahead. *J. Cyber Policy* **2017**, *2*, 338–354. [CrossRef]
40. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
41. Kan, L.; Wei, Y.; Muhammad, A.H.; Siyuan, W.; Gao, L.C.; Kai, H. A multiple blockchains architecture on inter-blockchain communication. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018.
42. Hwang, G.H.; Chen, P.H.; Lu, C.H.; Chiu, C.; Lin, H.C.; Jheng, A.J. InfiniteChain: A Multi-Chain Architecture with Distributed Auditing of Sidechains for Public Blockchains. Available online: https://link.springer.com/chapter/10.1007/978-3-319-94478-4_4 (accessed on 1 February 2020).
43. Chen, Z.D.; Zhuo, Y.U.; Duan, Z.B.; Kai, H.U. Inter-Blockchain Communication. Available online: http://dpi-proceedings.com/index.php/dtcse/article/view/12539/0 (accessed on 1 February 2020).
44. Pillai, B.; Biswas, K.; Muthukkumarasamy, V. Cross-chain interoperability among blockchain-based systems using transactions. *Knowl. Eng. Rev.* **2020**, *35*, E23. [CrossRef]
45. Kiayias, A.; Zindros, D. Proof-of-Work Sidechains. Available online: https://www.semanticscholar.org/paper/Proof-of-Work-Sidechains-Kiayias-Zindros/9a9961bc656739be93567a9ac61d4b5da761bd01 (accessed on 1 February 2020).
46. ACL. Available online: https://github.com/hyperledger/fabric-docs-i18n/blob/release-2.2/docs/locale/es/source/access_control.md (accessed on 1 November 2021).
47. MSP. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/msp.html (accessed on 1 March 2021).
48. Hyperledger Caliper—A Blockchain Benmark Tool. Available online: https://www.hyperledger.org/projects/caliper (accessed on 28 February 2020).
49. Pongnumkul, S.; Siripanpornchana, C.; Thajchayapong, S. Performance Analysis of Private Blockchain Platforms in Varying Workloads. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–6.