Towards Secure Cross-Organizational Data Transfer: A Blockchain-Enabled Message Broker Approach

Gleb Slepenkov[1] [0009-0001-9978-396X]

Saint Petersburg State University, 7-9 Universitetskaya Embankment, St Petersburg, Russia, 199034

Abstract. Secure data transmission (SDT) is crucial in modern information societies, particularly for cross-organizational data exchange. This paper investigates the application of blockchain technology to create robust SDT systems, combining the benefits of using blockchain as both a data access interface and a direct data transfer tool. We propose a novel architecture featuring a message broker-like design that overcomes data size limitations present in existing blockchain-based SDT solutions. The architecture incorporates blockchain, internal data storage within organizations, and connectors facilitating interaction between these components. The SDT process relies on smart contracts for managing data access, transfer, and confirmation, while the actual data is stored off-chain. We advocate for a private, two-layered blockchain implementation, employing PBFT consensus within smaller clusters for efficient message transfer and RAFT consensus for system management and data replication. This layered approach enhances both fault tolerance and scalability, paving the way for a practical and secure cross-organizational SDT system.

Keywords: Blockchain · Secure Data Transfer (SDT) · Data Security · Distributed Ledger Technology (DLT) · Message Broker · Data Access Control · Cross-Organizational Data Sharing · Scalability · Fault Tolerance

1 Introduction

In contemporary information societies, secure data transmission (SDT) is essential for maintaining data confidentiality and integrity. SDT systems are critical infrastructure components across diverse sectors, including financial institutions, governmental agencies, and enterprises.

Implementing robust SDT mechanisms becomes particularly challenging when facilitating cross-organizational data exchange or connecting geographically dispersed departments within a single organization. Such scenarios necessitate data transfer across external, public networks, thereby significantly elevating the risk of unauthorized data disclosure or modification during transit. Consequently, rigorous data integrity verification and sender authentication protocols are indispensable. Furthermore, confirmation of data reception is often a critical requirement. The heterogeneity of data exchange technologies employed by different entities can further complicate the SDT process.

Blockchain represents an emerging technology for SDT solutions. This technology by design provides core features essential for SDT, such as fault tolerance, ledger data immutability and zero trust between blockchain nodes. Moreover, many blockchains are designed to operate in unreliable public networks with nodes running in different heterogeneous clusters owned by separate stakeholders. For this reason, blockchains are especially suitable for cross-organizational data transfer. Furthermore, blockchain platforms often provide smart contracts, which allow implementing custom data transfer logic directly on the blockchain.

This suitability has stimulated significant research interest, as evidenced by comprehensive surveys exploring blockchain applications for data sharing and exchange [16] and for specific domains like smart transport [2]. These surveys highlight the potential of blockchain to address the challenges of SDT in diverse and demanding environments.

This study investigates existing paradigms of blockchain applications to the SDT process and proposes a new approach, which allows to combine the advantages of existing paradigms and create a SDT blockchain-based system that preserves key features of existing approaches but has a message broker-like architecture and relaxed data size limitations. The study is organized in the following way. Section 2 describes existing data transfer paradigms in detail. Section 3 represents the proposed system architecture. Section 4 describes implementation details of the proposed system. Finally, Section 5 concludes this article and denotes further research directions.

2 Related Work

The integration of blockchain technology into data transfer systems presents two primary paradigms: blockchain as a data access interface and blockchain as a direct data transfer tool. This section examines the existing literature, classifying studies according to these two distinct roles of blockchain in facilitating data exchange.

2.1 Blockchain as a Data Access Interface

This approach leverages blockchain as a secure and transparent mechanism for managing and controlling access to data that is typically stored off-chain. The blockchain records metadata about the data, access permissions, and transaction histories, thereby ensuring data integrity, auditability, and secure access control. The data itself is transferred using conventional methods.

Wang et al. introduced BBS, a big data sharing system utilizing blockchain for access control and data integrity management [17]. Similarly, Wang et al. proposed a blockchain-based model for sharing big data in the oil and gas sector, employing blockchain to enable secure data access and provenance tracking [18]. Yang et al. developed a sharing platform for wild bird data based on blockchain and IPFS, where blockchain governs access rights to data residing in IPFS [20]. A cross-organizational data sharing framework that uses blockchain probes to orchestrate and audit data access across various entities was presented by Jia et al. [9]. Gupta et al. explored the application of blockchain for securing data access within e-healthcare applications [8].

2.2 Blockchain as a Data Transfer Tool

This approach employs the blockchain directly to transfer data between participants. While data may occasionally be stored on-chain, it is more common for the blockchain to facilitate the transfer of data stored off-chain, often in conjunction with technologies such as peer-to-peer networks or distributed file systems.

Lin et al. proposed a multi-level blockchain architecture for secure data transfer in the Internet of Vehicles (IoV), directly leveraging blockchain for data exchange [12]. Peng et al. demonstrated the feasibility of building a peer-to-peer file storage and sharing system on a consortium blockchain, where the blockchain assists in discovering and securely transferring file chunks [14]. Priyadarshini et al. introduced a system for secured data transfer between fog nodes utilizing blockchain to enhance the security of the data transfer process itself [15].

2.3 Enabling Technologies and Security Considerations

Regardless of the chosen paradigm, several enabling technologies and security considerations are universally relevant. Kim et al. presented a hybrid decentralized PBFT blockchain framework for OpenStack message queues, aimed at improving fault tolerance and scalability, crucial aspects for reliable data transfer systems [11]. A publish-subscribe architecture to foster interoperability among different blockchain networks, thus facilitating data exchange across heterogeneous systems, was introduced by Ghaemi et al. [7]. Bagga et al. and Xu et al. explored blockchain-based authentication and key agreement protocols for IoV, enhancing the security of data transfer in vehicular contexts [3, 19]. Bogdanov et al. focused on the consensus mechanism, exploring a combination of PBFT and Raft [5], while Ai et al. proposed a Proof-of-Transactions consensus protocol [1], both striving for efficient and scalable agreement in distributed ledgers.

2.4 Comparison of Approaches

The two approaches, blockchain as a data access interface and blockchain as a data transfer tool, offer distinct advantages and disadvantages depending on the specific use case and requirements.

As illustrated in Table 1, the choice between these two approaches hinges on factors such as data size, scalability requirements, security priorities, and

Table 1. Comparison of Blockchain Approaches for Data Transfer

Feature	Blockchain as Data Access Interface	Blockchain as Data Transfer Tool
Data Storage	Primarily Off-Chain	Often Off-Chain, but metadata on-chain
Data Transfer	Traditional methods	Blockchain or Blockchain-assisted
Scalability	Higher, as data transfer is off-chain	Potentially Lower, dependent on blockchain throughput
Complexity	Lower	Higher
On-Chain Footprint	Smaller (Metadata only)	Larger (Metadata and potentially data chunks)
Suitability	Large datasets, access control focus	Smaller datasets, secure and direct transfer focus

the level of on-chain transparency desired. The data access interface approach is generally more suitable for scenarios involving large datasets and a primary focus on secure access control, while the data transfer tool approach is better suited for scenarios demanding secure and direct data transfer, even if it potentially introduces limitations in terms of scalability.

3 System Architecture Overview

3.1 System Components

This section describes the architecture of the proposed system. The main goal of such an architecture is to create a combination of the two main blockchain integration paradigms into data transfer systems described in Section 2 in order to create a system with a message broker-like architecture, convenient for data transfer, and relaxed data size limitations compared to existing solutions. Essentially, the architecture follows the trends described in related articles:

1. Blockchain usage as a message broker: [7], [11]. The key drawback of these solutions lies in the usage of the ledger for message transfer. It greatly limits the system scalability and transferred data size.
2. Blockchain as an interface to off-chain data (e.g., [9], [17]). Such solutions usually do not have any notification mechanisms. Hence, message receiving becomes a challenging task.

The proposed architecture assumes that the system should contain the following components:

– Blockchain as the main data transfer tool.
– Data storages internally used by counterparties (organizations) to store the data to be sent.
– Connectors — some application used by SDT process counterparties in order to create a connection between their data storages and blockchain nodes owned by the organization (mainly inspired by blockchain probes described in [9]).

A sample of such an architecture for SDT between two organizations is presented in Figure 1.

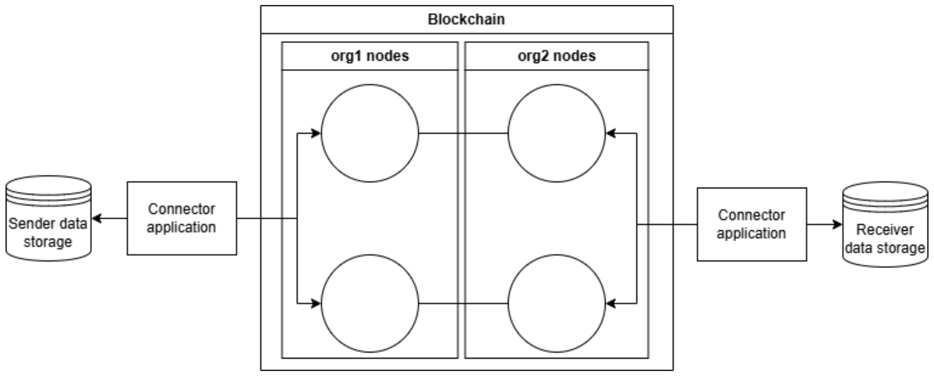


Fig. 1. System architecture for two organizations

3.2 SDT Process Scheme

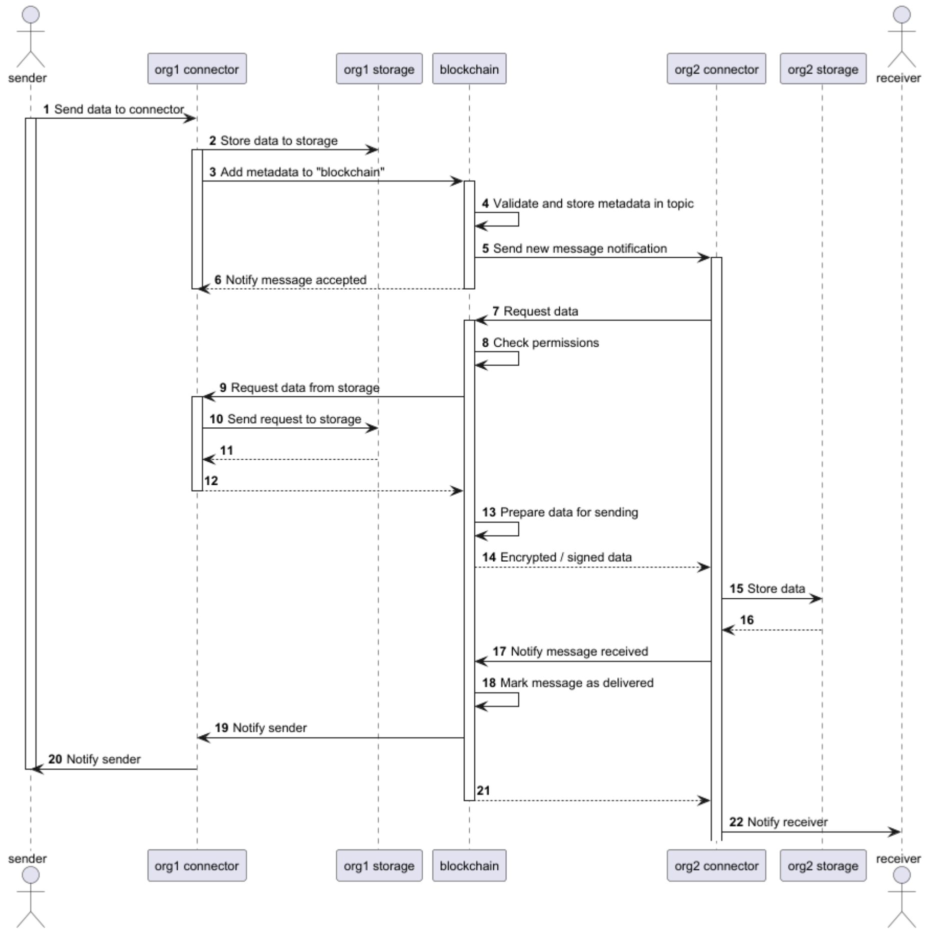
The SDT process is presented in Figure 2. It represents the following algorithm.

The data transfer process commences with the data sender (Org1) initiating a data transfer request to its connector, which serves as the interface to the blockchain. The Org1 connector then persists the data in Org1's storage and gathers associated metadata. Subsequently, the Org1 connector queries the blockchain to initiate the execution of the Secure Data Transfer (SDT) smart contract. The smart contract validates the provided metadata and records it on the ledger. Upon successful metadata validation, the smart contract informs the Org2 connector about the availability of a new message. The smart contract also acknowledges the acceptance of the message delivery to Org1.

To retrieve the data, the Org2 connector queries the blockchain nodes and executes the SDT smart contract. The smart contract verifies Org2's authorization to access the requested data. If authorized, the smart contract requests the data from the Org1 connector. Upon receiving the request, the Org1 connector retrieves the data from Org1's storage and transmits it back to the smart contract.

The smart contract then prepares the data for transfer, which may involve encryption and digital signing depending on the specific smart contract implementation. The prepared data is subsequently returned to the Org2 connector. The Org2 connector then stores the received data in Org2's storage.

To finalize the process, the Org2 connector notifies the blockchain about the message reception via the smart contract. The smart contract updates the message registry, marking the message as delivered. The smart contract also notifies the Org1 connector about the successful completion of the message delivery process. Finally, the Org1 connector informs the data sender about the completion of the data transfer process.



sender
org1 connector
org1 storage
blockchain
org2 connector
org2 storage
receiver
1 Send data to connector
2 Store data to storage
3 Add metadata to "blockchain"
4 Validate and store metadata in topic
5 Send new message notification
6 Notify message accepted
7 Request data
8 Check permissions
9 Request data from storage
10 Send request to storage
11
12
13 Prepare data for sending
14 Encrypted / signed data
15 Store data
16
17 Notify message received
18 Mark message as delivered
19 Notify sender
20 Notify sender
21
22 Notify receiver

Fig. 2. SDT process scheme

Such an algorithm has the following important features:

1. Loose coupling between sender and receiver due to blockchain usage as a message broker.
2. Better fault tolerance compared to regular message brokers.
3. More convenient message broker-like SDT interface compared to existing blockchain solutions.
4. Greater SDT process customization capabilities due to smart contract usage.

4 Implementation details

4.1 Blockchain Selection

Blockchain technology constitutes the core component of the SDT system. Consequently, the specific blockchain implementation exerts a significant influence on key system parameters, including:

- **Throughput:** The volume of data the system can transfer within a defined time interval.
- **Fault Tolerance:** The types of faults the system can withstand and the maximum number of node failures the system can accommodate while maintaining functionality.
- **Scalability:** The maximum permissible number of nodes within the system.

The parameters delineated above are primarily governed by the blockchain type and the employed consensus algorithm.

A review of the existing literature reveals a prevalent trend favoring private blockchain solutions. This preference arises from the necessity for controlled access, enhanced privacy, and the potential for higher throughput in many SDT scenarios, particularly within applications such as IoT and vehicular networks. While several studies mention the potential use of public blockchains, they are not typically presented as the primary option. Specific examples include:

- [14] proposes a peer-to-peer file storage and sharing system based on a consortium blockchain.
- [9] presents a cross-organizational data sharing framework based on blockchain-probes, indicative of a permissioned setting.
- Numerous papers addressing IoT applications (e.g., [1, 8]) often implicitly or explicitly assume a permissioned blockchain context to address security and scalability constraints.

Therefore, this paper posits that a private blockchain is the most suitable option for SDT system implementation.

In addition to selecting the appropriate blockchain type, employing the correct consensus algorithm is crucial. To enhance the SDT system's fault tolerance, this paper proposes the utilization of the Practical Byzantine Fault Tolerance (PBFT) algorithm, aligning with observed trends in the literature. However, PBFT is known to have inherent performance limitations.

Firstly, PBFT implementations necessitate the transfer of substantial data volumes between blockchain nodes to achieve consensus. For instance, the BFT-SMaRT algorithm, employed by Hyperledger Fabric [4], requires each node to transmit 80 MB of data to achieve a performance of 2500 transactions per second. This implies a total traffic volume of 720 MB within a cluster of only 10 nodes during all consensus algorithm stages.

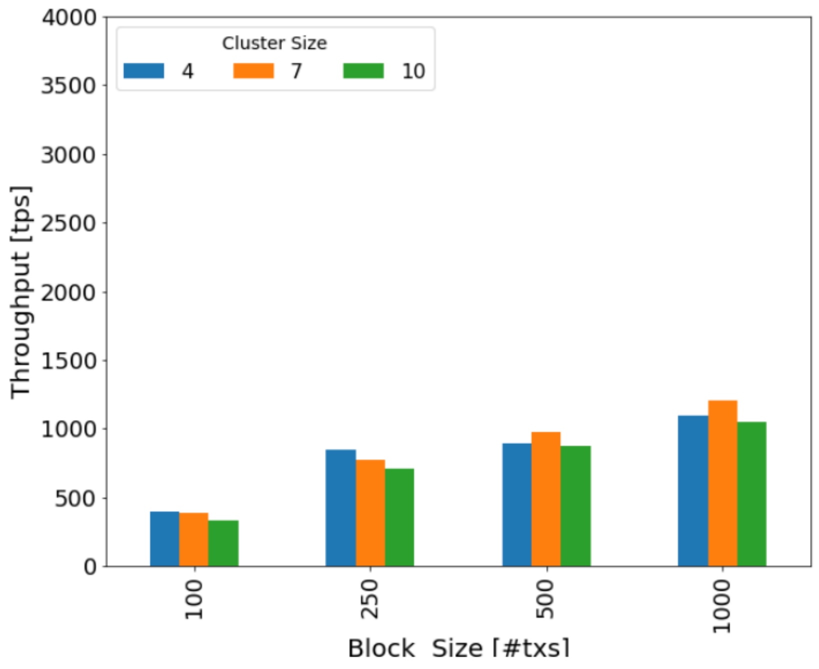


Fig. 3. BFT-SMaRT performance in a global network

Furthermore, the throughput of such a consensus algorithm is contingent upon the number of nodes within the system. Research [4] indicates that BFT-SMaRT achieves maximum throughput when the system comprises 7 nodes (Fig. 3).

Increasing the number of nodes beyond this optimum leads to a degradation in throughput. The maximum permissible number of nodes, according to research [10], is limited to approximately 100.

For the reasons articulated above, a pure PBFT blockchain is not ideally suited for the creation of scalable and fault-tolerant systems. Fortunately, this limitation can be addressed by leveraging the multi-layered blockchain architecture described in [5] and [12], coupled with the data flow separation into topics and partitions, a common practice in message brokers such as Apache Kafka [6].

This paper proposes the use of a two-layered blockchain architecture. The first layer employs the PBFT consensus algorithm. Given the algorithm's known limitations, this layer will be constructed from multiple, relatively small clusters, each consisting of up to 7 nodes. This layer embodies the concept of a partition, representing the smallest system component directly responsible for the execution of SDT smart contracts.

The subsequent layer leverages the RAFT consensus algorithm [13]. This layer is designed for two primary functions: system management (detailed in

Section 4.2) and asynchronous data replication following PBFT consensus rounds (described in Section 4.3). Consequently, this layer is analogous to the topic concept found in message brokers.

This architecture allows for the preservation of all fault-tolerance characteristics inherent in the PBFT consensus algorithm while simultaneously enhancing system scalability. Scaling the system does not necessitate an increase in the PBFT cluster size. Instead, a new, smaller PBFT layer-1 cluster is created and connected to the existing RAFT layer-2 cluster, which has a less stringent limit on the maximum number of nodes.

4.2 Data Flow Separation

This section delineates the proposed data flow separation scheme. As previously mentioned, the data flow of the SDT system is partitioned into topics and partitions.

A topic is defined as a data stream unified by a common theme or a list of consumers. To address the limitations outlined earlier, each topic is further subdivided into a list of partitions. An illustrative example of such a data flow separation into two partitions is depicted in Fig. 4.

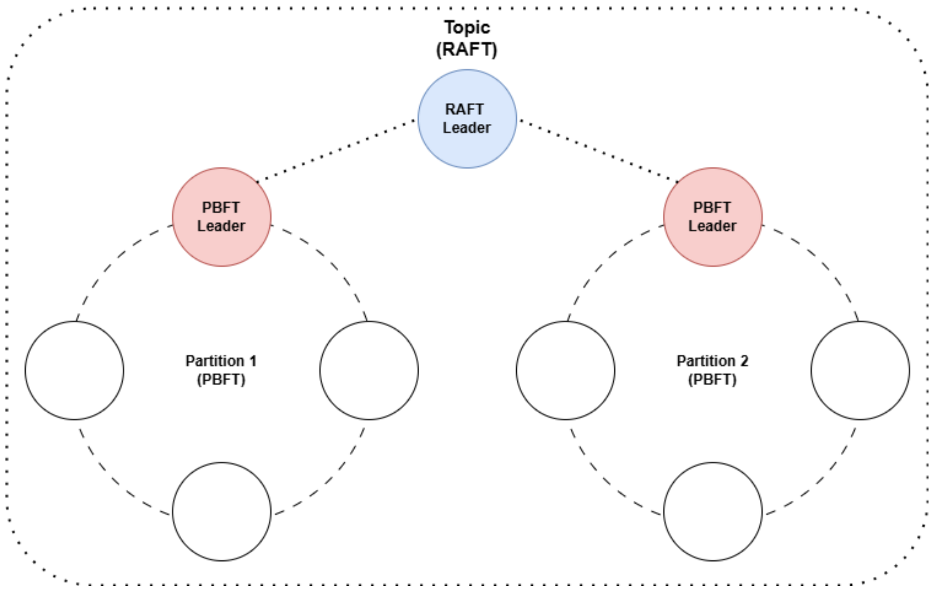


Fig. 4. Structure of a topic with two partitions

Two primary challenges must be addressed to implement this data flow separation:

– How to select a specific blockchain node for SDT process initialization while ensuring effective load balancing?
– How to select the appropriate partition for a specific message?

The selection of a specific load balancing algorithm constitutes a separate area of research and, therefore, falls outside the scope of this article.

Concerning the second challenge, this paper proposes the utilization of hash partitioning, a technique employed by Apache Kafka, for partition selection. The partition is determined by calculating the remainder of dividing the key's hash value by the total number of partitions.

To store the topic configuration data, the layer-2 ledger of the blockchain is proposed. Each node within this blockchain layer stores a list of partitions and a corresponding list of nodes for each partition. Any structural modifications within a topic, such as the creation of a new partition, are to be executed via a dedicated smart contract.

A key advantage of this approach lies in the replication of configuration data across all blockchain nodes. Consequently, data senders can retrieve this data from any topic node. This eliminates the need for dedicated control nodes, such as Apache Zookeeper nodes in Apache Kafka, thereby enhancing system fault tolerance by mitigating the risk of a single point of failure.

4.3 Data Replication

As previously discussed, a primary advantage of utilizing blockchain technology in a data transfer system is its ability to provide verifiable evidence of message delivery by immutably recording the data transfer history within its ledger. Given the system's separation into multiple partitions, verifying message delivery necessitates identifying the specific partition used for the message's transmission. This task is complicated by the fact that the system configuration may evolve over time, potentially requiring an iterative search across all existing partitions. To streamline the message delivery verification process, data replication to all topic nodes is essential.

To facilitate this data replication mechanism, this article introduces the concepts of global and local ledgers. The global ledger, maintained at the topic level, essentially represents the union of all partition-specific ledgers, which we term local ledgers. The global ledger is updated asynchronously with data from the local ledgers through a dedicated periodic task. This task implements the following algorithm:

1. Retrieve the list of partitions from the topic configuration ledger.
2. For each partition:
   (a) Identify the key of the last replicated block.
   (b) Query the local partition ledger to ascertain the presence of new data.
   (c) If new data is found, copy all new data blocks to the global ledger.

An illustrative example of this algorithm's execution for two partitions is presented in Fig. 5. In the figure, the last replicated blocks are highlighted in yellow, while the blocks slated for replication are depicted in green. Prior to the algorithm's execution, there are three blockchain nodes awaiting replication: one from Partition 1 and two from Partition 2. Following the replication process, these blocks are appended to their corresponding blockchains within the global ledger.

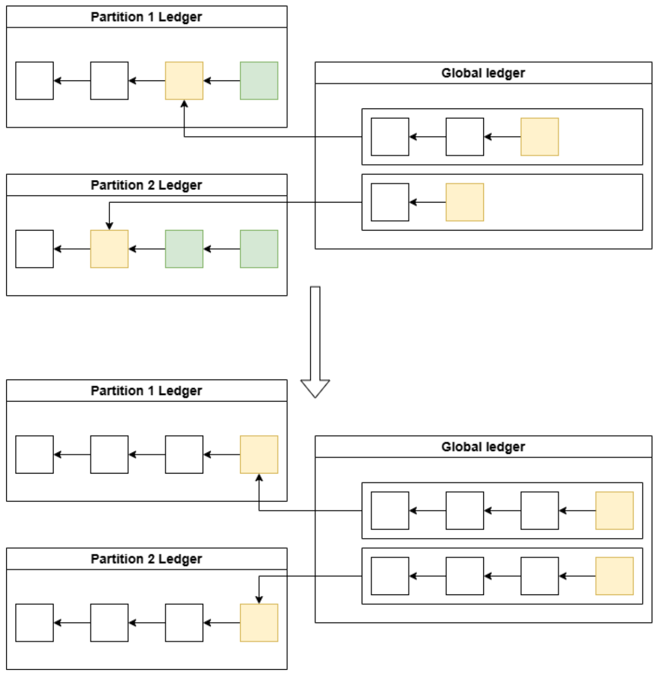


Fig. 5. Data replication to global ledger

Given that the algorithm operates on the second blockchain layer, all modifications to the global ledger are, again, governed by the RAFT consensus algorithm.

5 Summary

The paper proposes a novel architecture for secure data transfer (SDT) systems leveraging blockchain technology. It combines the advantages of two existing paradigms: blockchain as a data access interface and blockchain as a direct data transfer tool. The architecture features a message broker-like design, overcoming data size limitations of existing blockchain-based SDT solutions. The proposed system incorporates blockchain, data storages internal to organizations, and connectors that bridge the data storages with blockchain nodes. The SDT process involves the sender's connector initiating a smart contract on the blockchain, which manages data access, transfer, and confirmation, while the actual data is stored off-chain within the organization's data storage. The paper advocates for a private, two-layered blockchain implementation. The first layer uses PBFT consensus within smaller clusters for efficient message transfer (partitions), while the second layer employs RAFT consensus for system management and asynchronous data replication (topics). This layered approach enhances both fault tolerance and scalability.

Further research is going to focus mainly on the following directions:

1. Implementation of the proposed system and evaluation of the developed prototype.
2. Selection of specific consensus algorithm implementations.
3. Development of a load balancing mechanism for specific partition selection.

References

1. Ai, Z., Cui, W.: A proof-of-transactions blockchain consensus protocol for large-scale iot. IEEE Internet of Things Journal (2022). https://doi.org/10.1109/JIOT.2021.3108621
2. Bagga, P., Das, A.K.: Blockchain for Smart Transport Applications, pp. 125–154. Springer International Publishing, Cham (2022)
3. Bagga, P., et al.: Blockchain-based batch authentication protocol for internet of vehicles. Journal of Systems Architecture **113**, 101877 (2021)
4. Barger, A., et al.: A byzantine fault-tolerant consensus library for hyperledger fabric. In: 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 1–9. IEEE (2021)
5. Bogdanov, A., et al.: Combining pbft and raft for scalable and fault-tolerant distributed consensus. Physics of Particles and Nuclei **55**(3), 418–420 (2024)
6. Foundation, A.S.: Apache kafka documentation. https://kafka.apache.org/documentation/ (nd), accessed: 2025-03-09
7. Ghaemi, S., et al.: A pub-sub architecture to promote blockchain interoperability. arXiv preprint arXiv:2101.12331 (2021)
8. Gupta, S., Yadav, B., Gupta, B.: Security of IoT-based e-healthcare applications using blockchain, pp. 79–107. Springer International Publishing, Cham (2022)
9. Jia, X., et al.: Cross-organisational data sharing framework based on blockchain-probes. IET Networks **12**(2), 77–85 (2023)
10. Ke, Z., Park, N.: Performance modeling and analysis of hyperledger fabric. Cluster Computing **26**(5), 2681–2699 (2023)
11. Kim, Y., Park, J.: Hybrid decentralized pbft blockchain framework for openstack message queue. Human-centric Computing and Information Sciences **10**(1), 31 (2020)
12. Lin, H.Y.: Secure data transfer based on a multi-level blockchain for internet of vehicles. Sensors **23**(5), 2664 (2023)
13. Ongaro, D., Ousterhout, J.: The raft consensus algorithm. Lecture Notes CS **190**, 2022 (2015)

14. Peng, S., et al.: A peer-to-peer file storage and sharing system based on consortium blockchain. Future Generation Computer Systems **141**, 197–204 (2023)
15. Priyadarshini, R., Malarvizhi, N.: Secured data transfer between fog nodes using blockchain. In: Proceedings of the 2nd International Conference on Computational and Bio Engineering: CBE 2020. pp. 417–422. Springer Singapore (2021)
16. Song, R., et al.: A survey of blockchain-based schemes for data sharing and exchange. IEEE Transactions on Big Data (2023)
17. Wang, S., et al.: Bbs: A secure and autonomous blockchain-based big-data sharing system. Journal of Systems Architecture **150**, 103133 (2024)
18. Wang, Y.Y., Huang, S., Yu, X.: An oil and gas big data sharing model based on blockchain technology. IOP Conference Series: Earth and Environmental Science **651**(3), 032105 (2021)
19. Xu, Z., et al.: A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles. Journal of Parallel and Distributed Computing **149**, 29–39 (2021)
20. Yang, H., et al.: A research on the sharing platform of wild bird data in yunnan province based on blockchain and interstellar file system. Sensors **22**(18), 6961 (2022)