

A Decentralized Identity-based Cross-Domain Data Trading System

Kezhi Zhang^{1*}

¹Shouguang City Government Administration Service Center, Shouguang Shandong 262705, China

Abstract

This paper presents a decentralized identity-based cross-domain data trading system designed to address identity inconsistency, service unavailability, transaction non-atomicity, and insufficient regulatory granularity in blockchain-enabled data markets. The proposed architecture introduces a unified decentralized identity layer for users and data assets, a secure cross-domain DID resolution protocol, a consistency-preserving cache, and an availability mechanism that combines consistent-hash-based scope allocation, redundancy, incentives, malicious-node detection, and punishment. On this basis, modular smart contracts implement cross-domain information verification, identity-attribute regulation, transaction validation, payment approval, and data transfer, while decentralized custody nodes and parallel transmission improve data delivery robustness. A prototype was deployed on 20 CentOS 7 servers divided into two blockchain domains, with each server equipped with an eight-core 2.4 GHz processor, 8 GB RAM, and 50 GB SSD storage. Experiments using 100,000 ERC-1056 DID records show that cache-free cross-domain resolution maintains complete consistency, while a 50% cache hit rate causes 14% of first-resolution results to require secondary resolution; the proposed update and re-resolution mechanisms restore full consistency and improve the effective hit rate by 72%. Even when the cache hit rate rises further, the initial inconsistency ratio increases only to 19%. Without caching, average cross-domain identity resolution latency is 180 ms, whereas LRU caching at a 70% hit rate reduces it to 82 ms, a 54.4% decrease. At a transaction submission rate of 500 tps, submitted transactions wait an average of 2.3 blocks, and two sequential smart-contract executions require approximately 5 s. With a gas limit of 10,000,000, the base blockchain throughput is about 200 tps, while fully cross-domain trading sustains approximately 90 tps. Raising the gas limit to 30,000,000 increases throughput from roughly 270 to 360 tps across the tested transaction-ratio range. These results demonstrate that the proposed system achieves secure, regulatable, consistent, and practically efficient cross-domain data trading at scale. A protocol-level Monte Carlo analysis further shows that, with eight voting edge nodes and dual-node cross-verification, the collusive false-acceptance upper bound is 0.522% at a 30% independent malicious-node probability; a separate consistent-hash simulation over 2-16 domains confirms logarithmic lookup growth and decreasing scope remapping as the network scales.

Keywords: data factors; decentralized identity; blockchain; smart contract; incentive mechanism

Received on 17 July 2026, accepted on 03 September 2026, published on 12 October 2026

Copyright © 2026 Kezhi Zhang, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.14076

* Corresponding author. Email: zkg0212101872@163.com

1. Introduction

In the current digital era, data has become one of the most valuable factors of production. It plays a vital role in driving economic growth, boosting scientific and technological

innovation, and enhancing enterprise competitiveness. As a new type of production factor, data require dedicated mechanisms for market exchange [1], machine-learning model pricing [2], auditable outsourcing [3], and systematic valuation [4]. Therefore, it is necessary to build a new market

for data elements, so that data can flow to demanders smoothly and timely to fully realize its value.

Traditional centralized data trading systems have obvious advantages in transaction efficiency, such as enabling users to quickly search for and obtain required data, conducting data transactions efficiently, and facilitating maintenance and management. They are commonly built on cloud storage and centralized business systems, and mature distributed-system technologies can support high-frequency, low-latency transactions and highly scalable data-management services. However, under the centralized paradigm, inadequate security and supervision measures can easily lead to data leakage and abuse, violating users' data rights and interests [5]. To tackle these problems, emerging decentralized data trading systems adopt blockchain technology to provide tamper-proof and traceable security guarantees and implement transaction logic via smart contracts to protect the security and privacy of data assets and transactions [6]. While the decentralized model is generally less efficient than the centralized model in transaction processing and data management, it offers stronger guarantees of transparency and security.

However, even decentralized data trading systems built with blockchain technology still face a series of problems when handling cross-domain data transactions. To ensure transactions comply with system rules and maintain secure and controllable data trading, existing data trading systems generally require both parties to register accounts within the same system, and all purchased data assets must also be registered on that specific system. As a result, cross-regional, cross-industry and cross-border data transactions suffer from overly complex workflows and significant limitations. This not only hinders the development of a unified national market for data elements but also restricts the global sharing and utilization of data. In cross-domain data transaction scenarios, the current architecture of data trading systems is confronted with three core challenges.

1) The consistency of cross-domain identity resolution results. Mutual trust between the two parties of a data transaction is the foundation for smooth transactions. In data transactions, it is essential to establish trust in the identities of both participants and reach a consensus regarding the traded data assets. In cross-domain identity resolution mechanisms, nodes in two different domains can only access blockchain nodes in the other domain in a restricted manner to retrieve on-chain information [7]. This creates information asymmetry between cross-domain nodes, making them vulnerable to phishing attacks and man-in-the-middle attacks. Additionally, caching mechanisms may cause inconsistent cross-domain identity resolution results, whereby identity identifiers are resolved to outdated identity information or compromised identities with leaked private keys. Such inconsistencies can lead to the theft of funds and data assets on the data trading system, or result in buyers receiving low-quality data assets that do not match prior agreements [8].

2) The availability of cross-domain identity resolution services. A decentralized data trading system consists of participants organized in a decentralized manner. As an integral part of this system, cross-domain identity resolution

services are voluntarily operated and provided by participants [9]. Guaranteeing service availability involves two key aspects: ensuring the accuracy and real-time performance of participants' resolution services and maintaining a sufficient number of service providers to support the dynamically changing cross-domain resolution demands of the system [10].

3) Security and compliance issues of cross-domain data trading systems. First, direct cross-domain communication is difficult, making it challenging for the two domains to negotiate a single cross-domain data transaction. If centralized cross-domain communication services are adopted, a single point of failure will disrupt ongoing cross-domain data transactions, or even cause abnormal termination, resulting in losses of funds and data assets. In more severe cases, buyers may deceive sellers by falsely claiming payment; as sellers cannot directly verify the payment status, data may be stolen. Similarly, sellers may also defraud buyers of funds in the same way. Second, the lack of a cross-domain identity mechanism prevents smart contracts from conducting supervision based on the attributes of transaction participants and data assets. The identity mechanism of existing data trading systems only relies on identity identifiers and public identity attributes, failing to meet the diverse compliance requirements of laws and regulations across different regions and jurisdictions [11]. In a cross-domain data transaction, regulators need to break down domain boundaries, conduct supervision based on attribute data of both transaction parties, identify irregular behaviors, and verify the compliance of the entire transaction.

The main contributions of this paper are summarized in four aspects:

1. A secure and efficient cross-domain resolution protocol for decentralized identity (DID) identifiers is designed for decentralized identity technology. It enables decentralized data trading systems to build a unified identity layer in a decentralized manner, ensuring cross-domain consistency of both user identities and data asset identities.

2. An incentive mechanism is developed for the cross-domain identity resolution protocol, to guarantee the stable operation of the protocol and the availability of identity resolution services for cross-domain data transactions.

3. A secure and regulatable decentralized data trading system is constructed based on the above identity resolution protocol and incentive mechanism. The system guarantees the atomicity of cross-domain data transactions and supports fine-grained regulation at the identity attribute level for cross-domain data transactions.

4. The proposed system is implemented and verified in terms of feasibility and efficiency, providing a new technical approach for cross-domain data circulation.

2. Related Work

With the rapid development of the digital economy, data security governance has become a critical requirement for cross-domain data circulation, requiring technical mechanisms that jointly support privacy protection,

controllability, auditability, and regulatory compliance [12]. Technical solutions play a critical role throughout the entire data lifecycle, especially in the phase of data value realization. For example, searchable encryption enables practical retrieval over encrypted data without exposing plaintext queries or records [13-14]. Privacy-preserving public auditing can verify outsourced data integrity while protecting sensitive information from third-party auditors [15]. To reduce the efficiency loss introduced by security and audit controls, blockchain-based cross-domain security mechanisms and privacy computing can be combined to support secure data sharing and accountable data use [16]. The application and advancement of these technologies provide robust support for data security governance and serve as a key driver for the sound and sustainable development of the data industry.

However, building a data trading system requires multiple complementary security technologies across identity, communication, storage, and sharing. Blockchain-based cross-domain authentication can establish trust between entities from different administrative domains [17]. At the communication layer, physical-layer message-authentication techniques provide complementary mechanisms for authenticating transmitted messages [18-19]. Blockchain-based decentralized frameworks can support auditable task coordination and decentralized execution [20]. Proxy re-encryption enables a data owner to delegate access to encrypted data without exposing plaintext to the proxy [21], while attribute-based encryption with verifiable outsourced decryption supports fine-grained encrypted-data access with verifiable delegated decryption [22]. Broader cryptographic cloud-storage research further summarizes confidentiality, integrity, access-control, and secure sharing techniques for outsourced data [23]. These mechanisms address important components of secure data circulation, but they do not by themselves constitute an integrated cross-domain data-trading architecture.

Existing blockchain-based identity and authentication schemes have addressed scalable identity management, privacy-preserving identity proofs, and authentication in specific application domains [24-26]. However, these works are not designed as end-to-end cross-domain data-trading architectures and therefore do not jointly address identity-resolution consistency, resolver availability, transaction atomicity, and fine-grained regulation.

An identity system is a framework for managing and verifying the identities of individuals and entities, providing services including identity registration, authentication, management and authorization. Trading parties authenticate

each other via the identity system, and buyers establish trust in the metadata of the data, which forms the foundation of data transactions. Existing blockchain marketplaces employ heterogeneous identity representations. Agora and edge-cloud trading schemes mainly rely on blockchain accounts, public keys, and cryptographic bindings [27-28], while TrailChain focuses on data-ownership registration and traceability across multiple blockchain-enabled marketplaces [29]. FAST DATA supports decentralized actor onboarding and verification in an IIoT marketplace [30], and the subscription-based IoT trading infrastructure in [31] uses decentralized-ledger identities within its trading workflow. These approaches improve decentralized trust, but they do not provide a unified W3C-style cross-domain DID resolution layer with explicit DID-document freshness verification, resolver-availability guarantees, and credential-based attribute supervision. Without such a unified resolution mechanism, identity updates or resolver failures can leave external domains with stale or unverifiable identity state.

Data trading systems are subject to different regulatory rules depending on the industry, region and jurisdiction where they operate, and therefore need to adapt to compliance requirements across diverse scenarios. Cross-domain data transactions face stricter supervision than intra-domain transactions and must satisfy compliance rules in both relevant domains simultaneously. Many existing solutions embed transaction rules in smart contracts, but cross-domain attribute-level regulation additionally requires trusted transmission and verification of identity information. Reference [32] surveys general cross-chain mechanisms, reference [33] provides privacy-preserving secure data exchange in a cloud marketplace, and reference [34] constructs a blockchain-based trading system for big data. However, these approaches do not jointly provide a unified cross-domain identity-resolution layer and an attribute-level regulatory mechanism. Representative data-market systems [27,30,31,34] focus on privacy, fair exchange, decentralized trading, or quality-aware transaction execution; their identity mechanisms are platform-specific and are not designed to carry verifiable DID/VC attributes across heterogeneous domains. This gap motivates the unified cross-domain DID and credential-verification layer developed in this work.

To make the distinction from representative prior systems explicit, Table 1 compares the mechanisms at the system level. Because the cited studies use different chains, hardware, and workloads, their performance numbers are not normalized into a single throughput ranking; instead, the table reports the type of performance evidence available and uses the same cross-domain criteria for all systems.

Table 1. Comparison with representative blockchain data-trading and cross-domain identity systems

Study	Identity consistency / service availability	Atomicity and regulation	Performance evidence
Shen et al. [7]	Cross-domain device authentication; no DID cache-consistency or resolver-incentive mechanism.	Authentication-oriented; not an end-to-end data-trading regulation architecture.	Authentication latency/overhead under a different IIoT task.

Study	Identity consistency / service availability	Atomicity and regulation	Performance evidence
Agora [27]	No unified cross-domain DID layer; no cross-domain resolver availability mechanism.	Provides payment atomicity, data privacy, and output verifiability; no cross-jurisdiction DID/VC attribute regulation.	Ethereum computation overhead and deployment cost; no directly comparable cross-domain trading tps.
FAST DATA [30]	Decentralized actor identity and storage; no explicit cross-domain DID consistency/refresh protocol.	Fair payment and data delivery; no attribute-level cross-domain regulatory module.	Throughput, overhead, and scalability on Hyperledger Fabric under a different marketplace workload.
DataTBC [34]	Blockchain-based data trading without a unified cross-domain DID consistency layer.	Smart-contract matching/negotiation and quality-aware trading; no DID/VC cross-domain regulation.	Security, usability, and efficiency evaluation; end-to-end cross-domain tps not reported.
Proposed system	Merkle-root-verified cross-domain DID resolution, TTL/LRU cache refresh, incentives, redundancy, and malicious-node cross-verification.	Atomic cross-domain state coordination plus DID/VC attribute-level regulation in both domains.	82 ms cross-domain DID resolution at 70% cache hit; about 90 tps when all trades are cross-domain at GasLimit = 10,000,000.

Note: Performance evidence is reported in the native workload of each cited study; values from heterogeneous platforms are not converted into artificial like-for-like tps.

3. System Overview

3.1 System Model

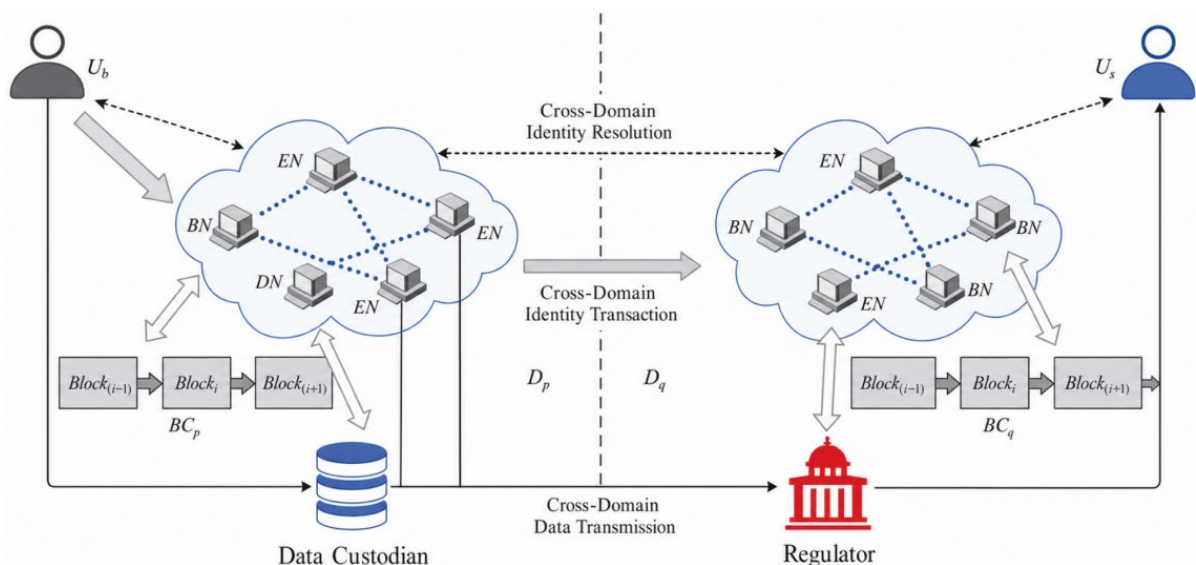


Figure 1. System model

The design of this paper refers to the architecture of existing data transaction models. The proposed decentralized data trading system involves four types of participants: Users, Blockchain Nodes, Data Custody Nodes, and Edge Nodes. As illustrated in Figure 1, the cross-domain data transaction scenario includes two decentralized data trading systems located in separate domains, denoted as D_p and D_q .

Notation is standardized as follows: subscripts p and q denote the local/buyer and external/seller domains,

respectively; BC_p/BC_q denote the corresponding blockchains, EN_p/EN_q their edge nodes, and SC_p/SC_q their transaction smart contracts. The same domain-subscript convention is used consistently in the formulas, protocol descriptions, and pseudocode.

1. Users. Users are buyers and sellers of data assets. Sellers upload data assets to the data trading system and set corresponding prices; buyers select target data assets, complete payments, and download the data after the

transaction is verified by the system. We use U_s to represent sellers and U_b to represent buyers.

2. Blockchain Nodes. In the decentralized data trading system, blockchain nodes verify and execute data asset transactions. The transaction logic, including identity verification, compliance verification, and fund verification, is implemented via on-chain smart contracts. We use BN to denote blockchain nodes.

3. Data Custody Nodes. Due to the inherent transparency and redundant storage characteristics of blockchains, raw data content cannot be directly stored on the blockchain. In the decentralized data trading system, the actual data assets are stored on off-chain nodes, while their hash values are anchored on the blockchain to guarantee data integrity during storage and transmission. Sellers store data assets on data

custody nodes, and buyers retrieve data from these nodes after completing purchases. We use DN to denote data custody nodes.

4. Edge Nodes. To ensure the security of the decentralized data trading system and data assets, internal system participants including users, blockchain nodes and data custody nodes shall restrict external access permissions. To enable cross-domain data transactions, edge nodes are deployed to forward cross-domain requests and data transmissions, as well as receive incoming external requests and data transmissions. An edge node may function simultaneously as a blockchain node or operate solely as a server that accesses blockchain nodes, forwards requests, and receives responses. We use EN to denote edge nodes.

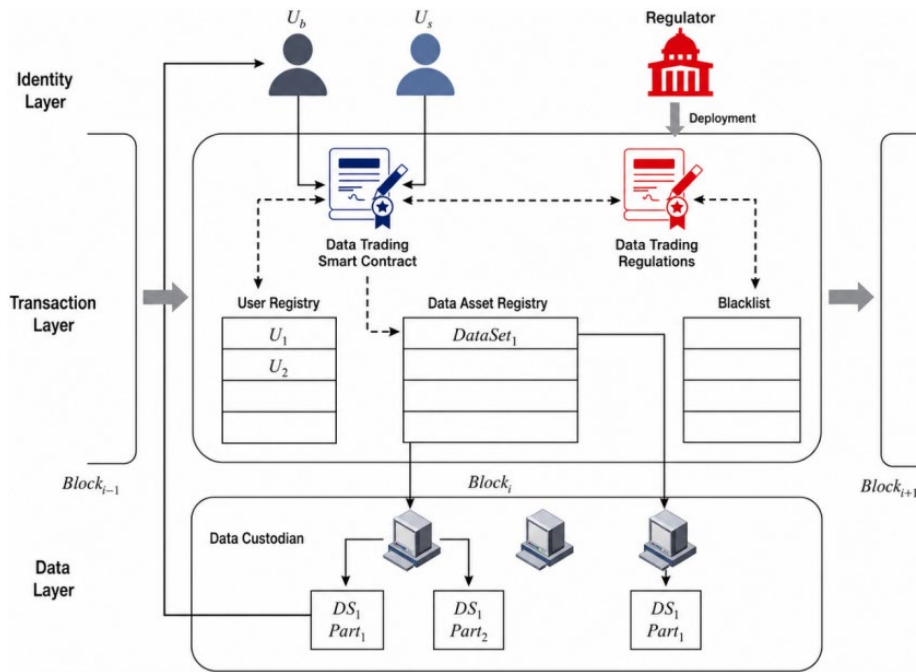


Figure 2. System architecture

As shown in Figure 2, the above system participants divide the decentralized data trading system into three layers: the identity layer, the transaction layer, and the data layer. The identity layer manages the identifiers and attributes of users and data assets. A trusted identity authentication mechanism serves as the security foundation for data transactions and data transmission, as well as the basis for data transaction regulation. The transaction layer executes data transactions, including identity verification, compliance verification, fund verification, etc., and determines whether to approve transactions according to the execution results. The data layer is responsible for data storage and transmission. In cross-domain data transaction scenarios, these three layers interact with external systems via cross-domain mechanisms to achieve cross-domain identity authentication, transaction execution, and data transmission.

3.2 Design Goals

1. Consistency of cross-domain identity resolution results. Consistency in cross-domain resolution scenarios means that identity resolution results remain identical across all domains. At time t , if user U_b in domain D_p and user U_s in domain D_q simultaneously initiate resolution requests for the same DID identifier in domain D_p , the two resolution results must be identical.

2. Availability of cross-domain identity resolution services. In cross-domain resolution scenarios, availability means that identity resolution requests submitted by users both inside and outside the domain can receive correct resolution results within a specified time frame.

3. Atomicity of cross-domain data transactions. Atomicity of cross-domain data transactions includes two core

requirements. In the data purchase phase, data ownership transfer and data transmission shall occur if and only if full payment is completed. In the data delivery phase, once payment is confirmed, the buyer must obtain the legal ownership and complete content of the agreed data assets.

4. Fine-grained regulation of cross-domain data transactions. For cross-domain data transactions, both participating domains shall support regulatory supervision based on the identity attributes of transaction participants and the metadata of data assets.

4. DID Cross-Domain Resolution Protocol

To guarantee the consistency of cross-domain identity resolution results, this section proposes a cross-domain identity resolution protocol. An identity resolution protocol is a communication protocol for resolving and verifying identity information. It defines a set of standardized rules and methods to enable interoperability across different systems for identity verification and resolution. The DID framework ensures

intra-domain resolution consistency, and the proposed protocol is built upon the DID framework to achieve consistent cross-domain identity resolution. To address inconsistent cross-domain identity resolution caused by caching mechanisms, a consistency-preserving caching mechanism is designed to improve cross-domain resolution efficiency while maintaining global consistency.

4.1 DID Framework

The identity layer of the system is implemented based on the DID framework. DID-based self-sovereign identity replaces platform-controlled accounts with cryptographically verifiable identifiers and user-controlled identity records, thereby supporting decentralized identity management and credential exchange [35]. The DID framework ensures the availability and consistency of intra-domain identity resolution services. As the identity layer of the decentralized data trading system, it provides identity management services for users and data assets.

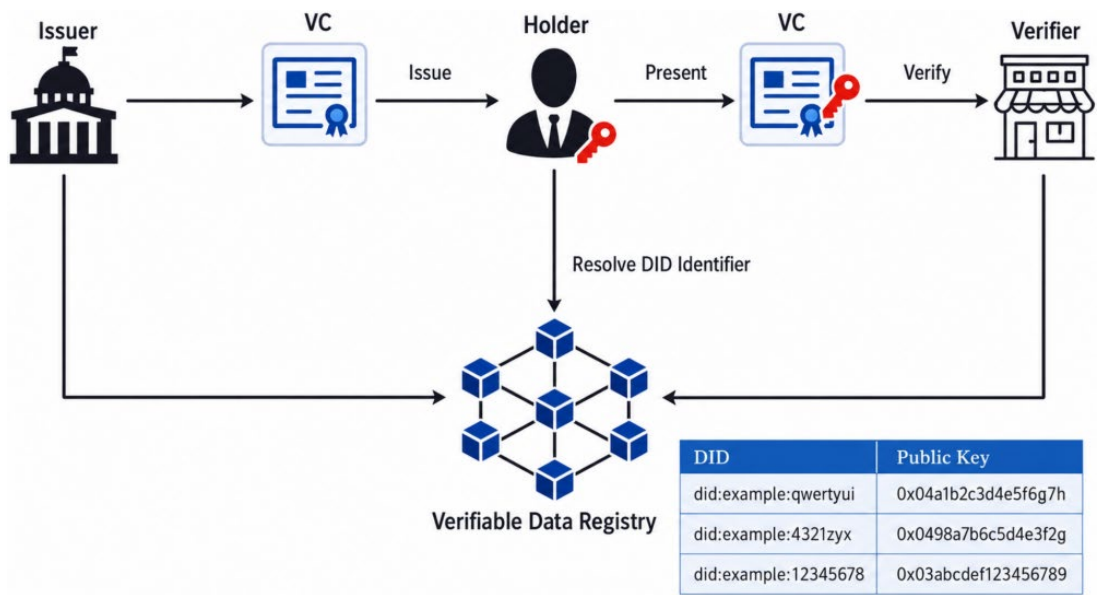


Figure 3. Decentralized identity framework

The DID framework is built upon a series of technologies and standards, including core components such as DID identifiers, DID documents, verifiable credentials (VC), and verifiable data registries. These components are widely regarded as the technical foundation of blockchain-enabled self-sovereign identity systems [36]. As illustrated in Figure 3, they provide users with an efficient and flexible mechanism for DID identifier resolution and identity attribute verification.

The DID identifier resolution mechanism involves two core data structures (DID identifiers and DID documents) as well as the verifiable data registry module. Users interact with the blockchain-based verifiable data registry to retrieve the

corresponding DID document from a DID identifier using on-chain information. Different DID methods define method-specific identifier syntax and create, read, update, and revoke operations. Academic surveys show that this diversity improves deployment flexibility but also introduces interoperability and resolution-consistency challenges across identity ecosystems [37].

The DID identity attribute verification mechanism is based on DID documents and verifiable credential (VC) data structures, with three core roles: the Issuer, the Holder, and the Verifier. The Issuer is an authoritative entity that issues verifiable credentials and signs them with its public key based on verified information about the Holder. The Holder, as the

credential owner, securely stores the credentials and presents them when required. The Verifier receives the credentials provided by the Holder and verifies the signature and content using information from the DID document (such as the Issuer's public key), to confirm the authenticity of the credential and the identity attributes of the Holder.

The verification process first resolves the DID document from the DID identifier, retrieving public keys and encryption schemes to ensure secure communication. The Issuer generates and signs a verifiable credential, which is then received and stored by the Holder. When the Holder needs to prove their identity, they present the credential to the Verifier. The Verifier retrieves the Issuer's public key from the DID document, verifies the credential signature and content, checks the credential's integrity, validity, and the Issuer's authority, and confirms the identity attributes of the Holder accordingly.

All core DID identity operations, including identity resolution, authentication, and registration, rely on direct access by DID users to the intra-domain blockchain. Existing DID ecosystems do not provide a universally trusted cross-domain identity resolution service, and many implementations depend on gateways or resolvers that bridge method-specific ledgers. Recent studies analyze DID method design, resolution interfaces, service endpoints, and interoperability requirements [38], while blockchain-assisted decentralized authentication in IoT environments demonstrates the value of ledger-backed credentials and dynamic trust establishment [39]. Nevertheless, when gateways act as third-party cross-domain services, fundamental issues remain, including service outages, stale or incorrect resolution results, malicious responses, and weak accountability.

4.2 Cross-Domain Identity Resolution Protocol

Since external nodes cannot directly access intra-domain blockchain nodes, existing DID resolution verification schemes cannot apply Merkle proofs to verify the correctness and timeliness of cross-domain identity resolution results. Accordingly, it is essential to design a cross-domain identity resolution protocol that guarantees global consistency of identity resolution.

Several cross-domain DID resolution schemes already exist, yet they rely on centralized cross-domain resolvers, which undermines the decentralized nature of the DID framework. To construct a cross-domain identity mechanism, this work builds upon the DID framework and the overall system architecture proposed in this paper.

Based on the proposed data trading system model, two new operations are defined to achieve consistent cross-domain identity resolution. To enable intra-domain users to verify the resolution results of external DID identifiers, the latest root hash of the external blockchain must be recorded on the local blockchain. This operation is performed by edge nodes with cross-domain access permissions and is defined as $BCp.Set(BCq.rootHash)$. Edge nodes initiate or respond to cross-domain requests, such as updating root hashes or resolving DID identifiers. We define cross-domain requests initiated by edge nodes as $ENp.request(BCq.req) \rightarrow resp$.

Using the above operations, a cross-domain identity resolution protocol that guarantees consistent resolution results can be constructed. As illustrated in Figure 4, the mechanism consists of three phases: the pre-resolution data preparation phase, the cross-domain request-and-response phase during resolution, and the post-resolution verification phase for resolution results.

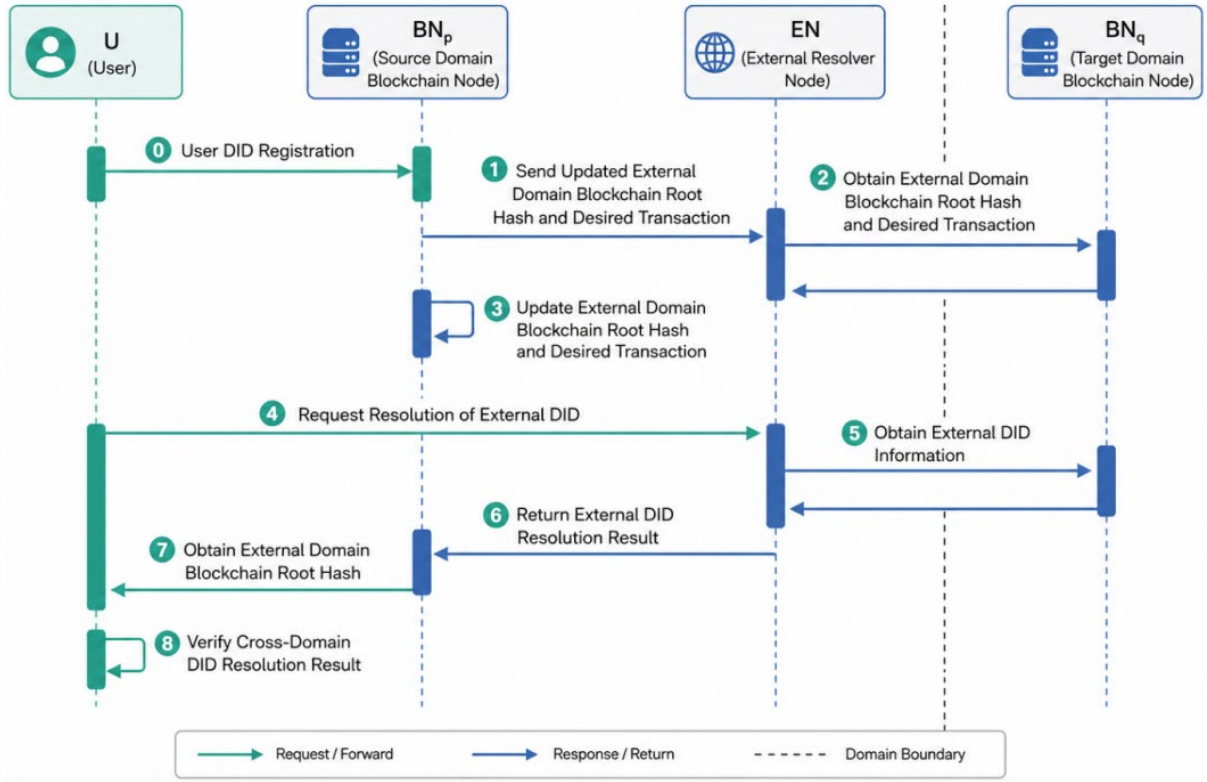


Figure 4. Cross-domain identity resolution protocol

Specific procedures of each phase:

1) Data Preparation Phase

① The blockchain within domain D_p promptly processes registration and modification requests for user DID identifiers and DID documents, ensuring the correctness and real-time validity of intra-domain identity information. User U_b in domain D_p may invoke $BC_p.Set(DID, DIDDoc)$ to update the DID document information on the blockchain according to changes in their identity information.

② The set of intra-domain edge nodes $\{EN_1, EN_2, \dots, EN_n\}$ periodically synchronizes the root hashes of external blockchains for which they are responsible, and records these root hashes onto the local blockchain. Assuming EN_i has access to domain D_q , EN_i periodically sends the request $EN_i.request(BC_q, QueryLatestRootHash) \rightarrow \{BlockNumber, rootHash\}$, and calls $BC_p.Set(BC_q, rootHash)$ to update the recorded root hash of domain D_q on the blockchain.

2) Cross-Domain Request and Response Phase

① User U_b locates the edge node EN_i responsible for the target external domain, sends a cross-domain identity resolution request $U_b.request(EN_i, \{BC_q, DID_s\})$, and waits for a response.

② The edge node forwards the identity resolution request: $EN_i.request(BC_q, \{DID_s\}) \rightarrow \{DIDs, DIDDocs, MerkleProof_s\}$, and performs cross-domain identity resolution.

③ The edge node responds to the user's identity resolution request with the result $\{DID_s, DIDDocs, MerkleProof_s,$

$SigEN_i\}$. The returned DID document and Merkle proof fields are signed with the public key corresponding to the DID identifier of edge node EN_i , to guarantee non-repudiation of the resolution result.

3) Resolution Result Verification Phase

① The user sends the request $BC_p.Get(BC_q, rootHash, blockNumber) \rightarrow rootHash$ to retrieve the resolution result and the root hash value of BC_q from the local blockchain.

② The user first verifies whether the DID document in the resolution result has expired, whether it matches the corresponding DID identifier, and whether it contains a valid signature from the edge node. After passing these checks, the user verifies the Merkle proof of the resolution result against the retrieved root hash value, so as to confirm the integrity of the resolution result.

4.3 Caching Mechanism for Cross-Domain Identity Resolution

Caching can reduce repeated lookups in identity-resolution systems [40], but TTL-based distributed caches may temporarily serve stale copies after authoritative records are updated [41]. In a cross-domain resolver, cache freshness must therefore be checked explicitly. In the proposed protocol, the block number, TTL, Merkle proof, and latest externally committed root hash are jointly used to reject stale or tampered resolution results.

To maintain resolution consistency, each cache record contains six core fields: DID identifier, DID document, Merkle Proof, block number, time to live (TTL), and least recently used (LRU) timestamp. In addition to the DID identifier and corresponding DID document, the cache stores the Merkle Proof and relevant block number, so that users can verify the integrity of resolution results.

The TTL of a cached entry is set to the minimum value among the TTL specified by the DID document, the cross-domain agreed cache TTL, and the global system TTL, to

ensure data timeliness. Each record also stores its last access timestamp to assess data popularity. Using both TTL and LRU fields, the system can evict expired entries, trigger updates, and determine cache replacement priorities under limited storage capacity.

Based on this record structure, we design an identity resolution cache update algorithm (as illustrated in Figure 5), which actively refreshes the local cache by referencing the latest on-chain root hash values and local cached data.

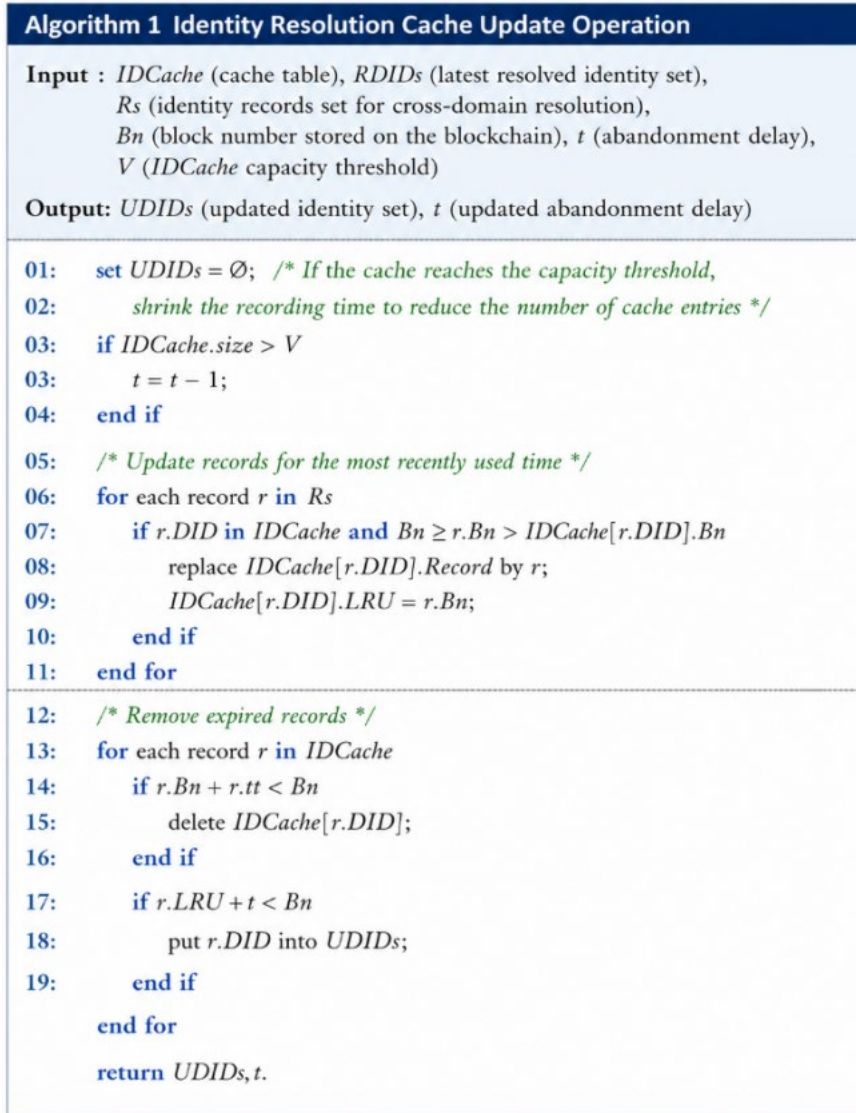


Figure 5. Pseudocode of the identity resolution cache update operation

Algorithm 1 is invoked after each new block is confirmed on the blockchain and actively updates cached resolution results based on the returned *UDIDs*. In addition to active updates, the cache also supports passive updates. A cross-domain resolution request typically involves two edge nodes located in different domains. For example, in a cross-domain

resolution request, EN_p from domain D_p requests the resolution of DID_t from EN_q in domain D_q . Once the corresponding DID document information of DID_t is modified, EN_p will actively send an update request to EN_q to refresh the cached data of DID_t . These two update strategies (active and passive) reduce the frequency of repeated cross-

domain resolution requests, improve overall resolution efficiency, and guarantee the timeliness and accuracy of cached data.

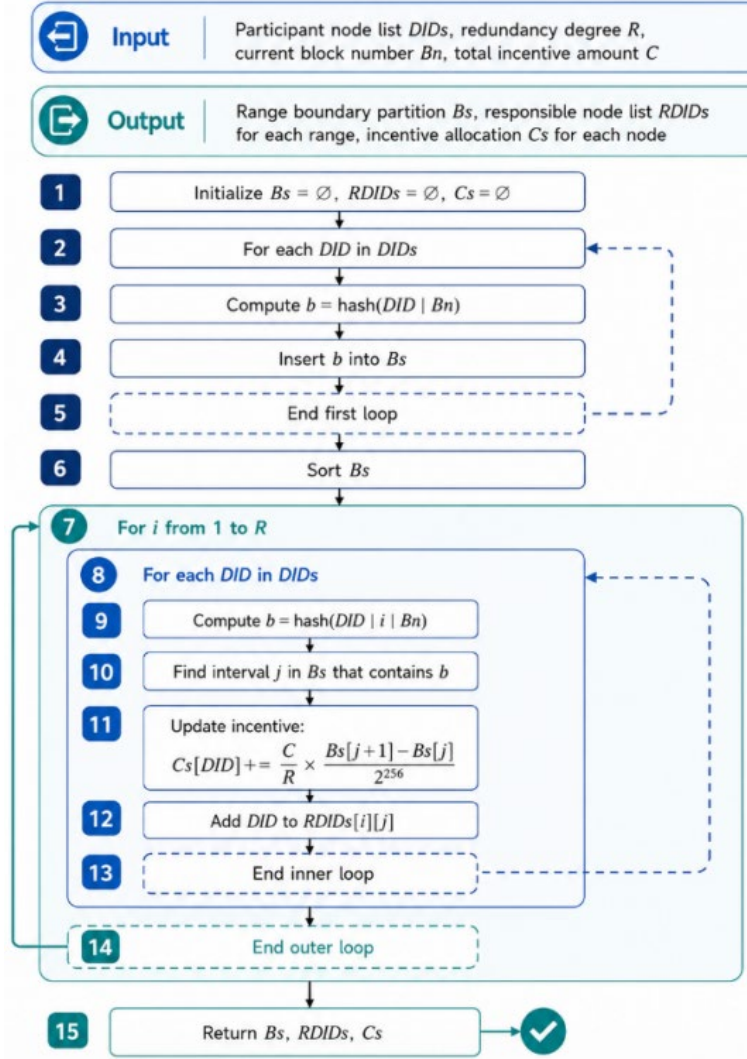
Complexity analysis of Algorithm 1. Let C denote the number of cache entries and $|R_s|$ the number of newly returned or refreshed identity records. With a hash-table cache, membership tests and replacements are $O(1)$ on average. The first update loop therefore costs $O(|R_s|)$, while expiration/LRU inspection costs $O(C)$, giving $O(|R_s| + C)$ time and $O(C)$ cache space, with at most $O(C)$ temporary UDID identifiers in the worst case. The passive path performs an $O(1)$ cache lookup and triggers a cross-domain refresh only for the DID that was actually modified. Consequently, the active-passive strategy does not scan the external identity space and confines maintenance work to the local cache and changed records, which preserves the lightweight property as the global DID population grows.

5. Availability Guarantee Mechanism for Cross-Domain Identity Resolution

The cross-domain identity resolution and caching mechanism can ensure the consistency of cross-domain identity resolution, as well as the integrity and real-time performance of resolution results obtained by users. This mechanism relies on the correctly and timely updated root hashes of external blockchains. If edge nodes fail to update or incorrectly update the root hashes, the cross-domain identity resolution service cannot function properly even with a consistent cross-domain identity resolution protocol. In addition, edge nodes with cross-domain access capabilities can freely join and exit the cross-domain identity resolution process, which makes it impossible to guarantee that the number of participating nodes meets the minimum requirement for stable operation of the cross-domain resolution service, nor can the availability of participating nodes be ensured.

Therefore, a sound availability guarantee mechanism is required to organize edge nodes, detect and punish malicious and inactive edge nodes, and reward edge nodes that provide normal services, so as to ensure the stable operation of the cross-domain identity resolution service. The availability guarantee mechanism for cross-domain identity resolution proposed in this section consists of two parts: 1) The resolution scope division mechanism and incentive mechanism for edge nodes. This mechanism allocates resolution scopes to participating nodes, rewards edge nodes involved in cross-domain identity resolution according to their workload and motivates edge nodes to actively conduct external domain identity resolution within their authorized scope. 2) The detection and punishment mechanism for malicious nodes. This mechanism identifies abnormal resolution behaviors and detects malicious nodes in a timely manner, so that corresponding measures can be taken to punish malicious nodes and prevent malicious behaviors.

In the cross-domain resolution system, each edge node of an external domain has a unique DID identifier, and these identifiers form the set $EN = \{EN_1, EN_2, \dots, EN_n\}$. Meanwhile, the DID identifiers of the external domain q to be resolved also form the set $Q = \{Q_1, Q_2, \dots, Q_m\}$. By hashing these DID identifiers, they can be uniformly distributed across an integer space of size 2256. As shown in Algorithm 2, the resolution scope division mechanism adopts the consistent hashing algorithm, which maps the hashed value $\text{Hash}(EN_i)$ of EN_i onto a circular integer space. Consistent hashing features excellent load balancing capability and scalability, facilitating dynamic node joining and removal. Only minimal scope adjustments are required when such operations occur, thereby reducing maintenance costs. On this hash ring, if $\text{Hash}(EN_i)$ is adjacent to $\text{Hash}(EN_j)$, then all $\forall Q_k \in [\text{Hash}(EN_i), \text{Hash}(EN_j))$ shall be resolved by edge node EN_i .

Algorithm 2. Range Partitioning and Incentive Mechanism**Figure 6.** Pseudocode of the range-partitioning and incentive mechanism

To ensure the availability and security of the resolution process, the system sets a redundancy parameter R in Algorithm 2 and assigns multiple edge nodes to each resolution interval. Users can verify the correctness of resolution results by comparing the responses returned by these nodes. In addition, to mitigate the risk of degraded security or potential attacks when nodes undertake long-term resolution tasks for a specific interval, the system periodically updates the resolution scopes assigned to nodes. This update process is implemented in Algorithm 2 by using the current block number $BlockNumber$ and EN_i as inputs to the hash function, so as to reassign the positions of edge nodes on the hash ring on a regular basis.

On this basis, the blockchain allocates rewards according to the size of the resolution scope assigned to each node. Nodes responsible for a larger resolution scope handle more resolution requests and thus receive a higher proportion of rewards, which further encourages node participation and

performance optimization. This mechanism not only improves system efficiency but also enhances the overall security and reliability of the network.

There are two feasible attack modes for malicious nodes in the cross-domain identity resolution protocol. The first is uploading incorrect root hash values of external blockchains, misleading users into believing that erroneous resolution results are valid on the blockchain. The second is returning resolution results generated from outdated identity information even after the external domain identity information has been updated.

For the first type of attack, the system prevents such attacks by introducing a voting mechanism for root hash updates. All participating edge nodes are authorized to update the root hash values of external blockchains and receive corresponding rewards. If a malicious node updates an incorrect root hash value, multiple candidate root hash values will emerge. The participating nodes then conduct a voting

process to select the correct root hash value and penalize the malicious node. When malicious nodes account for less than half of all participating edge nodes, this mechanism can

guarantee the correctness and timeliness of the external blockchain root hash values.

Algorithm 3. Secure Resolution Mechanism

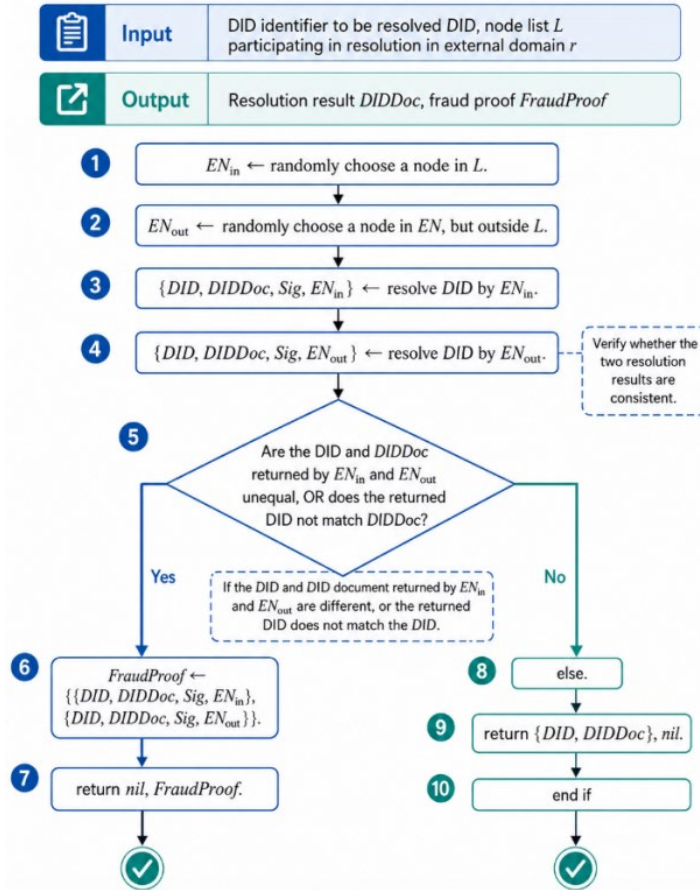


Figure 7. Pseudocode of the secure resolution mechanism

When a user initiates a resolution request, a cross-verification method is adopted in this mechanism to improve security and reduce the probability of requests being routed to malicious nodes. As shown in Figure 7, the user first randomly sends a resolution request to the edge node EN_{in} responsible for the range of the target DID identifier to be resolved. Then the user randomly selects another edge node EN_{out} outside this resolution range (to prevent malicious nodes from increasing their probability of being assigned a resolution scope by registering multiple DID identifiers) and sends a resolution request to it. EN_{out} will request resolution from the external domain and obtain the corresponding result.

The user can verify the correctness of the resolution result by comparing the results returned by EN_{in} and EN_{out} . If EN_{in} is a malicious node, the user submits the resolution result signed by EN_{in} to the blockchain as evidence. The blockchain will reward the reporting user and penalize the malicious node EN_{in} . Meanwhile, other participating edge nodes can also act as pseudo-users to verify resolution results from other

edge nodes via the above method. Through frequent recursive queries and verification, malicious nodes can be rapidly detected, penalized and eliminated.

Security analysis against collusion and cache poisoning. The root-hash voting guarantee in Section 5 is conditional on an honest majority among the authorized voting edge-node identities. Under that condition, a forged DID document or a poisoned cache entry cannot pass verification because the returned document, edge-node signature, block number, and Merkle proof must all be consistent with the latest externally committed root hash. A stale but otherwise valid cached record is detected by the block-number/TTL checks and is forced into re-resolution; thus, cache poisoning can increase communication or latency, but it cannot silently change the accepted identity state while the latest root hash remains honest. For collusive edge nodes, the dual-node check requires the in-range and out-of-range responders to agree, so collusion is effective only when both selected responders are malicious and the root-hash voting layer is also compromised.

The quantitative probability of this joint event is evaluated in Section 7.4.1. This analysis assumes that each voting identity corresponds to an authorized service node; Sybil-resistant admission or stake-based registration remains a deployment-layer requirement rather than a property of the hash-ring algorithm itself.

6. Design of a Cross-Domain Data Trading System

Blockchain security is a fundamental requirement for cross-domain data trading [42]. Beyond general ledger security, the proposed system must additionally guarantee transaction atomicity and fine-grained regulation. The cross-domain identity resolution mechanism supports these goals by transmitting externally committed states with integrity and

freshness checks. Attribute-based access-control mechanisms can support fine-grained authorization over identity-related attributes [43]. In the proposed architecture, such policies are instantiated using attributes derived from DID documents and verifiable credentials, allowing each domain to enforce its own compliance rules.

6.1 Modular Decomposition of the Data Trading Layer

This section first defines the modules to be designed in the data layer to support cross-domain data transactions. The data trading layer is implemented based on blockchain, so its modules are developed via smart contracts. All modules defined in Figure 8 are realized with smart contracts and deployed to run on the blockchain.

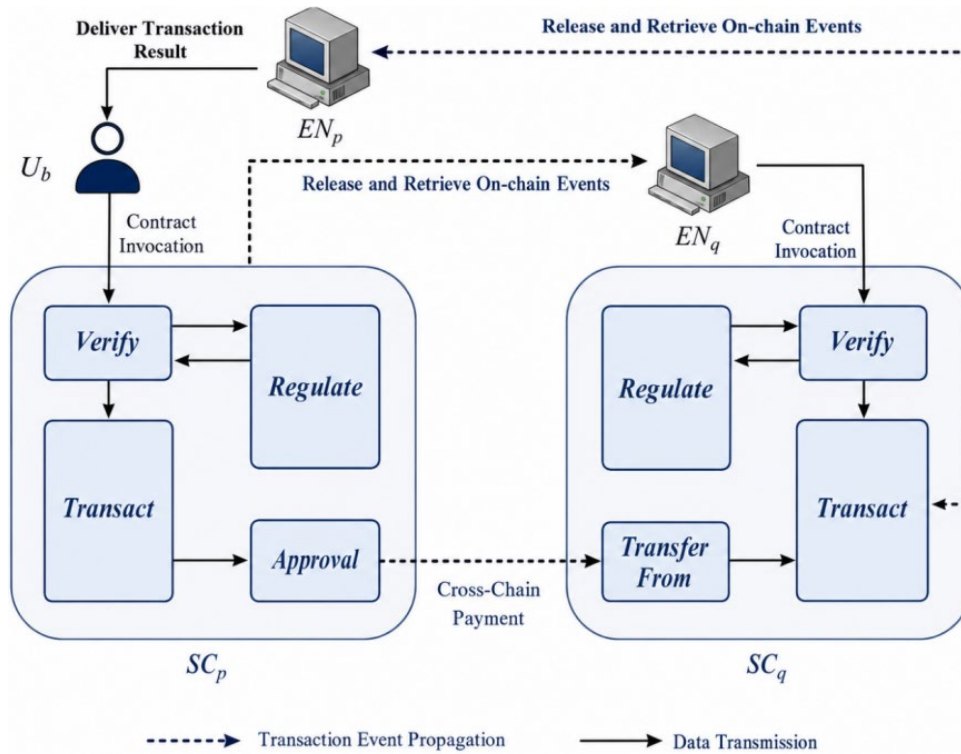


Figure 8. Cross-domain data-trading smart-contract structure

1) Cross-Chain Information Verification Module This module verifies information transmitted from external domains, such as cross-domain resolved identities and certain states on external blockchains. It adopts the scheme proposed in Section 4 to ensure the integrity and real-time performance of transmitted data. The function of this module is defined as:

$$SC_p.Verify(BC_q, rootHash, Key, Value) \rightarrow Bool \quad (1)$$

2) Regulation Module The regulation module is developed in compliance with the local laws and regulations applicable

to the data trading system and supervises the identities of both trading parties and the traded data assets. It enables fine-grained regulation of cross-border data assets by retrieving information from DID documents resolved via the DID identifiers of data assets. It can also specify identity attributes and require both trading parties to submit corresponding verifiable credentials to implement attribute-based regulation. The function of this module is defined as:

$$SC_p.Regulate(DID, [Attr, statement]) \rightarrow Bool \quad (2)$$

3) Transaction Verification Module This module executes transaction logic, including verifying whether the payment amount from the buyer complies with the price defined by the seller, additional transaction terms, and special agreements between the two parties. It has no return value but emits a set of smart contract events to record the final transaction results permanently on the blockchain. The function of this module is defined as:

$$SC_p.T \text{ransact} ([\text{param}, \text{statement}]) \Rightarrow \text{Event}(\text{BuySuccess}, [\text{data}]) \quad (3)$$

4) Cross-Chain Payment Module There are multiple existing cross-chain payment schemes, and this paper does not delve into the detailed technical implementation of cross-chain payment. The core function of cross-chain payment can be abstracted as transferring funds from an account on one blockchain to an account on another blockchain. Accordingly, this module uses Ethereum-compatible fungible-token transfer interfaces. Empirical analyses of ERC-20 contracts show that standardized token interfaces improve interoperability but require strict conformance and behavior validation [44]. The function is defined as follows:

$$SC_p.Approve(DID_b, DID_s, \text{num}) \Rightarrow \text{Event}(\text{TransferSuccess}), \\ SC_q.TransferFrom(DID_b, DID_s, \text{num}) \quad (4)$$

6.2 Cross-Domain Data Trading Mechanism

The cross-domain data asset trading process is illustrated in Figure 8, which consists of two phases executed on the blockchain smart contracts of the buyer domain D_p and the seller domain D_q , respectively.

1) Buyer-side Contract Procedures

① After the buyer user U_b obtains full information about external data assets via the cross-domain identity mechanism and decides to purchase the data, the user U_b calls the cross-chain information verification module to verify the information of the target asset and the seller's identity.

$$SC_p.Verify(BC_q, \text{rootHash}, DID_s, DIDDoc_c) \& \\ SC_p.Verify(BC_q, \text{rootHash}, DID_b, DIDDoc_b) \quad (5)$$

② Once the identity verification passes, the regulation module is invoked to supervise the transaction. Attribute information can be obtained either by parsing DID documents or by actively submitting verifiable credentials.

$$SC_p.Regulate(DID_s, \{DIDDoc_s, VC_1, \dots, VC_n\}, \text{statement}) \& \\ SC_p.Regulate(DID_b, \{DIDDoc_b, VC_1, \dots, VC_n\}, \text{statement}) \& SC_p.Regulate(DID_b, \{DIDDoc_b\}, \text{statement}) \quad (6)$$

③ After regulatory compliance check, the transaction verification module is called with essential transaction parameters, including the price of the target data and the payment amount.

$$SC_p.Transact(\{DID_b, DIDDoc_b, token, number, \dots\}, \text{statement}) \quad (7)$$

④ After the transaction verification module confirms the transaction, the buyer system approves the transaction, and $SC_p.Approve(DID_b, DID_s, \text{num})$ is invoked to complete the cross-chain transfer.

⑤ The local edge node EN_b collects events emitted by the transaction verification module, including the transaction approval proof BuySuccess , as well as the payment confirmation event TransferSuccess emitted by the cross-chain payment module. EN_b constructs a transaction containing $\{\text{BuySuccess}, \text{TransferSuccess}, DID_b, DID_s, \text{Sig}_b\}$ and sends it to the blockchain BC_q to invoke the seller smart contract SC_q .

2) Seller-side Contract Procedures

① In addition to verifying the external identity information, the cross-chain information verification module also verifies whether the transaction has been approved and whether the cross-chain payment has been completed successfully.

$$SC_q.Verify(BC_p, \text{rootHash}, \text{tx}, \text{BuySuccess}) \& \\ SC_q.Verify(BC_p, \text{rootHash}, \text{tx}, \text{TransferSuccess}) \quad (8)$$

② The regulation module is basically consistent with that of the buyer contract, while the statement is adjusted according to the laws and regulations applicable to the current system.

③ The transaction verification module verifies whether the price of the digital asset matches the cross-chain payment amount, and performs other verifications in accordance with the system transaction rules.

④ After the transaction verification module confirms the transaction, the seller system approves the transaction, and $SC_q.TransferFrom(DID_b, DID_s, \text{num})$ is invoked to complete the cross-chain transfer.

⑤ The edge node EN_p of domain BC_p sends an external request $EN_p.request(BC_q, \{\text{GetSellState}\})$ to retrieve events emitted by the transaction verification module of domain BC_q . The response includes the transaction approval proof, the root hash of the data block, the list of custodian nodes storing the data assets, i.e., $\{\text{SellSuccess}, \text{rootHash}, \{\text{DN}_d\}\}$, as well as events emitted by the cross-chain payment module. EN_p forwards the cross-domain response to the user to complete the cross-chain transaction. The user can then initiate an external data transmission request with the provided proof.

In this mechanism, the regulatory compliance of cross-domain data transactions is guaranteed by the smart contract regulation module and the identity layer. The identity layer resolves DID identifiers to obtain public attributes of data assets and both trading parties. Additionally, the two parties can submit more detailed identity attributes to the blockchain system via verifiable credentials. The regulation module

implements an attribute-based smart contract invocation mechanism to ensure that cross-domain data transactions are always executed under supervision.

Thus, fund security and data security rely on a trusted cross-chain information transmission mechanism. The cross-domain information mentioned above can be regarded as the identity information of contracts SC_p and SC_q . The cross-domain identity mechanism proposed in Sections 3 and 4 ensures the credibility and security of cross-domain state information in cross-domain data transactions, thereby safeguarding both data security and fund security within the cross-domain data trading mechanism.

6.3 Cross-Domain Data Transmission Mechanism

After the data transaction is approved by both parties' data trading systems and the cross-domain payment is completed, the process proceeds to the cross-domain data transmission phase. This phase involves three participants: the data custodian, the edge nodes of the buyer and seller systems, and the buyer. During this phase, both the integrity and transmission efficiency of cross-domain data must be ensured.

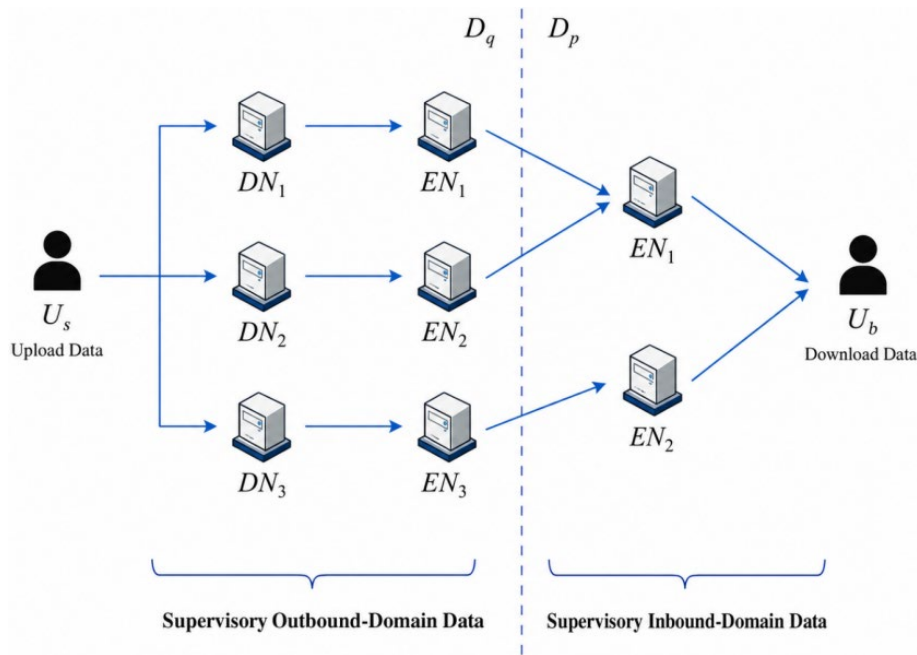


Figure 9. Cross-domain data-transmission process

As illustrated in Figure 9, the cross-domain data transmission process consists of three steps: 1) data transmission from the data custodian node to the edge node; 2) cross-domain transmission between edge nodes of the two domains; 3) data transmission from the edge node to the buyer. This workflow includes two intra-domain data transmissions and one cross-domain data transmission. Large data volumes will result in substantial overhead. Therefore, this paper draws on DHT-oriented blockchain storage optimization [45] and Ethereum-backed cloud data-provenance assurance [46], together with parallel transmission, to improve data-placement efficiency, traceability, and cross-domain delivery performance.

First, after the seller deposits data to data custodian nodes, the data is split into fixed-size data blocks for storage. Then, based on the resolution scope partitioning algorithm described in Section 5, the data custodian nodes store these blocks across participating custodian nodes according to each block's hash value. When data transmission is initiated, the buyer uses evidence from cross-domain transaction events

(including SellSuccess, BuySuccess, rootHashd and other information) to request all data blocks and reconstructs the complete data asset once transmission finishes.

During this transmission process, the edge nodes of both domains are responsible for auditing the content of cross-domain data blocks. The audit verifies whether each block belongs to the traded data asset and whether the block content complies with cross-domain access regulations.

At the cross-domain data transmission stage, payment has already been settled to the seller, and the buyer has obtained valid data transmission proof, so this mechanism eliminates core data and fund security risks. The original data is fragmented and redundantly stored across decentralized data custodian nodes. This decentralized storage scheme ensures that the buyer can still retrieve the complete data asset even if some malicious custodian nodes refuse service, while parallel transmission guarantees efficient data retrieval. After transaction confirmation, the buyer transfers the cross-domain payment to a cross-chain escrow account. If the seller

delivers valid data transmission proof to the buyer, the seller may directly withdraw the funds from the escrow account; otherwise, the funds will be automatically returned to the buyer after a predetermined lock-up period. This strategy protects the fund and data security of both the buyer and the seller.

The system uses an incentive mechanism to encourage more edge nodes and data custodian nodes to participate in data transmission and storage, enabling parallel transmission to accelerate data transfer efficiency. When the proportion of participating data custodian nodes and cross-domain edge nodes matches the proportion of the transmission bandwidth of their interconnections, channel congestion and idle bandwidth can be avoided, thus achieving optimal transmission efficiency. Given the variable number of participating nodes, splitting data into smaller blocks, applying dynamic link switching for transmission, and performing separate deduplication can also achieve near-optimal transmission efficiency.

7 Implementation and Evaluation

7.1 System Implementation

To implement the data trading system proposed in this paper, we set up the following experimental environment. The experimental environment consists of 20 servers, each equipped with a 2.4 GHz 8-core CPU, 8 GB RAM, a 50 GB SSD, and running the CentOS 7 operating system. To simulate cross-domain data trading scenarios, the servers are divided into two domains. Servers within each domain are connected via a high-speed local area network, while separate network segments and firewalls are configured between the two domains to ensure intra-domain security. Each server runs a Go-based Ethereum client, and the blockchain nodes within each domain form an independent blockchain via a consensus algorithm. Private-blockchain benchmarking research identifies transaction execution, consensus, and storage as principal throughput and latency bottlenecks [47].

```
function processTransaction(
    IdentityAttributes memory buyer,
    IdentityAttributes memory seller,
    IdentityAttributes memory product,
    string memory buyerMessage,
    bytes memory buyerSignature,
    string memory sellerMessage,
    bytes memory sellerSignature,
    string memory productMessage,
    bytes memory productSignature
) public returns (bool) {
    // Verify signatures of all participants
    require(
        verify(buyer.issuerAddress, buyerMessage, buyerSignature),
        "Buyer identity invalid"
    );
    require(
        verify(seller.issuerAddress, sellerMessage, sellerSignature),
        "Seller identity invalid"
    );
    require(
        verify(product.issuerAddress, productMessage, productSignature),
        "Product identity invalid"
    );

    // Validate identity attributes
    require(regulate1(buyer), "Buyer identity not approved");
    require(regulate2(seller), "Seller identity not approved");
    require(regulate3(product), "Product identity not approved");

    // Execute the transaction
    return transact(
        buyer.walletAddress,
        seller.walletAddress,
        product.value
    );
}
```

Figure 10. Partial implementation of the cross-domain data-trading smart contract

As illustrated in Figure 10, the on chain DID identifier registration and resolution smart contracts, as well as the buyer and seller smart contracts, are developed using Solidity [48]. Intra-domain users and data custodian nodes interact

with blockchain nodes through a Python Web3-compatible client to perform operations including DID registration and resolution, data purchase and payment, and data publication on the blockchain. Similar private-Ethereum studies use such

client interfaces to measure transaction latency and throughput [49].

7.2 System Analysis

This section analyzes the system against the design objectives, to verify whether the cross-domain identity resolution mechanism and the data trading system meet the required design goals.

1) Consistency of Cross-Domain Identity Resolution Results. The proposed cross-domain identity resolution protocol verifies the correctness of resolution results using the external block root hashes recorded on the local blockchain and checks the corresponding block numbers to confirm whether the results are the latest version. If cached results are outdated or not the latest, cross-verification is adopted to retrieve correct and up-to-date resolution results. Furthermore, to resolve the potential staleness issue caused by caching, a passive cache update mechanism is introduced to guarantee consistency of resolution results without compromising resolution efficiency.

2) Availability of Cross-Domain Identity Resolution Service. Assume that in block t , there are n participating nodes in the cross-domain identity resolution service, and the total reward allocated in this block is s . The average reward obtained by each participating node in this block is s/n . When the number of participating nodes is insufficient to handle current cross-domain resolution requests, the total reward s on the blockchain can be increased, raising the average reward per node. This incentivizes more edge nodes to join and enhances the overall capacity of the cross-domain identity resolution service. If malicious nodes exist, the detection and penalty mechanisms generate fraud evidence against malicious behaviors, enabling the penalty or exclusion of malicious nodes to preserve service availability. Rewards for users who identify malicious nodes encourage edge nodes to actively verify resolution results from other nodes, so that nodes disrupting service availability can be detected as quickly as possible.

3) Atomicity of Cross-Domain Data Transactions. The data purchase process of cross-domain data transactions

consists of two parts: smart contract execution and cross-domain transmission of transaction states. Smart contract execution is atomic, and the state information of SC_p and SC_q is transmitted across domains via the cross-domain identity resolution mechanism. This mechanism ensures consistency and availability, so that state information can be transmitted correctly and in a timely manner even with malicious nodes in the two domains, preventing abnormal termination of cross-domain data transactions. In addition, a rollback mechanism is implemented for cross-domain data transactions. If a transaction fails due to identity verification issues, regulatory compliance issues, or other exceptions, edge nodes can actively initiate a rollback transaction to revert intermediate state changes and refund the payment. For data delivery, the decentralized nature of data custodianship ensures normal cross-domain data transmission even if individual malicious data custodian nodes exist.

4) Fine-Grained Regulation of Cross-Domain Data Transactions. As shown in the example smart contract in Figure 10, the cross-domain data trading contract SC_p accepts parameters including the buyer's and seller's DID identifiers, DID documents, verifiable credentials, as well as the DID identifier and DID document of the traded data asset. SC_p uses the identity verification module to verify the correspondence between DID identifiers and DID documents and validates the integrity of verifiable credentials by checking their signatures. After successful verification, regulation can be conducted at the identifier level, while the content of DID documents and verifiable credentials enables attribute-based fine-grained regulation.

7.3 Performance Evaluation

The purpose of the system performance evaluation is to verify the feasibility of conducting cross-domain data transactions using the proposed system. Therefore, this section first tests each layer separately according to the system's layered architecture, then runs a complete cross-domain data transaction workflow, analyzes system performance bottlenecks, and discusses feasible optimization strategies.

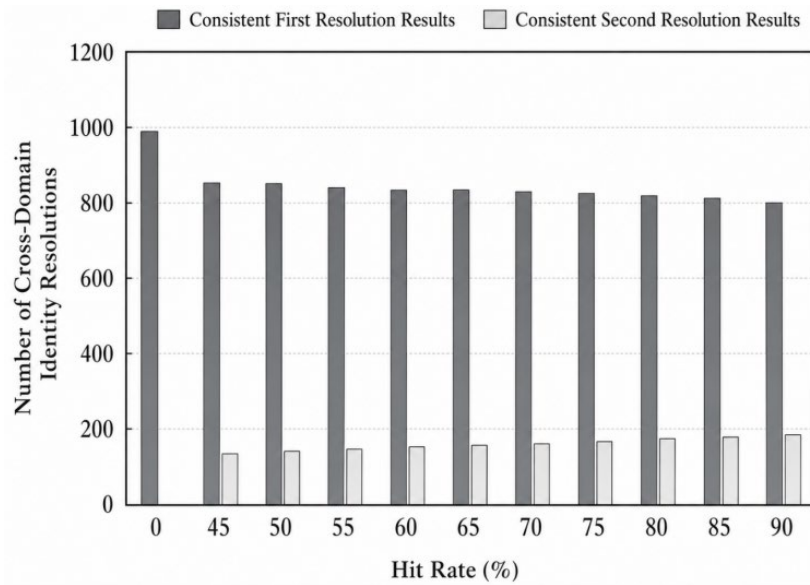


Figure 11. Cross-domain identity-resolution consistency

To verify the consistency of cross-domain identity resolution, multiple cross-domain identity resolution requests are initiated simultaneously across the two domains to check whether the resolution results remain consistent. Different cache hit ratios have distinct impacts on consistency; an excessively high cache hit ratio may lead to inconsistent cross-domain identity resolution results. Thus, we conduct consistency comparisons under different cache hit ratios. As shown in Figure 11, cache-free cross-domain identity resolution can fully guarantee result consistency. When the cache hit ratio reaches 50%, 14% of initial resolution results become inconsistent and require secondary resolution to restore consistency, which improves the effective cache hit ratio by 72%. As the overall cache hit ratio continues to rise, thanks to the passive cache update mechanism, the proportion of inconsistent initial resolution results only rises slightly to 19%, and full consistency is restored via the secondary resolution mechanism. These results confirm that the proposed scheme effectively maintains system consistency. Identity resolution is normally executed once in off-chain applications or communications. Even when secondary

resolution is required, the resulting latency accounts for only a small fraction of other communication delays (such as page rendering and data transmission), so it has no obvious impact on practical application performance.

On the identity layer, we implement an Ethereum-compatible lightweight DID registry following the blockchain-based identity-management architecture surveyed in [50]. The registry records 100,000 DID identifiers and their corresponding attribute information on the blockchain, after which we measure the average latency of cross-domain identity resolution. As shown in Figure 12, each cross-domain identity resolution involves 2 intra-domain communications (between users and edge nodes) and 2 cross-domain communications (between edge nodes across domains), resulting in a relatively high average latency of 180 ms. Given the locality property of identity resolution, caching can be applied to accelerate the resolution process. After adopting the LRU caching mechanism to raise the cross-domain resolution cache hit ratio to 70%, the average cross-domain identity resolution latency is effectively reduced to 82 ms.

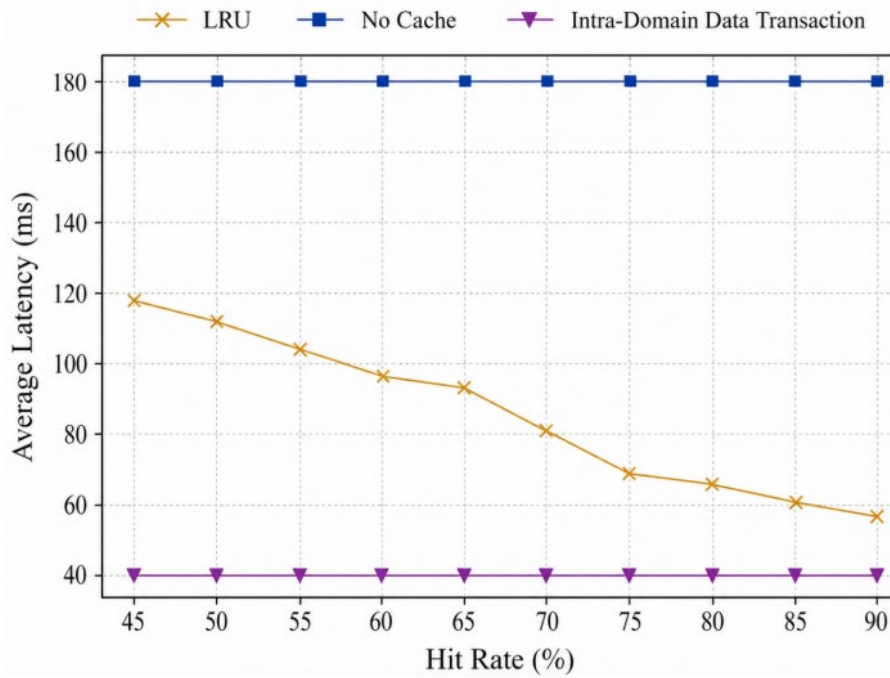


Figure 12. Cross-domain identity-resolution latency

On the transaction layer, the buyer and seller smart contracts are implemented in Solidity, and the Web3.py library is used in the user client to enable interactions with both external and local blockchains. A single successful cross-domain data transaction requires at least 2 EVM smart contract executions and 4 interactions between the user client and the blockchain (the buyer invoking the buyer contract, edge nodes invoking the seller contract, edge nodes relaying transaction success events to each other, and edge nodes relaying transaction success events back to the user). The primary performance bottleneck on the transaction layer is smart contract execution. Since blockchain transactions that invoke smart contracts must be packaged into blocks before execution, the execution latency is non-deterministic. When the blockchain is under heavy transaction load, transactions may need to wait for multiple blocks to be confirmed. The blockchain has a block time of 1 second, with an average of 200 transactions per block. The time taken for a transaction to be included in

a block is closely related to its gas fee. As shown in Figure 13, when data transactions are submitted at a rate of 500 transactions per second, blockchain network congestion occurs, resulting in an average wait of 2.3 blocks per transaction. Two sequential smart contract executions take approximately 5 seconds. Under non-congested conditions, cross-domain transactions have roughly twice the latency of intra-domain transactions. Even under congestion, intra-domain transactions generally require at least one block confirmation, and the cross-domain transaction latency remains acceptable for user experience. As the transaction submission rate increases, end-to-end transaction latency further rises. The latency from client-blockchain interactions is negligible. Therefore, improving overall transaction layer efficiency requires increasing blockchain throughput. End users can also reduce data transaction latency by setting higher gas fees to accelerate transaction confirmation.

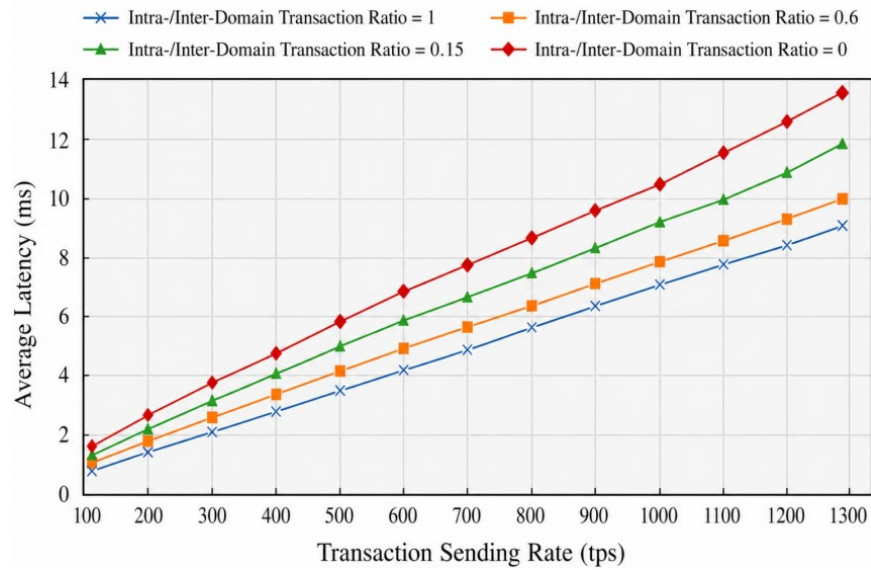


Figure 13. Cross-domain data-trading contract execution latency

In practical application scenarios, the overall system efficiency is reflected by the throughput of data transaction operations. In this set of experiments, we construct a large number of intra-domain and cross-domain data transactions in batches and run transactions with varying ratios of cross-domain trades. The experimental setup contains two domains, each with 8 blockchain nodes, 4 of which also serve as edge nodes. The base blockchain throughput reaches approximately 200 tps with a block gas limit of 10,000,000. However, data transactions account for only a subset of total blockchain transactions, and the effective data transaction throughput decreases as the proportion of cross-domain transactions increases. As shown in Figure 14, when all data transactions are cross-domain, the throughput stabilizes at roughly 90 tps. Although each cross-domain data transaction only requires one on-chain transaction within each domain, processing cross-domain requests originating from the other

domain consumes half of the block space in the local domain. Additionally, cross-domain data transactions require extra regulatory compliance checks and emit additional cross-domain transaction events, which increases the gas cost per transaction and further reduces the number of transactions that can fit into a single block. To improve system throughput, two key directions are increasing the block gas limit (GasLimit) and reducing block confirmation time. For example, Layer-2 protocols [51] can reduce on-chain transaction costs so that more transactions can be processed per unit time; Directed Acyclic Graph (DAG) technology enables parallel block processing to shorten the average block interval. Other advanced blockchain technologies including sharding and parallel execution can also be adopted to further boost the cross-domain data transaction throughput of the proposed system.

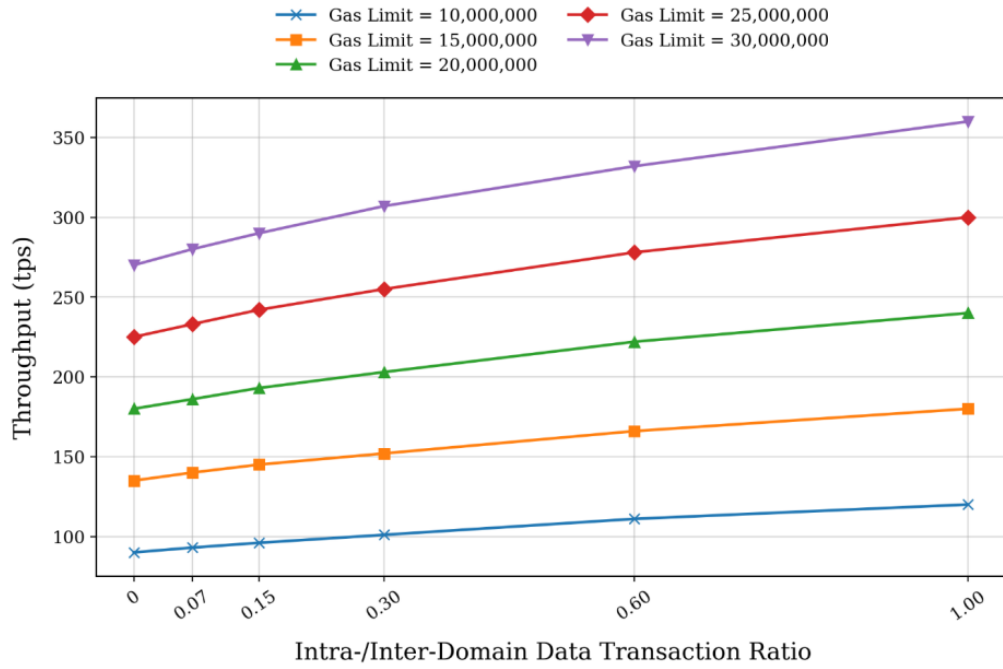


Figure 14. Cross-domain data-trading throughput

7.4. Supplementary Robustness, Multi-Domain Scalability, and Comparative Evaluation

The two-domain prototype measurements above are complemented with reproducible protocol-level simulations. These simulations evaluate the probability structure of the security mechanism and the scaling behavior of consistent hashing; they are reported separately from end-to-end prototype throughput so that simulated protocol behavior is not confused with measured system performance.

7.4.1 Quantitative robustness against collusive malicious nodes

For each malicious-node probability p in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, we simulated 1,000,000 resolution attempts with eight root-hash voting edge nodes and two independently selected cross-verification responders (random seed 20260831). A false acceptance is conservatively counted only when a strict majority of the eight voters is malicious, and both verification responders are malicious. Thus, the analytical upper bound is

$$P_{FA}(p) \leq p^2 \sum_{l=8}^8 \binom{8}{l} p^l (1-p)^{8-l}. \text{ This is a conservative}$$

collusion model: when the realized malicious fraction is strictly below one half, the majority-voting assumption itself gives a correct root and forged, or stale data are rejected by the Merkle-proof/latest-block checks.

Table 2. Protocol-level collusive false-acceptance analysis

p (%)	Both verifiers malicious p^2 (%)	Majority-root compromise (%)	Analytical P_{FA} upper bound (%)	Monte Carlo false acceptance (%)
10	1.00	0.0432	0.00043	0.0007
20	4.00	1.0406	0.04163	0.0432
30	9.00	5.7968	0.52171	0.5153
40	16.00	17.3670	2.77873	2.7714
50	25.00	36.3281	9.08203	9.2032

The Monte Carlo estimates closely follow the analytical upper bound. At $p = 0.30$, the simulated joint false acceptance

rate is 0.5153%, and the analytical bound is 0.52171%. The rapid increase near the 50% boundary confirms that

preserving an honest voting majority is the key deployment assumption; below that boundary, dual-node cross-verification reduces the probability that a colluding pair can even present matching malicious responses.

7.4.2 Multi-domain scalability of consistent-hash scope allocation

To extend the evaluation beyond two blockchain domains without claiming unmeasured full-stack network latency, we

performed a consistent-hash simulation with 2, 4, 8, and 16 domains, four edge nodes per domain, and uniformly hashed DID identifiers. For each setting, 200 independent runs were executed with 10,000 DID lookups per run. We then added one new four-node domain and measured the fraction of DID scopes remapped. Binary-search lookup on the sorted hash ring requires $O(\log n)$ comparisons, while only the intervals captured by the new nodes are reassigned.

Table 3. Multi-domain consistent-hash scalability simulation

Domains	Edge nodes	Binary-search depth ($\text{ceil}(\log_2 n)$)	Remapped DIDs after adding one 4-node domain (%)	95% CI (percentage points)
2	8	3	33.57	+/- 1.76
4	16	4	20.35	+/- 1.32
8	32	5	10.96	+/- 0.69
16	64	6	5.97	+/- 0.36

The simulated remapping ratios are close to the consistent-hashing expectation for adding four nodes, namely $4/(n+4)$. As the deployment grows from 2 to 16 domains, the lookup depth increases only from 3 to 6 comparisons, whereas the average scope remapping after adding a new domain decreases from 33.57% to 5.97%. These results support the use of consistent hashing for incremental multi-domain expansion while keeping redistribution localized.

7.4.3 Controlled baseline and cross-work comparison

A direct tps ranking against published systems would be misleading because Agora [27], FAST DATA [30], and DataTBC [34] use different consensus stacks, hardware, data paths, and workload definitions, and they do not expose a compatible cross-domain DID resolver that can be executed in the present prototype. We therefore combine the architecture-level comparison in Table 1 with controlled baselines on the same implementation. This isolates the overhead of the proposed cross-domain mechanisms without converting heterogeneous literature measurements into artificial equivalents.

Table 4. Controlled baselines on the same prototype

Configuration	Metric	Result	Interpretation
Intra-domain DID resolution	Average latency	about 40 ms	Local-resolution lower-bound baseline in Figure 12.
Cross-domain DID resolution, no cache	Average latency	180 ms	Cross-domain baseline with four communications.
Proposed LRU cache, 70% hit rate	Average latency	82 ms	54.4% lower than the no-cache cross-domain baseline.
Base blockchain, GasLimit = 10,000,000	Throughput	about 200 tps	Raw chain-capacity baseline.
100% cross-domain data trading	Throughput	about 90 tps	Includes dual-domain processing, regulation checks, and cross-domain events.

Together, Tables 1-4 show that the main contribution is not a claim of universally higher raw blockchain throughput, but the integration of cross-domain identity consistency, availability incentives, transaction atomicity, and attribute-level regulation with quantified and acceptable overhead on a common prototype.

8. Conclusions

This study developed and evaluated a decentralized identity-based cross-domain data trading system that integrates identity resolution, incentive-compatible service organization, smart-contract-based regulation, atomic transaction execution, and decentralized data delivery. Unlike conventional blockchain marketplaces that require buyers, sellers, and data assets to be registered within one platform, the proposed system establishes a unified cross-domain identity layer through DID documents, verifiable credentials,

external blockchain root hashes, Merkle proofs, and signed responses from edge nodes. The resolution protocol, active and passive cache-update strategies, redundant range assignment, and cross-verification mechanism jointly reduce the risks of stale identity records, malicious responses, unavailable resolvers, and unaccountable service behavior. Consistent hashing distributes resolution workloads among edge nodes, while scope-proportional rewards and fraud penalties encourage sustained and correct participation. At the transaction layer, the Verify, Regulate, Transact, Approval, and TransferFrom modules allow both domains to validate external identities, enforce local attribute-based rules, confirm payment states, and coordinate ownership transfer. Decentralized custody, block fragmentation, redundant storage, escrow, and parallel transmission further protect data availability and the interests of both trading parties.

The prototype evaluation confirms the feasibility of the design. The system was implemented on 20 servers organized into two independent Ethereum domains. Tests involving 100,000 ERC-1056 identities showed that cross-domain resolution without caching produced fully consistent results. At a 50% cache hit rate, 14% of initial results required a second resolution, but the secondary verification mechanism restored 100% consistency and raised the effective cache hit rate by 72%. As the cache hit rate increased, initial inconsistency remained limited to 19%. The LRU cache reduced average resolution latency from 180 ms to 82 ms at a 70% hit rate, corresponding to a 54.4% reduction. Transaction experiments showed that, at 500 submitted transactions per second, transactions waited an average of 2.3 blocks, and two sequential contract executions took about 5 s. With a 10,000,000 gas limit, the blockchain achieved approximately 200 tps, and the throughput under entirely cross-domain transactions remained about 90 tps. Increasing the gas limit to 30,000,000 raised measured throughput to approximately 270-360 tps. In the supplementary protocol-level analysis, the collusive false-acceptance upper bound is 0.52171% at a 30% independent malicious-node probability, closely matching the 0.5153% Monte Carlo estimate, while consistent-hash simulation over 2-16 domains shows that scope remapping after adding one four-node domain decreases from 33.57% to 5.97% and lookup depth grows only logarithmically. Overall, the results demonstrate that secure identity consistency, fine-grained compliance, transaction atomicity, and acceptable performance can be achieved simultaneously. The architecture also remains modular, allowing regulatory policies and payment mechanisms to be adapted independently across jurisdictions and application sectors.

Future work will evaluate larger full-stack multi-domain deployments and quantify concrete gas and latency trade-offs of four directions. First, Layer-2 rollups can batch Verify/Regulate/Transact calls and move proof generation off-chain so that gas savings per completed cross-domain trade can be measured directly. Second, sharding can partition DID registries and transaction state by domain or DID prefix, with cross-shard receipts carrying root commitments; future experiments will measure cross-shard confirmation delay and

state-migration overhead. Third, DAG-based ordering and parallel execution can schedule independent DID resolutions and non-conflicting transactions concurrently to reduce confirmation latency under burst traffic. Fourth, zero-knowledge proofs can allow users to prove regulatory predicates over verifiable credentials without disclosing raw attributes, and the resulting proof-generation, verification-gas, and privacy-performance trade-offs will be evaluated under realistic compliance workloads. These directions extend the current architecture without changing its separation of identity, transaction, and data layers.

References

- [1] Fernandez R C, Subramaniam P, Franklin M J. Data market platforms: Trading data assets to solve data problems[J]. *Proceedings of the VLDB Endowment*, 2020, 13(11): 1933-1947. DOI:10.14778/3407790.3407800
- [2] Chen L, Koutris P, Kumar A. Towards model-based pricing for machine learning in a data marketplace[C]//*Proceedings of the 2019 International Conference on Management of Data*. New York: ACM, 2019: 1535-1552
- [3] Wang Y J, Wu Q H, Qin B, et al. Identity-based data outsourcing with comprehensive auditing in clouds[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(4): 940-952. DOI:10.1109/TIFS.2016.2646913
- [4] Pei J. A survey on data pricing: From economics to data science[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2022, 34(10): 4586-4608. DOI:10.1109/TKDE.2020.3045927
- [5] Zhao Y Q, Yu Y, Li Y N, et al. Machine learning based privacy-preserving fair data trading in big data market[J]. *Information Sciences*, 2019, 478: 449-460
- [6] Bajoudah S, Dong C, Missier P. Toward a decentralized, trust-less marketplace for brokered IoT data trading using blockchain[C]//*Proceedings of 2019 IEEE International Conference on Blockchain (Blockchain)*. Piscataway, NJ: IEEE, 2019: 339-346
- [7] Shen M, Liu H S, Zhu L H, et al. Blockchain-assisted secure device authentication for cross-domain industrial IoT[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(5): 942-954
- [8] Su L Y, Shen X Y, Du X Y, et al. Evil under the sun: Understanding and discovering attacks on Ethereum decentralized applications[C]//*Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)*. Berkeley, CA: USENIX Association, 2021: 1307-1324
- [9] Lu Q H, Xu X W, Liu Y, et al. uBaaS: A unified blockchain as a service platform[J]. *Future Generation Computer Systems*, 2019, 101: 564-575
- [10] Han R, Yan Z, Liang X Q, Yang L T. How can incentive mechanisms and blockchain benefit with each other? A survey[J]. *ACM Computing Surveys*, 2023, 55(7): Article 136, 1-38. DOI:10.1145/3539604
- [11] Islam M M, Islam M K, Shahjalal M, et al. A low-cost cross-border payment system based on auditable cryptocurrency with consortium blockchain: Joint digital currency[J]. *IEEE Transactions on Services Computing*, 2023, 16(3): 1616-1629. DOI:10.1109/TSC.2022.3207224
- [12] Lv M. A cross-domain data flow security governance and privacy protection mechanism for distributed digital art platforms[J]. *EAI Endorsed Transactions on Scalable Information Systems*, 2026, 12(11). DOI:10.4108/eetsis.13268

- [13] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]//Proceedings of the 2000 IEEE Symposium on Security and Privacy. Los Alamitos, CA: IEEE Computer Society, 2000: 44-55. DOI:10.1109/SECPR1.2000.848445
- [14] Cash D, Jaeger J, Jarecki S, et al. Dynamic searchable encryption in very-large databases: Data structures and implementation[C]//Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS). Reston, VA: Internet Society, 2014. DOI:10.14722/NDSS.2014.23264
- [15] Wang C, Chow S S M, Wang Q, et al. Privacy-preserving public auditing for secure cloud storage[J]. IEEE Transactions on Computers, 2013, 62(2): 362-375. DOI:10.1109/TC.2011.245
- [16] Yang X, Hu J. Agricultural supply chain security management based on distributed blockchain and time series analysis: Focus on cross-domain data security and privacy computing[J]. EAI Endorsed Transactions on Scalable Information Systems, 2026, 12(11). DOI:10.4108/eetsis.12166
- [17] Feng C S, Liu B, Guo Z, et al. Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of drones[J]. IEEE Internet of Things Journal, 2022, 9(8): 6224-6238. DOI:10.1109/JIOT.2021.3113321
- [18] Chen D J, Zhang N, Cheng N, et al. Physical layer based message authentication with secure channel codes[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 17(5): 1079-1093. DOI:10.1109/TDSC.2018.2846258
- [19] Chen D J, Jiang S Q, Zhang N, et al. On message authentication channel capacity over a wiretap channel[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 3107-3122
- [20] Li M, Weng J, Yang A J. CrowdBC: A blockchain-based decentralized framework for crowdsourcing[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(6): 1251-1266. DOI:10.1109/TPDS.2018.2881735
- [21] Qin Z G, Xiong H, Wu S K, Batamuliza J. A survey of proxy re-encryption for secure data sharing in cloud computing[J]. IEEE Transactions on Services Computing, 2016. DOI:10.1109/TSC.2016.2551238
- [22] Lai J Z, Deng R H, Guan C W, et al. Attribute-based encryption with verifiable outsourced decryption[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(8): 1343-1354
- [23] Zhang L, Xiong H, Huang Q, Li J, Choo K K R, Li J. Cryptographic solutions for cloud storage: Challenges and research opportunities[J]. IEEE Transactions on Services Computing, 2022, 15(1): 567-587. DOI:10.1109/TSC.2019.2937764
- [24] Wan Z G, Liu W, Cui H. HIBChain: A hierarchical identity-based blockchain system for large-scale IoT[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(2): 1286-1301. DOI:10.1109/TDSC.2022.3152797
- [25] Yang X H, Li W J. A zero-knowledge-proof-based digital identity management scheme in blockchain[J]. Computers & Security, 2020, 99: 102050
- [26] Xu J, Xue K P, Tian H Y, et al. An identity management and authentication scheme based on redactable blockchain for mobile networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6): 6688-6698
- [27] Koutsos V, Papadopoulos D, Chatzopoulos D, et al. Agora: A Privacy-Aware Data Marketplace[J/OL]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(6): 3728-3740
- [28] Li C L, Liang S Y, Zhang J, et al. Blockchain-based data trading in edge-cloud computing environment[J/OL]. Information Processing & Management, 2022, 59(1): 102786
- [29] Gupta P, Dedeoglu V, Kanhere S S, et al. TrailChain: Traceability of data ownership across blockchain-enabled multiple marketplaces[J]. Journal of Network and Computer Applications, 2022, 203: 103389
- [30] Dixit A, Singh A, Rahulamathavan Y, et al. FAST DATA: A fair, secure, and trusted decentralized IIoT data marketplace enabled by blockchain[J]. IEEE Internet of Things Journal, 2023, 10(4): 2934-2944
- [31] Lin C H V, Huang C C J, Yuan Y H, et al. A fully decentralized infrastructure for subscription-based IoT data trading[C/OL]//Proceedings of 2020 IEEE International Conference on Blockchain (Blockchain). Piscataway, NJ: IEEE, 2020: 162-169
- [32] Ou W, Huang S, Zheng J, et al. An overview on cross-chain: Mechanism, platforms, challenges and advances[J]. Computer Networks, 2022, 218: 109378
- [33] Gilad-Bachrach R, Laine K, Lauter K, Rindal P, Rosulek M. Secure Data Exchange: A marketplace in the cloud[C]//Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop. New York: ACM, 2019: 117-128. DOI:10.1145/3338466.3358924
- [34] Hu D H, Li Y F, Pan L X, et al. A blockchain-based trading system for big data[J]. Computer Networks, 2021, 191: 107994
- [35] Muehle A, Gruener A, Gayvoronskaya T, Meinel C. A survey on essential components of a self-sovereign identity[J]. Computer Science Review, 2018, 30: 80-86. DOI:10.1016/j.cosrev.2018.10.002
- [36] Ferdous M S, Chowdhury F, Alassafi M O. In search of self-sovereign identity leveraging blockchain technology[J]. IEEE Access, 2019, 7: 103059-103079. DOI:10.1109/ACCESS.2019.2931173
- [37] Soltani R, Nguyen U T, An A. A survey of self-sovereign identity ecosystem[J]. Security and Communication Networks, 2021: 8873429. DOI:10.1155/2021/8873429
- [38] Butincu C N, Alexandrescu A. Design aspects of decentralized identifiers and self-sovereign identity systems[J]. IEEE Access, 2024, 12: 60928-60942. DOI:10.1109/ACCESS.2024.3394537
- [39] Madhu G C, Peroumal V. Blockchain-assisted authentication and energy-efficient clustering framework for secure IoT communication[J]. EAI Endorsed Transactions on Internet of Things, 2025, 11(1). DOI:10.4108/eetiot.9520
- [40] Ren Y Z, Xie R C, Yu F R, Huang T, Liu Y J. Potential identity resolution systems for the industrial Internet of Things: A survey[J]. IEEE Communications Surveys & Tutorials, 2021, 23(1): 391-430. DOI:10.1109/COMST.2020.3045136
- [41] Cohen E, Halperin E, Kaplan H. Performance aspects of distributed caches using TTL-based consistency[J]. Theoretical Computer Science, 2005, 331(1): 73-96. DOI:10.1016/j.tcs.2004.09.033
- [42] Li X, Jiang P, Chen T, et al. A survey on the security of blockchain systems[J]. Future Generation Computer Systems, 2020, 107: 841-853. DOI:10.1016/j.future.2017.08.020
- [43] Li H. A cloud environment security access control scheme based on federated learning and fuzzy logic integration[J]. EAI Endorsed Transactions on Scalable Information Systems, 2026, 12(9). DOI:10.4108/eetsis.11731
- [44] Chen T, Zhang Y, Li Z, et al. TokenScope: Automatically detecting inconsistent behaviors of cryptocurrency tokens in Ethereum[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2019: 1503-1520. DOI:10.1145/3319535.3345664
- [45] Xie Z, Zheng L. Optimization algorithm for blockchain data storage and retrieval based on DHT[J]. EAI Endorsed Transactions on Scalable Information Systems, 2026, 12(7). DOI:10.4108/eetsis.8189

- [46] Narayan G, Pavitra Haveri, Rashmi B, Yashwardhan Deewan. Framework for data provenance assurance in cloud environment using Ethereum blockchain[J]. EAI Endorsed Transactions on Scalable Information Systems, 2023, 11(2). DOI:10.4108/eetsis.3536
- [47] Dinh T T A, Wang J, Chen G, et al. BLOCKBENCH: A framework for analyzing private blockchains[C]//Proceedings of the 2017 ACM International Conference on Management of Data. New York: ACM, 2017: 1085-1100. DOI:10.1145/3035918.3064033
- [48] Dannen C. Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners[M]. Berkeley, CA: Apress, 2017. DOI:10.1007/978-1-4842-2535-6
- [49] Rouhani S, Deters R. Performance analysis of Ethereum transactions in private blockchain[C]//Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science. Piscataway, NJ: IEEE, 2017: 70-74. DOI:10.1109/ICSESS.2017.8342866
- [50] Ahmed M R, Islam A K M M, Shatabda S, Islam S. Blockchain-based identity management system and self-sovereign identity ecosystem: A comprehensive survey[J]. IEEE Access, 2022, 10: 113436-113481. DOI:10.1109/ACCESS.2022.3216643
- [51] Gudgeon L, Moreno-Sanchez P, Roos S, et al. SoK: Layer-two blockchain protocols[C]//Financial Cryptography and Data Security. Cham: Springer, 2020: 201-226. DOI:10.1007/978-3-030-51280-4_12