GeoBlocks: Trustless Geospatial Data Sharing with Accountability and Decentralized Access Control

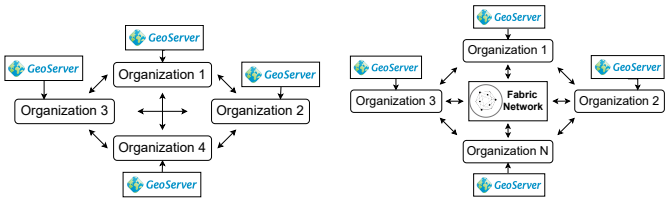
***Abstract*—Multi-organizational geospatial cloud infrastructures often require data sharing across different hierarchies to support geospatial development, decision-making, and collaborative initiatives. Geospatial data exchange between organizations relies on point-to-point communications between siloed GeoServers that fail to support selective data exchange within a large pool of participants, which contain untrustworthy parties that can harm based on the data they distribute or illegitimately obtain. To address such challenges, we propose a reliable open-source permission-based blockchain platform, called *GeoBlocks*, designed explicitly for geospatial data, which stores and enforces Access Control Lists (ACLs), while an application layer facilitates inbound requests from one GeoServer to another based on the ACL. Access control rights are established through individual agreements between organizations and recorded on the blockchain ledger using smart contract transactions. In addition, *GeoBlocks* supports a dynamic programmable dispute resolution mechanism based on the ACL rights. We develop and evaluate *GeoBlocks* with a proof of concept implementation and analyze the query processing performance under different scenarios.**

***Index Terms*—Federated Learning, Blockchain, Local Differential Privacy, Incentivization, Hyperledger Fabric**

I. Introduction

Numerous governmental and private organizations use Geospatial Information Systems (GIS) extensively, enhancing the decision-making process using spatially referenced data over the web [1]. Such organizations often store their unique geographic data on platforms like *GeoServers* [2] that provide a single interface for spatial data storage, processing, and retrieval through spatial queries. However, most organizational GISs are operated on a silo and point-to-point basis; consequently, they support spatial queries only on the local database connected with that organization's GeoServer. Therefore, it becomes difficult to run multi-organizational spatial data queries directly on GeoServers as users often need to develop complex query architecture by dismantling the queries, executing them separately on individual GeoServers, and then aggregating the results [3]–[5]. Such a task is eminently application-specific and often involves manual efforts, significantly increasing the query processing time and affecting the access security, transparency and accountability of the platform [6].

For example, we consider a use-case of communication and information exchange between multiple organizations involving point-to-point sharing, as depicted in Figure 1a. In a traditional data-sharing model, organizations have limited simultaneous access, which lacks control and accountability of the data access for multi-organizational queries that need an organization-defined data control and authentication mechanism while ensuring non-repudiation. Furthermore, the lack of trust is a bottleneck in developing a multi-organizational GIS for organizations having varying data sharing and governance policies. Considering the limitations of such traditional GIS data sharing models, in this paper, we develop the design, architecture, and implementation for a multi-organizational GIS, called *GeoBlocks*, which leverages the distributed ledger technologies (DLT) [7]–[9] to develop a model supporting inter-connectivity and interoperability across multiple organizations requiring complex GIS query processing (Figure 1b).



(a) Traditional model (b) *GeoBlocks*

Figure 1: Comparison of a traditional data sharing model and the *GeoBlocks* prototype.

We make use of blockchain, which is a well-adopted distributed ledger that has been explored for various applications from diverse domains [10]–[13] to ensure trustless data sharing with non-repudiation, transparency, and immutability of records or logs. However, merely interconnecting GeoServers from different organizations over a blockchain network does not solve the problem. Instead, multiple research challenges must be addressed to ensure trustless data access over multi-organizational GIS, where individual organizations maintain their public GeoServers.

Decentralized Access Control: There is a need to develop a decentralized access control over multi-organizational GIS so information access can be granted and transferred seamlessly across organizations while conforming to their data-sharing policies. The existing approaches rely on centralized single-point access control [14], [15], significantly affecting the query execution time due to multiple rounds of ACL invocations during the query processing.

Provenance Tracking: It is essential to track the data access for multi-organizational queries to maintain data provenance, which can help in conflict resolution when the data governance policies of two organizations do not conform.

Dispute Resolution: In the case of an access dispute, it is crucial to resolve the same based on access logs maintained over the blockchain. Developing an automated dispute resolution platform for avoiding *Head-of-line Blocking* during data

access is vital.

Our Contributions. Owing to these challenges of accountable and transparent spatial data access over multiple organizations, we propose *GeoBlocks*, a multi-GIS platform that interconnects multiple GeoServers for seamless data access across organizations. The core idea behind the design of *GeoBlocks* is to integrate the GeoServers over a blockchain-based architecture at the back end and use smart contracts to support access control and data provenance through access logging, along with services for dispute resolution. The novelty of our solution is to explore the features of spatial GIS to develop a reputation system that can be used to discourage participants from launching malicious activities and aid in dispute resolution. Further, the proposed architecture enforces policies through an access control list (ACL) on the private blockchain while recording the requests and responses from individual organizations through a witness cosigning method. In contrast to the existing literature, our contributions to this paper are as follows.

(1) Development of a Multi-GIS Spatial Data Access Platform: To the best of our knowledge, *GeoBlocks* is the first of its kind that develops a multi-organizational GIS architecture by interconnecting the GeoServers from different organizations over a blockchain network. The developed system interconnects GeoServers with a blockchain-based architecture through Hyperledger Fabric [16], to log requests and access for provenance tracking of the data in a trustless architecture.

(2) Development of a Decentralized Access Control List: We design and implement a decentralized access control list over *GeoBlocks* through which access control can be granted and managed with the help of smart-contract executions. The novelty of this proposed solution is to consider witness cosigning while granting access or logging the data requests and access over the multi-organizational GIS platform.

(3) Reputation System for Dispute Resolution: *GeoBlocks* uses a reputation-based architecture to discourage malicious participants from invoking access requests on the platform, which also helps in dispute resolution in case a malicious access is reported, or "*incorrect*" data is being returned. Such a reputation system makes the system robust under a decentralized data access architecture.

(4) Implementation and Evaluation of *GeoBlocks*: We have implemented *GeoBlocks* using Hyperledger Fabric by interconnecting GeoServers over a blockchain network. In addition, we have used Hyperledger Indy to manage the identities of the participating organizations over a decentralized platform architecture to support identity-based access control. The implementation of *GeoBlocks* has been open-sourced and thoroughly evaluated under different scenarios. The implementation indicates that the proposed platform can support accountability and decentralized access control for multi-organizational data access with minimal overhead.

II. Exemplification

Consider a scenario where an organization, say, the *United States Geological Survey (USGS)* continuously needs access to the geospatial data from multiple other organizations as it operates by monitoring and analyzing the Earth's internal interactions to deliver information, serving governments to take immediate actions. Such organizations function by repeatedly accessing and fetching data from other organizations like the *World Meteorological Organization (WMO)*, which provides high-quality authoritative information about the weather, climate, and atmosphere. In a traditional data-sharing mechanism, each organization would require authorization and authentication from the other entities involved. This limits simultaneous access, and since it uses a point-to-point sharing methodology, it lacks control and an accountability mechanism for multi-organizational queries. Moreover, traditional data-sharing models neither provide a mechanism for organization-defined data control nor ensure non-repudiation. This is a bottleneck in developing a multi-organizational GIS for seamless data sharing and query processing across organizations having different data sharing and governance policies.

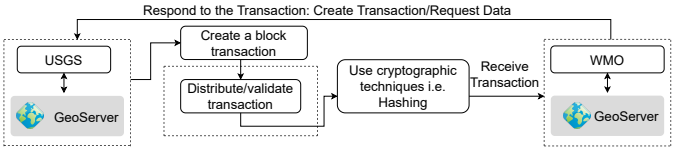


Figure 2: Blockchain-based data sharing model.

GeoBlocks leverages the distributed ledger technologies (DLT) [7]–[9] to develop an inter-connectivity and interoperability model for multi-organizational complex GIS query processing. DLT is the foundation for blockchain technology. Figure 2 shows how a transaction (exchange of information) occurs between one or more organizations using blockchain. It makes use of ledgers that are shared, immutable, and traceable. The following section provides the details of the DLT and its use for decentralized platform development. However, before discussing the details of the decentralized platform architecture, we first highlight the threat model considered in this paper for multi-organization GIS development.

A. Threat Model

Geospatial data belonging to public sector organizations is susceptible in nature. Security and authenticity become a major requirement in such cases. We consider the following types of threats while considering a multi-organizational GIS platform.

1) **Phishing**: Here, the attacker tries to gain access to an entity's credential information. This can be done by sending information from a source that seems to be legitimate but isn't. This makes establishing an entity's identity over the blockchain system highly important and essential for spatial data access.
2) **Sybil Attacks**: To gain influence over a network, the attacker tries to create multiple false identities for malicious reasons, like to launch a denial of service (DoS) attack. Such situations arising in public sector organizations can create problems on a large scale to the level where it might affect inter-country relations.

3) **Non-repudiation**: One of the major issues with multi-organizational GIS is that some Geoservers might exhibit malicious or byzantine behavior by denying the data they shared against a user query. Ensuring data provenance and access tracking over such platforms is important.

We propose *GeoBlocks* considering these security challenges prevalent in a traditional blockchain data-sharing system. *GeoBlocks* uses an Access Control List(ACL) to establish the authenticity of an entity in the peer-to-peer network every time a transaction is made. We next discuss the basic preliminaries and the literature related to spatial data-sharing models and DLTs.

III. Preliminaries and Related Work

This section briefly describes the background of distributed identity and highlights the works related to this paper while summarizing the limitations of the existing literature.

A. Decentralized Identifiers (DID)

An individual identifier intended to be self-sovereign and globally resolvable is known as a decentralized identifier (DID) [17], [18]. Without relying on a centralized authority or registry, DIDs are often used in decentralized identification systems to uniquely identify people, organizations, or devices. Decentralized identifier methods (DID methods) are used to construct DIDs. DID methods specify the guidelines and techniques for establishing, resolving, and managing DIDs.

The *DID method specifier* and the *DID method-specific identifier* are the two fundamental parts of a DID. The method-specific identifier is the unique identifier within the DID method, whereas the method specifier determines which DID method is being utilized. To ensure decentralization, security, and integrity, DIDs use distributed ledger technologies like blockchain. They enable entities/organizations to control their digital identities, providing secure and verifiable interactions across various domains, such as identity verification, access control, and data exchange. A decentralized network or infrastructure is used to resolve a DID. The related DID method's resolver maps the DID to its pertinent data or services connected to the identification, including public keys, authentication mechanisms, or metadata. For example, the "`did:ethr`" (Ethereum DID technique) is built on the Ethereum blockchain, which uses smart contracts to offer a decentralized identity solution. A DID using the "`did:ethr`" looks as follows.

```
did:ethr:0x123456...bcdef
```

Here, the DID method specifier is "`did:ethr`," and the method-specific identifier is: "`0x123456...`." A unique identifier for an entity on the Ethereum blockchain, an Ethereum address, is used as the identifier in this case. The "`did:ethr`" DID allows storing the identifiers' multiple properties and cryptographic keys on the Ethereum network. Public keys used for authentication and verification and other identity-related metadata can be included in these properties. To retrieve the relevant data and cryptographic keys for the identifier while resolving this DID, the appropriate smart contract on the Ethereum blockchain has to be accessed.

B. Related Work

Blockchain and DLTs have been used widely for ensuring coordination and cooperation in multi-organizational scenarios [19]–[23]. Several works [18], [24]–[27] in the literature have focused on designing access control mechanisms on top of decentralized architecture developed using DLTs and blockchains. Cruz *et al.* [28] focused on designing a role-based access control platform using the Ethereum blockchain and Solidity smart contracts. This approach allowed the initialization of roles and implemented a challenge-response protocol for authentication and ownership verification of roles and users. The authors in [29] have reviewed many state-of-the-art systems and organization-specific use cases of ACL. They have also presented a detailed survey of ACL's application in blockchain. Existing works have also demonstrated the advantages of blockchain-based access control solutions over centralized cloud-based ACL architecture [30]. These solutions process access lists on a decentralized ledger by leveraging a blockchain's decentralization and tamper-proof nature, ensuring secure and auditable content access [31]. Another similar solution [32] utilized blockchain to store encrypted data in the cloud and recorded access policies as blockchain transactions. These works provided valuable insights into implementing blockchain access control mechanisms; however, they are primarily developed for one-time authentication and access control over platforms like IoT and smart environments, and do not consider organizational policies during access control.

A few recent works have explored blockchain-based access control for geospatial data or cross-organizational data sharing. Leka *et al.* [33] explored the use of blockchain technology, particularly the Ethereum public ledger, for storing and sharing geographical data. They also incorporated artificial intelligence for automatic data categorization. Their approach aimed to create a shared platform for data producers and consumers in the geographical data domain. In another work [34], the authors have investigated the potential of blockchain technology in addressing coordination and standardization issues in Indian land records management. A recent work [35] has focused on decentralized access control for cross-organizational data sharing by considering the role-based access control (RBAC) model. One of the significant limitations of these works is that they primarily focus on access control mechanisms for data access only and do not consider multi-organizational geospatial query processing scenarios when access control needs to be continuous and should handle query execution policies of individual organizations and the associated disputes at the runtime. A dispute resolution needs to be coupled with access control during the transaction processing, particularly because spatial queries may access data simultaneously from different geoservers having different access policies and then join the results obtained from different databases handled by various organizations. Any dispute during such "`join`" operations (or any other similar queries) must be resolved at

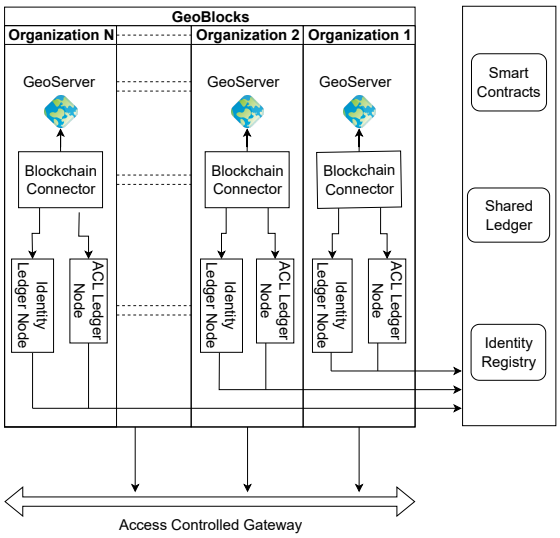


Figure 3: System Overview

the runtime to ensure the robustness and reliability of multi-organization query processing.

IV. System Overview and Design Details

This paper aims to develop a decentralized data-sharing system for trustless spatial data-sharing platforms to share data while ensuring accountability, access control, privacy, and security. We propose *GeoBlocks* to help achieve these objectives by providing a reliable mechanism for dispute resolution without causing long-term harm to the appropriate parties involved.

A. System Architecture

The *GeoBlocks* prototype presents the design of a permissioned ledger-based data-sharing system that can ingest, process, and export geospatial data in a decentralized setting. Figure 3 shows the components of *GeoBlocks*. The smart contracts node manages the *Access Control Lists* (ACL) along with the recorded disputes and assists in constructing a reputation system. The shared ledger node stores all transactions and data, whereas the identity registry database node maintains participating organizations' identities. Each organization consists of an *ACL Ledger Node*, an *Identity Ledger Node*, and an *Access Control Gateway* along with a *Blockchain Connector* and the associated *Geoserver*. Connections between these nodes facilitate data exchange, access control enforcement, and identity verification.

1) Components: GeoBlocks uses several smart contracts responsible for managing critical functionalities of the system, including ACL, dispute management, and reputation system. The shared ledger represents a distributed database that stores transactions and data associated with the blockchain network. The **smart contracts node** is connected to the shared ledger, indicating that the smart contracts interact with and utilize the ledger for transaction processing and information logging. The **identity registry node** functions as a database that stores

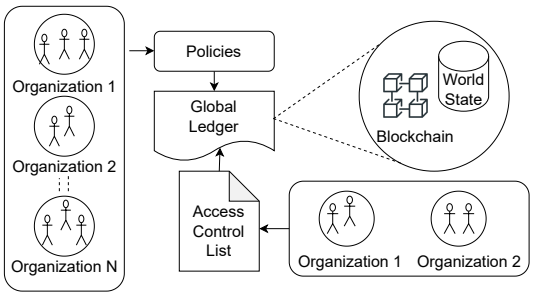


Figure 4: Loading an ACL to the global chain.

the identity of participating organizations in the system. It maintains information about the organizational entities involved (the organizations that participate in *GeoBlocks* through their Geoservers), ensuring proper identification and authentication, where each entity is identified by a **Decentralized Identifier (DID)**. The organizational entities use the shared ledger over a permissioned blockchain network to access multi-organizational data and perform query processing. The **identity ledger node** is connected to the identity registry database node, facilitating the management and verification of participant identities through DIDs. The **ACL ledger node** manages an organization-specific ACL ledger, which contains access control rules and permissions for the organization's data. It is connected to both the shared ledger and the access control gateway. The **access control gateway** controls access to the organization's data. It is connected to the shared ledger through the ACL ledger node, the identity registry through the identity ledger node, and the geoserver that stores and provides access to geospatial data for that organization. Finally, the **blockchain connector** enables communication with the shared ledger, facilitating access control verification and transaction processing. We next discuss how *GeoBlocks* provides access control, policy enforcement through a reputation-based system, and dispute resolution on top of this shared ledger platform.

B. Access Control

To enforce accountability and distributed access control, *GeoBlocks* stores an ACL on the shared ledger over the global blockchain. Figure 4 illustrates loading an ACL in the global blockchain network. Online data-sharing agreements are translated into ACLs and committed to the blockchain through smart contracts. The ACL entry in the ledger includes the identity of the entity being granted access, the identifier and qualifier of the data (based on GeoJSON format for GIS data), validity period, importance, minimum behavior score required for a valid request, and dispute safeguard time before a dispute can be raised. At its core, the ACL entry change requires the agreement of most participants (depending on the consensus algorithm used for the underlying blockchain) and the explicit digital signatures of the corresponding parties who agreed on the change. The process starts with determining and deciding on policies by all participating organizations, which

are then committed to the blockchain. In *GeoBlocks*, an ACL entry sample is as follows.

```
ACL = ("did:fetchdata:fkwbpef023fnvfv",
"org2-56", {"type":"MultiLineString",
"coordinates": [[[-74.0108,40.7093],
[-74.0105, 40.7096]]],
"L111"}, 1664797500, 2, 5000, 300)
```

Here, ''did:fetchdata:fkwbpef023fnvfv' is the DID of the entity which has to be allowed to access the geospatial data from the geoserver of the corresponding organization. 'org2-56' is the data identifier and follows the organization Name-index format, and '–' is concatenated with the organization's name and data index number string. {"type":"MultiLineString", "coordinates": [[[-74.0108,40.7093], [-74.0105, 40.7096]]], "L111"} is the data qualifier which is a tuple of the 'geometry' (GIS data) following the standard GeoJSON format. {*1664797500, 2, 5000, 300*} represent the validity period, importance, minimum behavior score, and dispute safeguard time, respectively. We next discuss the method for enforcing policies based on the ACL entries.

1) Enforceable Policies and Role of Witness: GeoBlocks uses deterministic logic to model the structure of policies to be enforced. These policies are stored in the ACL on a global blockchain, accessible to all participating organizations. The ACL records the information about who has access to which data, which is agreed upon by all the participating organizations. When an entity requests data from another entity, it needs to make an individual request to that specific entity. However, for multi-organizational geospatial queries from an end-user, if the data is accessed from a group of organizations forming an entity, a request must be made to any of the many organizations. Upon receiving a request for data retrieval, the entity owning the data verifies the access rights by referring to the ACL stored on the global blockchain. Suppose the request is valid and the entity has the necessary access rights. In that case, *GeoBlocks* responds by sharing the requested data using the gossip [36] protocol that allows each peer to constantly receive current and consistent data from the other peers.

Additionally, the hash of the data is shared on the blockchain to ensure data integrity and transparency. If, during the process, the ACL check fails and the requesting entity has insufficient access rights, an error response is sent back to the requesting entity indicating the denial of access to the requested data. In cases where the requesting entity believes it has received incorrect data or did not receive the requested data, it can raise a "*dispute*" on the global blockchain. This dispute allows for the issue to be addressed and resolved by involving all relevant parties and ensuring transparency in the resolution process. In *GeoBlocks*, all entities participating in the global blockchain network, except those directly involved in the transaction, act as **witnesses** that play a crucial role in maintaining the integrity and reliability of the system by observing and validating the requests and responses exchanged between the entities. By enforcing policies through the ACL on the public blockchain, recording requests and responses, and involving witnesses in the process, the system aims to ensure accountability, transparency, and data integrity in the geospatial data-sharing environment.

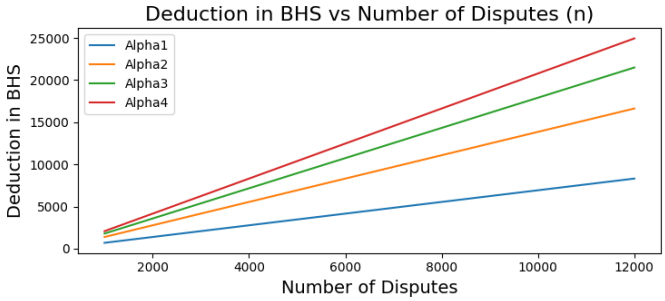


Figure 5: Graphical representation of deductions in BHS.

2) Reputation System: GeoBlocks incorporates a reputation system to discourage participants from engaging in malicious practices and collusion while providing a mechanism for dispute resolution without causing long-term harm to fair parties. The reputation system operates on two key factors: an entity's "*Behavior Score*" (BHS) and the "*importance*" of an ACL entry. Let β be the value for an entity's behavior score, where $\beta_{min} \leq \beta \leq \beta_{max}$, where β_{min} and β_{max} are the minimum and maximum value of β. Similarly, let α be the level of importance for an entity, where $\alpha_{min} \leq \alpha \leq \alpha_{max}$; α_{min} and α_{max} are the minimum and maximum values of α. Notably, α_{min}, α_{max}, β_{min}, and β_{max} are configuration parameters. In our implementation, we consider $\alpha_{min} = 0$ and $\alpha_{max} = 4$. Whenever *GeoBlocks* detects a dispute/misbehavior by an organization (such as trying to falsify the ACL), it updates β as follows.

$$\beta_n = [2\alpha]^{(n-1)/2}$$

where n is the number of valid disputes reported and agreed upon by the network. This formula ensures that the deductions increase exponentially, implying a higher penalty for repeated offenses. This reduction in behavior score increases exponentially with each dispute occurrence, as seen in Figure 5. By implementing this reputation system, *GeoBlocks* incentivizes participants to maintain a positive β and discourages malicious actions. The progressive deductions in β serve as a deterrent, boosting the possibility of redemption and improvement over time.

C. Dispute Resolution

GeoBlocks considers the following types of disputes during data access after/during authentication.

• **False Request Claim Dispute**: In this case, the requesting entity has not genuinely requested the data; however, the receiving entity claims it has asked for one. The requesting entity can raise an invalid dispute, and the entity on the receiving end of the request can only claim that it did not

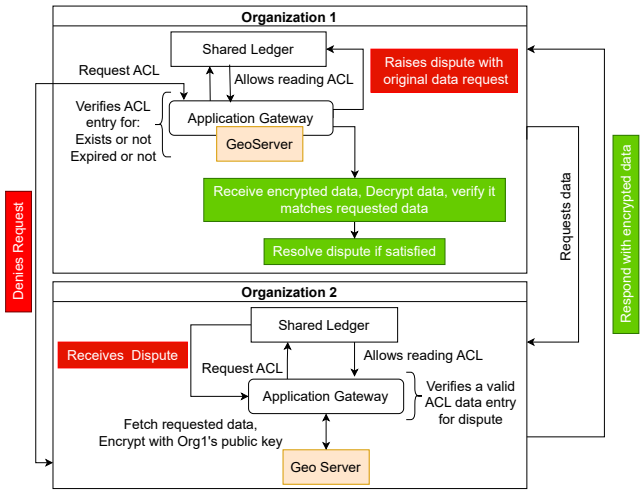


Figure 6: Dispute resolution flow

receive the said request. This claim will be recorded as a response to the dispute in addition to the response data. Such scenarios do not result in a denial of service. Still, repeated disputes of a similar kind can negatively impact the reputation of one entity with no implications for the other entity.

- **Genuine Request Overdue Dispute**: The requesting entity has genuinely requested some data but has not received it. It can raise a dispute, and this claim can be validated via the ledger update.
- **Genuine Request Inadequacy Dispute**: The requesting entity has received data for their request, but it is not what they wanted, or the data is incomplete. This can be verified by matching the hash of the data with the one on the ledger. This requesting entity can then raise a dispute.
- **Synergy Dispute**: A collaborative scenario where data exchange was to take place across entities, but they happen to fall in any of the categories mentioned above. In such cases, both entities can raise disputes for their respective claims.
- **False Chunk Dispute**: The requesting entity claims to receive a fragment of the requested data, which is incorrect, giving rise to a generic case where an entity has multiple issues with the exchange.
- **Common Transaction Dispute**: Two or more entities involved in a transaction raise a dispute for the same event but for different reasons. Each of them can raise a dispute on the global chain and request a resolution for their particular issue by providing relevant proof. We don't treat disputes per transaction as one; instead, each dispute is compared to other disputes in isolation. The witnesses of all disputes will be all the other entities on the blockchain but not involved in the transaction.

Figure 6 depicts the protocol that *GeoBlocks* follows for the dispute resolution mechanism, with an example scenario. Organization 1 (ORG1) requests data from Organization 2 (ORG2), which is denied in case of an invalid request. The Application Gateway of ORG1 verifies the existence and expiry of the requested ACL data entry. If the request is valid, it raises a dispute with ORG2, which it receives through the shared ledger. ORG2 then verifies the ACL data entry and fetches data from its Geoserver as requested. The data is then encrypted and sent back to ORG1, which decrypts the data and checks whether it matches their requirement. Consequently, the dispute is resolved. As a general resolution technique, *GeoBlocks* considers the following two cases for dispute resolution.

Case 1: Hash Mismatch or Incorrect Data Received: The *Genuine Request Inadequacy Dispute*, *False Chunk Dispute* and *Synergy Dispute* fall under this category. The requesting entity (RE) sends a dispute message to the entity from where it has to fetch the data, including the hash of the data received and the expected hash. The data-providing entity (DPE) verifies the hash of the data stored on their system. If the hashes match, the DPE responds to the RE, indicating that the data is correct. If the hashes do not match, the DPE investigates the issue. If the data is incorrect, it sends a corrected version to the RE. The RE then verifies the data and marks the dispute as resolved.

Case 2: Complex Dispute Resolution: The *False Request Claim Dispute*, *Genuine Request Overdue Dispute* and *Common Transaction Dispute* fall under this category. The RE sends a dispute message to the DPE, including a detailed dispute description. The DPE then investigates the dispute. If the DPE can resolve the dispute, they send a resolution message to the RE. Otherwise, they request mediation; the RE and DPE agree on a mediator. The mediators are usually the other entities whose identity has been established over the network. The mediator investigates the dispute and proposes a resolution. The RE and DPE accept or reject the resolution.

Figure 7 depicts the timing diagram for the dispute resolution mechanism implemented in *GeoBlocks*. For highly convoluted disputes, the ultimate resolution is to share the data directly on the global blockchain, where it can be visible to everyone. While there may be better solutions, it ensures no denial of service. This approach does not cause long-term damage to the involved entity because the reduction in behavior score will have a more significant impact on the malicious entity. The assumption is that the involved party will regain its reputation quickly as it will only be involved in a few disputes. Dispute resolution protocols aim to maintain fairness and not sabotage the involved entity while holding the malicious entity accountable for its actions.

V. Implementation and Results

This section discusses the technologies used to implement the *GeoBlocks* prototype, considering the key factors of the access control strategy, i.e., efficiency, security, and privacy. Figure 8 shows the overview of the implementation specifying the technologies used in each component. Communication between organizations occurs using a secure HTTPS protocol. *GeoBlocks* utilizes Hyperledger Fabric [37], a blockchain framework developed by the Hyperledger community[1], to create a secure communication channel among the various entities

[1]https://www.hyperledger.org/ (Last accessed: December 20, 2023)

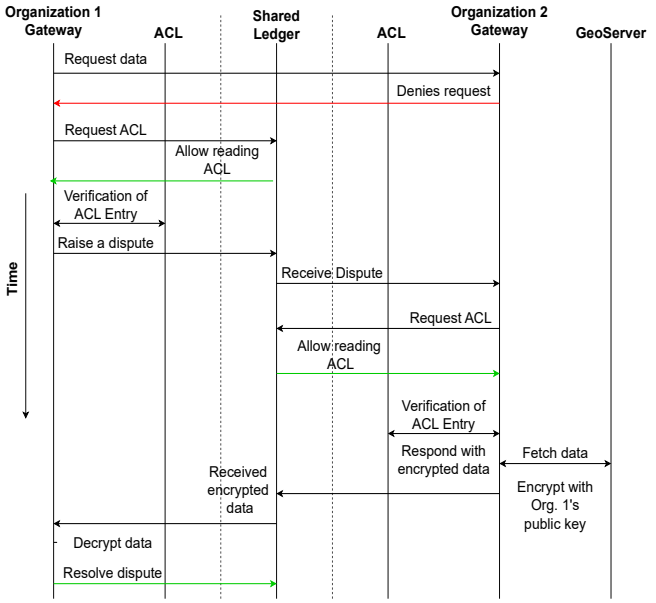


Figure 7: Timing diagram of the dispute resolution process.

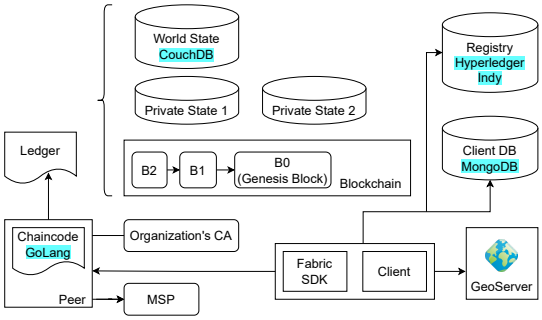


Figure 8: Implementation overview.

involved. Hyperledger Fabric is an open-source platform that offers a modular architecture administering the customization and flexibility in designing access control mechanisms over a permissioned network. It uses smart contracts called Chaincode to implement transactions between numerous entities. *GeoBlocks* utilizes Chaincode written in Golang, which manages the ACL. Moreover, the policies are also written in Golang, which are used to implement the access control system. Golang is preferable due to its performance and concurrency capabilities.

Another essential feature of Hyperledger Fabric is its support for private data sharing. Through *private data collection* (PDC), organizations can selectively share data with specific participants on the network while keeping it confidential from others. This feature enhances privacy and enables controlled data sharing in scenarios where particular data needs to be kept private or shared only with authorized entities. By utilizing Hyperledger Fabric's Chaincode and SDK, the system can efficiently manage the ACL on the ledger, enforce access control policies, and provide a robust and secure environment for data sharing and transactions. *GeoBlocks* leverages the platform's flexibility, programming language support, and privacy features, ensuring the implementation of efficient and secure access control mechanisms.

Hyperledger Fabric supports two types of peer databases, LevelDB [38] and CouchDB[2]. The *GeoBlocks* prototype uses CouchDB to implement the *World State* component, which stores the current values of the set of the ledger state. CouchDB models data using JSON, offering more flexible and rich queries because of its indexing capabilities. *GeoBlocks* also makes use of MongoDB[3], which is an open-source document-oriented database. It falls under the category of NoSQL databases and stores data in a format called BSON, similar to JSON. It offers flexibility in schema design, making it suitable for applications where the data structure may evolve. It is highly scalable and efficient in handling large volumes of data. In addition to storing general metadata, *GeoBlocks* uses MongoDB for storing and managing shareable GIS data.

In *GeoBlocks*, Hyperledger Indy[4] is used to store DIDs, which is a framework that offers tools, libraries, and components for creating digital identities based on distributed ledgers. It enables identity interoperability across different applications and administrative domains. In *GeoBlocks*, the client application (Geoserver) uses the Indy network to associate the *Fabric Membership Service Providers* (MSPs) of other organizations with their real-world identities, enhancing the trust and authenticity of the identity verification process.

A. Proof of Concept and Performance Results

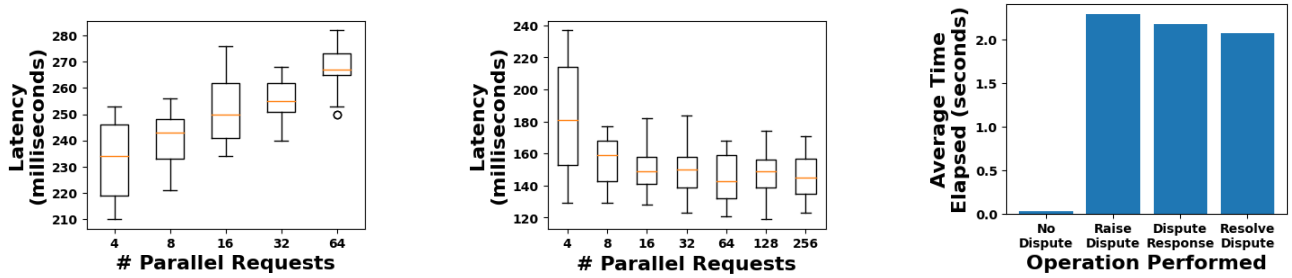
Here, we discuss the implementation of a basic Proof of Concept (PoC) implementation of the *GeoBlocks* and present the performance results obtained from testing the system.

Application Gateway: In *GeoBlocks*, the Application Gateway serves as the central control unit for each organization. It interacts with the GeoServer and the shared ledger (for ACL and Dispute Resolution). It is responsible for external communication, allowing data exchange with other organizations' gateways. The implementation is done using *Node.js*. It handles GIS data translation, encoding data for network transfer, and converting data to GeoJSON format for the dispute resolution smart contract.

GeoServer: GeoServer is the data store for all organization's geospatial data in *GeoBlocks*. It provides an interface for viewing and managing the data. The Application Gateway requests data from GeoServer when sending it to another organization based on ACL conditions. *GeoBlocks* modifies and uses the default GeoServer interface for a convenient and user-friendly interaction.

ACL Smart Contract: The ACL smart contract is deployed on the shared ledger where it stores ACL entries that describe the access rights to data for various organizations. The smart contract ensures that defined rules and permissions govern data access.

[2]https://couchdb.apache.org (Last accessed: December 20, 2023)
[3]https://www.mongodb.com/ (Last accessed: December 20, 2023)
[4]https://www.hyperledger.org/projects/hyperledger-indy (Last accessed: December 20, 2023)



(a) Latency v/s number for parallel requests for fetching different ACL entries

(b) Latency v/s number for parallel requests for fetching same ACL entries

(c) Dispute resolution operations v/s Average time taken in seconds

Figure 9: Performance of *GeoBlocks*

Dispute Resolution Smart Contract: In *GeoBlocks*, the dispute resolution smart contract is also deployed on the same shared ledger. It manages and resolves disputes that may arise among participating organizations. The current implementation follows a simple three-step process: the conflicted entity raises a dispute, the defendant entity responds with data from GeoServer, following which the conflicted entity can mark the dispute as resolved if it is satisfied. These implemented components work together to facilitate the construction of *GeoBlocks*, ensuring trustless data sharing, access control, and dispute resolution within the system.

1) Evaluation Setup: The analysis aims to evaluate *GeoBlocks*'s performance under real-world conditions where different organizations would collaboratively interact to share GIS data and execute spatial queries over multiple spatial databases. *Latency* was chosen as the key performance indicator, directly reflecting the speed at which an organization can execute each system function. The goal was to measure the time taken for critical operations in a multi-organization setting. For the proof of concept (PoC) implementation, we considered four organizations with geoservers, spatial databases, and access control policies with standard benchmarking services [39]–[41]. We use a machine with the following configurations to host the GeoServer and blockchain nodes for an organization – an Intel Core i5 processor, Ubuntu 22.04.1 as the operating system, 16GB RAM, and an SSD with PCIe NVMe interface for storage. Multiple instances of GeoServer, organization peer nodes, and application gateways were run on the same machine using different ports. A request-response model was employed to measure the latency in fulfilling multi-organizational spatial queries (requests) and obtaining operation results. The average time taken for various operations was measured at the application gateways of participating organizations.

2) Results: We conducted experiments by gradually increasing the number of spatial queries triggered in parallel (parallel requests) over *GeoBlocks*. Figure 9a shows that when we initially access the same ACL entries repeatedly, it takes more time to process the request, whereas for the same entry, if we increase the number of parallel requests, the system uses caching, and the overall system latency reduces. Increasing the number of parallel requests beyond a certain point will not significantly increase throughput, elucidating the colossal benefit of *GeoBlocks*. Figure 9b shows the results from experiments conducted for different ACL entry requests in parallel. The curve shows that the latency increases as the number of parallel requests increases. This is because the server has to handle more requests simultaneously, which can lead to congestion. It also shows that the rate of increase in latency slows down as the number of parallel requests increases. This is because the server can optimize its performance for a large number of parallel requests. Overall, the curve shows a trade-off between latency and throughput. Figure 9c depicts the average time taken by an organization in *GeoBlocks* to resolve and respond to disputes. Disputes are expected to introduce a significant time overhead compared to direct data exchange, indicating a consistent increase in interactions with the blockchain network. However, dispute-related operations are expected to occur less frequently than direct transfers; the overhead is deemed acceptable, considering the benefits offered by the system.

VI. Conclusion and Future Works

This paper discussed the design, development, and implementation of *GeoBlocks*, a multi-organizational spatial data sharing and query execution platform, while ensuring accountability, distributed access control, and dispute resolution. Considering the requirements for spatial query execution over multiple data sources simultaneously, our method provides a robust, easy-to-deploy, and scalable solution for multi-organizational data access. We developed efficient smart contracts for loading ACLs based on individual organizations' access control and data-sharing policies, authenticate organizations during query execution, and handle disputes programmatically during multi-organizational data access. Notably, the current PoC implementation validates the data access provenance and analyzes the performance parameters for ACL loading and dispute resolution. However, as a future extension of this work, we plan to implement complex query execution on this platform (like spatial join operation) through smart contracts and develop spatial query optimizations on such decentralized platforms.

REFERENCES

[1] S. A. Elwood, "Gis use in community planning: A multidimensional analysis of empowerment," *Environment and Planning A: Economy and Space*, vol. 34, no. 5, pp. 905–922, 2002.

[2] https://geoserver.org/, 2023, [Accessed 09-Jun-2023].

[3] Z. He, G. Liu, X. Ma, and Q. Chen, "GeoBeam: A distributed computing framework for spatial data," *Computers & Geosciences*, vol. 131, pp. 15–22, 2019.

[4] M. Bakli, M. Sakr, and E. Zimányi, "Distributed spatiotemporal trajectory query processing in SQL," in *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 2020, pp. 87–98.

[5] Q. Xu, L. Xiang, H. Wang, X. Guan, and H. Wu, "GeoMapViz: a framework for distributed management and geospatial data visualization based on massive spatiotemporal data streams," in *IOP Conference Series: Earth and Environmental Science*, vol. 1004, no. 1. IOP Publishing, 2022, p. 012017.

[6] S. Xiong, X. Ouyang, and W. Xiong, "Distributed or centralized: An experimental study on spatial database systems for processing big trajectory data," in *2023 IEEE 8th International Conference on Big Data Analytics (ICBDA)*. IEEE, 2023, pp. 8–13.

[7] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020.

[8] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–41, 2021.

[9] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.

[10] K. Nguyen, G. Ghinita, M. Naveed, and C. Shahabi, "A privacy-preserving, accountable and spam-resilient geo-marketplace," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 299–308.

[11] D. Liu, C. Huang, J. Ni, X. Lin, and X. S. Shen, "Blockchain-cloud transparent data marketing: Consortium management and fairness," *IEEE Transactions on Computers*, vol. 71, no. 12, pp. 3322–3335, 2022.

[12] M. Baza, N. Lasla, M. M. Mahmoud, G. Srivastava, and M. Abdallah, "B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1214–1229, 2019.

[13] R. Shivers, M. A. Rahman, M. J. H. Faruk, H. Shahriar, A. Cuzzocrea, and V. Clincy, "Ride-hailing for autonomous vehicles: Hyperledger fabric-based secure and decentralize blockchain platform," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 5450–5459.

[14] Z. Lv, X. Li, H. Lv, and W. Xiu, "BIM big data storage in WebVRGIS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2566–2573, 2019.

[15] S. Chaudhari, P. Venkatachalam, and K. M. Buddhiraju, "Secure outsourcing of geospatial vector data," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 871–874.

[16] 2023. [Online]. Available: https://www.hyperledger.org/use/fabric

[17] G. Kondova and J. Erbguth, "Self-sovereign identity on public blockchains and the GDPR," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 342–345.

[18] C. H.-J. Braun, V. Papanchev, and T. Käfer, "SISSI: An architecture for semantic interoperable self-sovereign identity-based access control on the web," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 3011–3021.

[19] L. Yuan, Q. He, S. Tan, B. Li, J. Yu, F. Chen, H. Jin, and Y. Yang, "Coopedge: A decentralized blockchain-based platform for cooperative edge computing," in *Proceedings of the Web Conference 2021*, 2021, pp. 2245–2257.

[20] M. Sopek, P. Gradzki, W. Kosowski, D. Kuziski, R. Trójczak, and R. Trypuz, "GraphChain: a distributed database with explicit semantics and chained rdf graphs," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 1171–1178.

[21] C. Aebeloe, G. Montoya, and K. Hose, "ColChain: Collaborative linked data networks," in *Proceedings of the Web Conference 2021*, 2021, pp. 1385–1396.

[22] E. Tallyn, J. Revans, E. Morgan, and D. Murray-Rust, "GeoPact: Engaging publics in location-aware smart contracts through technological assemblies," in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, 2020, pp. 799–811.

[23] E. Tallyn, J. Revans, E. Morgan, K. Fisken, and D. Murray-Rust, "Enacting the last mile: experiences of smart contracts in courier deliveries," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–14.

[24] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered access control framework for smart devices in green internet of things," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 3, pp. 1–20, 2021.

[25] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "IoT passport: A blockchain-based trust framework for collaborative internet-of-things," in *Proceedings of the 24th ACM symposium on access control models and technologies*, 2019, pp. 83–92.

[26] H. Guo, E. Meamari, and C.-C. Shen, "Multi-authority attribute-based access control with smart contract," in *Proceedings of the 2019 international conference on blockchain technology*, 2019, pp. 6–11.

[27] X. Li, Z. Wang, V. C. Leung, H. Ji, Y. Liu, and H. Zhang, "Blockchain-empowered data-driven networks: A survey and outlook," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–38, 2021.

[28] J. P. Cruz, Y. Kaji, and N. Yanai, "Rbac-sc: Role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12 240–12 251, 2018.

[29] L. Golightly, P. Modesti, R. Garcia, and V. Chang, "Securing distributed systems: A survey on access control techniques for cloud, blockchain, iot and sdn," *Cyber Security and Applications*, p. 100015, 2023.

[30] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the iot scenario," *IEEE Access*, vol. 7, pp. 34 045–34 059, 2019.

[31] Q. Lyu, Y. Qi, X. Zhang, H. Liu, Q. Wang, and N. Zheng, "Sbac: A secure blockchain-based access control framework for information-centric networking," *Journal of Network and Computer Applications*, vol. 149, p. 102444, 2020.

[32] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5g," *IET Communications*, vol. 12, no. 5, pp. 527–532.

[33] E. Leka, L. Lamani, B. Selimi, and E. Deçolli, "Design and implementation of smart contract: A use case for geo-spatial data sharing," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 1565–1570.

[34] M. Bal, "Securing property rights in india through distributed ledger technology," 2017.

[35] K. Gai, Y. She, L. Zhu, K.-K. R. Choo, and Z. Wan, "A blockchain-based access control scheme for zero trust cross-organizational data sharing," *ACM Transactions on Internet Technology (TOIT)*, 2022.

[36] K. Birman, "The promise, and limitations, of gossip protocols," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 5, pp. 8–13, 2007.

[37] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[38] L. Wang, G. Ding, Y. Zhao, D. Wu, and C. He, "Optimization of LevelDB by separating key and value," in *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. IEEE, 2017, pp. 421–428.

[39] G. Ingemarsson, "Database performance for GIS: A comparison of database schemas for measurements with spatial attributes," 2019.

[40] O. Zavala-Romero, E. P. Chassignet, J. Zavala-Hidalgo, P. Velissariou, H. Pandav, and A. Meyer-Baese, "OWGIS 2.0: open source java application that builds web GIS interfaces for desktop and mobile devices," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 311–320.

[41] O. Boucelma, M. Essid, and Z. Lacroix, "A WFS-based mediation system for GIS interoperability," in *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, 2002, pp. 23–28.