International Journal of Safety and Security Engineering
Vol. 15, No. 2, February, 2025, pp. 219-231
Journal homepage: http://iieta.org/journals/ijsse

# Decentralized and Secure Access Control Model for Multi-Cloud Data Storage

Mohammed El Moudni*, Elhoussaine Ziyati

C3S Research Laboratory, High School of Technology, Hassan II University, Casablanca 20000, Morocco

Corresponding Author Email: mohammed.elmoudni1-etu@etu.univh2c.ma

https://doi.org/10.18280/ijsse.150203

**Received:** 16 January 2025
**Revised:** 9 February 2025
**Accepted:** 20 February 2025
**Available online:** 28 February 2025

***Keywords:***
*blockchain, cloud storage security, data privacy, access control, multi-cloud*

## ABSTRACT

Cloud storage has transformed data management by offering improved scalability, flexibility, and efficiency. However, as more sensitive information is stored in the cloud, security concerns have grown significantly. Traditional cloud storage systems, which rely on centralized access control policies, are vulnerable to risks, such as unauthorized access and data breaches. This study introduces a blockchain-based approach that enhances access control in multi-cloud storage environments. The proposed model uses a decentralized access control paradigm to improve security, while also creating an immutable record of all data activities. Access permissions are managed dynamically to prevent unauthorized parties from accessing the data and encryption keys. Every access attempt is logged securely, providing a transparent and verifiable history for all user interactions. The layered design of the framework combines strong security, transparency, and scalability, making it well suited for handling sensitive data. The experimental results highlight the efficiency of the system in key areas such as encryption speed, upload latency, and scalability. These findings demonstrate the ability of the framework to address the constraints of conventional centralized access control paradigms, particularly in terms of auditing and logging.

## 1. INTRODUCTION

The rapid growth in cloud storage adoption has changed the landscape of data handling in many sectors [1]. This has simplified data storage and sharing and improved operational efficiency [2]. However, as sensitive data increasingly reside in the cloud, concerns about its security have grown [3]. Centralized data access control models present various risks, including unauthorized access and breaches, which can have significant financial and reputational effects [4].

Blockchain has emerged as a valuable technology for addressing these concerns by providing a decentralized and secure platform characterized by transparency, immutability, and robust cryptographic mechanisms [5]. Its distributed design eliminates the need for a central authority, minimizes risks, and ensures secure authorized access to data [6]. Smart contracts further improve this framework by allowing automated access control, dynamic authorization management, and the keeping of a transparent and auditable log of data access activities [7, 8]. This process of automation reduces reliance on third-party, optimizes access management, and creates a reliable audit trail for compliance and audit purposes [9].

Despite these benefits, the integration of blockchain technology into cloud data access poses considerable challenges. Scalability and latency are major concerns, as well as compatibility with traditional cloud architectures [10]. In addition, consensus mechanisms critical to blockchain integrity can introduce delays that conflict with the time complexity required by cloud services [11].

### 1.1 Our contribution

This paper presents a model developed to improve the security of multi-cloud data storage access by combining decentralized access control with an immutable audit trail. This model relies on a private blockchain for transparency and integrates role-based access control (RBAC) for fine-grained access policies. Smart contracts automate access validation, ensure efficiency, and reduce the processing time. Through implementation and evaluation, this model addresses the main limitations of the existing solutions, providing a good balance between security, scalability, and performance.

### 1.2 Organization of the paper

The structure of this research is as follows. The next section examines the background to this research. The third section presents current research on security in cloud storage, particularly in relation to blockchain-based access control models. Section 4 presents the proposed model and explains its architecture and main components. Section 5 details the experimental results and analysis. Finally, Section 6 concludes the study and suggests future research directions.

## 2. CONTEXT OF THE STUDY

It is important to understand a set of concepts related to the proposed system before exploring the other aspects.

## 2.1 Cloud computing

Cloud computing is a paradigm that allows individuals and companies to obtain a wide range of computing resources, such as storage, processing, and software, via the Internet without depending on the local infrastructure. This approach provides substantial advantages, including flexibility, scalability, and cost efficiency, because users can utilize resources on demand without the burden of maintaining physical hardware or servers [12, 13].

Cloud computing covers three main deployment scenarios: the public cloud offers services provided by third-party vendors and is accessible to anyone on the internet. A private cloud is intended for a single organization, ensuring greater control and enhanced security. Finally, the hybrid cloud combines public and private clouds, enabling organizations to achieve better flexibility and cost optimization [14].

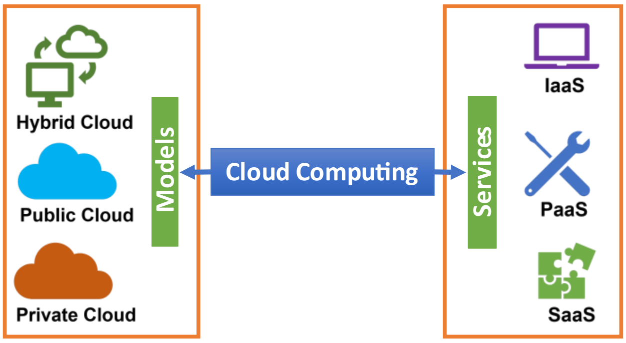


**Figure 1.** Cloud computing

As illustrated in Figure 1, cloud services are grouped into three major paradigms: Infrastructure as a Service (IaaS), which provides shared virtualized computing resources such as storage; Platform as a Service (PaaS), which is designed to offer a framework for developing and deploying applications without the need to maintain the associated infrastructure; and Software as a Service (SaaS), which provides ready-to-use applications that can be accessed via the Internet [15].

### 2.1.1 Cloud storage

Cloud data storage enables users to save and access their data over the Internet, removing their reliance on the local physical infrastructure. This approach provides significant advantages such as the ability to scale storage capacity as required and the convenience of accessing data from anywhere. It also helps reduce costs by eliminating the need for users to manage and upgrade their hardware systems [16, 17].

**Table 1.** Cloud data storage categories

| Type | Principle | Advantages |
|---|---|---|
| **Object** | Stores data as objects, suitable for large and unstructured data. | Highly scalable and cost-effective. |
| **File** | Organizes data in files and directories, commonly used for file sharing and collaboration. | Easy to use, supports hierarchical organization. |
| **Block** | Divides data into blocks, ideal for applications needing high performance and low latency. | Provides high performance and low latency. |

Cloud storage has become an essential part of the modern corporate infrastructure and is widely used, as mentioned in Table 1. By leveraging cloud storage, we can ensure that their data are securely stored and readily accessible [18, 19]. In cloud storage, maintaining multiple copies of data across multiple regions and availability zones is a crucial security strategy known as data redundancy.

As illustrated in Figure 1, cloud services are grouped into three major paradigms: Infrastructure as a Service (IaaS), which provides shared virtualized computing resources such as storage; Platform as a Service (PaaS), which is designed to offer a framework for developing and deploying applications without the need to maintain the associated infrastructure; and Software as a Service (SaaS), which provides ready-to-use applications that can be accessed via the Internet.

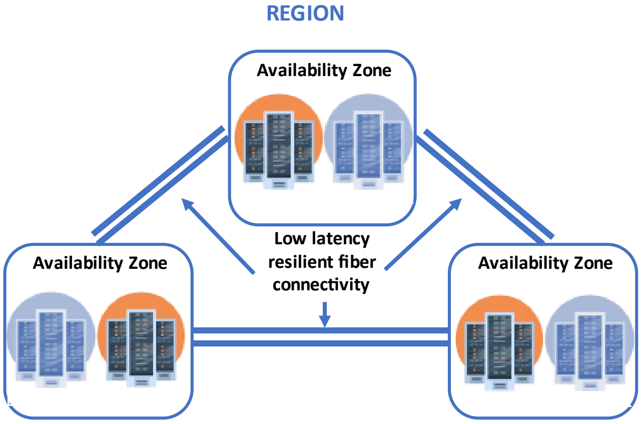


**Figure 2.** Concept of cloud region

As shown in Figure 2, a region in the cloud architecture refers to a specific geographic area with multiple datacenters. Within each region, the availability zones are separate and isolated locations designed to reduce the effects of failures in other zones.

Let $R(C)$ be the set of regions for a CSP $C$:

$$R(C) = \{R_i, R_2, R_3, \dots, R_n\} \;\; with \;\; i \in N, \;\; i \geq 1 \qquad (1)$$

Each region $R_i$ contains a set of availability zones, denoted as $AZ(R_i)$, where each AZ has a minimum of 2 data centers.

$$AZ(R_i) \; = \{AZ_{ij} \big| \, j = 1, \dots, m_i\} \;\; , \;\; m_i \in N, \; m_i \; \geq 2 \qquad (2)$$

Each availability zone $AZ_{ij}$ contains a set of datacenters, $DC(AZ_{ij})$ with at least one datacenter per AZ.

$$DC\left(AZ_{ij}\right) \; = \{DC_{ijk} \big| \, k = 1,2, \dots, k_{ij}\} \, with \;\; k_{ij} \in N \qquad (3)$$

For a given region $R_i$, the set of services $S(R_i)$ available is a subset of all services offered by the CSP. Thus, if a CSP $C$ offers a variety of services, the specific set of services available in each region $R_i$ may differ. Let $S(C)$ denote the set of all services provided by CSP $C$, for each region $R_i$, the set of available services $S(R_i)$ is a subset of $S(C)$.

### 2.1.2 Access control

Access control in cloud computing focuses on determining who can access cloud resources, the conditions under which access is granted, and the actions that they are allowed to perform. It helps to ensure the security of data and upholding the overall cloud security.

As illustrated in Table 2, access control models are often used in the cloud, they can be mathematically expressed, in RBAC, resource access is defined as:

$$Access(U,R) = U_{i=1}^{n}\ Permissions\ (R_i) \\ \text{where } R_i \in Roles(U) \tag{4}$$

where, $U$ is the user, $R$ is the resource, $Permissions\ (R_i)$ refers to the permissions associated with each role $R_i$, and $Roles(U)$ denotes the set of roles assigned to the user. The Discretionary Access Control (DAC) can be represented as:

$$Access(U,R) = Owner\ (R)\ \cup DelegAccess(U,R) \tag{5}$$

where, the resource owner or delegated users determine the access. The Mandatory Access Control (MAC) is as follows:

$$Access(U,R) = Sec_{Level}(U)\ \geq Sec_{Level(R)} \tag{6}$$

**Table 2.** Access control models

| Model | Description | Key Features |
|---|---|---|
| **Role-Based Access Control (RBAC)** | Access is tied to user roles, with permissions linked to roles. | Simplifies management, supports hierarchies, scalable. |
| **Discretionary Access Control (DAC)** | Resource owners control access by granting or revoking permissions. | Flexible but prone to mismanagement risks. |
| **Mandatory Access Control (MAC)** | Central authority enforces access using security labels and classifications. | Strong enforcement, suitable for high-security needs. |
| **Attribute-Based Access Control (ABAC)** | Access is determined by attributes of users, resources, and context. | Fine-grained, dynamic, and adaptable. |

Ensuring that access relies on security classification. Attribute-Based Access Control (ABAC) is modeled as:

$$Access(U,R) = f\ (A_U, A_R, A_E) \tag{7}$$

where, $A_U$, $A_R$ and $A_E$ are attributes of the user, resource, and environment, respectively, and $f$ is a policy function determining access.

## 2.2 Blockchain

This technology employs a decentralized digital ledger to facilitate secure, open, and unalterable record-keeping. Information is stored in interconnected blocks, forming a sequential chain that makes data manipulation virtually impossible without network consensus [20].

In addition to the structure and workflow shown in Figure 3, blockchain employs cryptographic techniques to enhance security and protect data from unauthorized access. Due to these characteristics, blockchain has taken off as a highly attractive technology for a wide range of applications, particularly in cloud data storage, where maintaining data integrity and trust is essential [21].

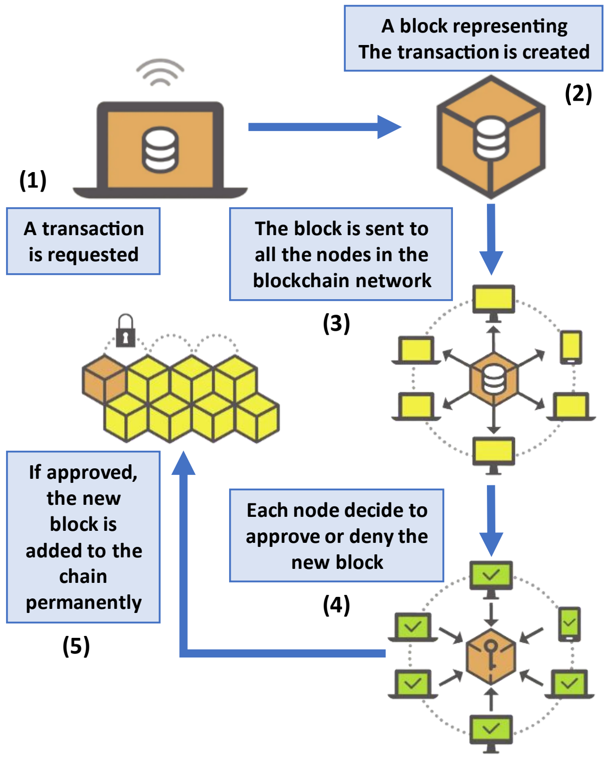

A block representing The transaction is created
(2)
(1)
A transaction is requested
The block is sent to all the nodes in the blockchain network
(3)
If approved, the new block is added to the chain permanently
Each node decide to approve or deny the new block
(4)
(5)

**Figure 3.** General workflow of blockchain transaction

# 3. STATE OF THE ART

Researchers have tackled the challenges of providing secure and efficient access control in decentralized and cloud-based systems. These studies aim to address important issues, such as ensuring data privacy, improving scalability, and optimizing system performance, while suggesting new methods to improve the security and flexibility of access management.

## 3.1 Literature survey

Many researchers have explored the challenges of proposing reliable data control access mechanisms in the multi-cloud, with a particular focus on blockchain-based approaches.

Wang et al. [22] designed a decentralized data access model using Attribute-Based Encryption (ABE) to improve privacy and reduce central dependency. Zhang et al. [23] proposed a hierarchical access control system with blockchain and polynomial commitment for secure authentication, but it needs high computational resources.

Cheng et al. [24] introduced a multi-authority ABAC system with blockchain for secure data sharing in IoT environments, though setting up multi-authorities is complex. Yang and Tsai [25] improved blockchain storage using RBAC and Threshold Secret Sharing, which increased storage efficiency but faced integration challenges. Kanakasabapathi and Judith [26] combined RBAC and ABAC with Key-Policy ABE (KP-ABE) for fine-grained access control, but key management created overhead.

Agrawal et al. [27] used hybrid encryption with public blockchain for fog computing, ensuring data integrity and privacy, but computational demands were high. Rangappa et al. [28] used Sharding ZKP in private blockchain for secure multi-cloud systems, but implementation was difficult. Panda et al. [29] applied CP-ABE in a hybrid ABAC system to provide flexible access control, but key management was a challenge.

Du et al. [30] used Fully Homomorphic Encryption (FHE) in private ABAC for collaborative data sharing but faced high computation costs. Dai et al. [31] applied blockchain-based ABAC to digital twins, enabling scalable systems but requiring significant resources. Neela [32] used RBAC with Hash Authentication Codes for secure outsourcing but had performance issues under high traffic.

Liu et al. [33] proposed DAC with Proxy Re-Encryption for secure medical data sharing, though managing re-encryption was complex. Singh and Singh [34] implemented ABAC with RSA for IoT, supporting privacy but struggling with high-volume environments. Ullah et al. [35] used fine-grained control with ECC for lightweight IoT security, but it lacked scalability.

Roy and Ghosh [36] applied ABAC with SMPC to ensure privacy and secure storage, but it caused delays in real-time operations.

Das et al. [37] used SMPC in ABAC for secure data storage, achieving privacy but increasing computational overhead. Singh and Rathee [38] integrated smart contracts with ABAC and RSA for healthcare systems, enabling dynamic consent but with implementation difficulties.

Rajkumar et al. [39] combined RBAC and ABAC with RSA and AES for scalable IoT systems, though the setup was complex. Bhatt et al. [40] used ABAC with ABE in public blockchain for AWS IoT, offering strong security but requiring heavy resources for large-scale deployments.

### 3.2 Discussion

Table 3 includes several abbreviations, defined here for clarity. Attribute-Based Access Control (ABAC), Role-Based Access Control (RBAC), and Discretionary Access Control (DAC) are different models for managing user permissions and access to cloud resources, as discussed in Table 2. Cryptographic methods like Secure Multi-Party Computation (SMPC) and Fully Homomorphic Encryption (FHE) help maintain privacy by allowing computations directly on encrypted data. Attribute-Based Encryption (ABE) and its forms, Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE), ensure secure sharing of data based on user attributes. Zero-Knowledge Proof (ZKP) allows one to confirm information without disclosing sensitive details.

**Table 3.** Comparative analysis of proposed blockchain-based access control models

| Paper | Access Control Mechanism | Blockchain Model | Cryptographic Technique | Advantages | Disadvantages |
|---|---|---|---|---|---|
| **[22]** | ABAC | Private | Attribute-Based Encryption (ABE) | Efficient attribute privacy and decentralized control | Overhead in managing attributes |
| **[23]** | Hierarchical ABAC | Public | Polynomial Commitment | Secure authentication and hierarchical management | Computational resource consumption |
| **[24]** | Multi-Authority ABAC | Hybrid | Attribute-Based Encryption (ABE) | Decentralized and multi-authority data sharing | Complex multi-authority setup |
| **[25]** | RBAC | Consortium | Threshold Secret Sharing | Optimized storage and enhanced blockchain security | Cold and hot block integration challenges |
| **[26]** | Hybrid (RBAC+ABAC) | Consortium | KP-ABE | Fine-grained access and data sharing | Overhead in key management |
| **[27]** | ABAC | Public | Hybrid Encryption | Integrity and privacy for distributed fog data | High computational demands |
| **[28]** | ABAC | Private | Sharding ZKP | Strong anonymity and security in multi-cloud | Implementation complexity |
| **[29]** | ABAC | Hybrid | CP-ABE | Flexible and decentralized cloud control | Key management challenges |
| **[30]** | ABAC | Private | Fully Homomorphic Encryption (FHE) | Collaborative access for hierarchical data | High computation costs |
| **[31]** | ABAC | Private | Attribute-Based Encryption (ABE) | Scalable and secure access for digital twins | Resource-intensive |
| **[32]** | RBAC | Public | Hash Authentication Codes | Integrity and outsourcing security | Performance degradation under high requests |
| **[33]** | DAC | Public | Proxy Re-Encryption | Efficient medical data sharing | Infrastructure for re-encryption |
| **[34]** | ABAC | Consortium | RSA | Dual-data storage with privacy preservation | Performance limitations in IoT |
| **[35]** | ABAC | Public | Fine-Grained Mechanisms | Fine-grained control for IoT | Overhead in IoT computations |
| **[36]** | ABAC | Public | Elliptic Curve Cryptography (ECC) | Lightweight IoT access control | Scalability limitations |
| **[37]** | ABAC | Hybrid | SMPC | Privacy-preserving and secure storage | Latency in real-time operations |
| **[38]** | Smart Contract-Based ABAC | Hybrid | RSA with Smart Contracts | Scalable healthcare access control | Dynamic consent complexity |
| **[39]** | Hybrid (RBAC+ABAC) | Public | RSA with AES | Decentralized and scalable IoT management | Setup complexity |
| **[40]** | ABAC | Public | Attribute-Based Encryption (ABE) | Secure access for AWS IoT | Resource-heavy operations |

Other encryption techniques, including Rivest–Shamir–Adleman (RSA), Advanced Encryption Standard (AES), and Elliptic Curve Cryptography (ECC), protect data through encryption. Additional methods such as Polynomial Commitment, Proxy Re-Encryption, Fine-Grained Mechanisms, Sharding, and Hash-based Message Authentication Codes (HMAC) enhance the overall security, scalability, and integrity of data

As outlined in Table 3, the existing literature review offers a comprehensive analysis of blockchain-based access control systems for the cloud. These approaches, categorized by the access control models discussed in Table 2, include RBAC, DAC, MAC, ABAC, and hybrid combinations. They leverage various blockchain models, such as Private, Public, Hybrid, and Consortium models, to address different security and scalability requirements. Among these, ABAC and hybrid models are prominent because of their ability to handle complex policies and offer fine-grained control. In addition, techniques such as Zero-Knowledge Proofs (ZKPs) and Attribute-Based Encryption (ABE) are frequently applied to enhance privacy and data security.

Nevertheless, these approaches face challenges such as high computational costs from cryptographic methods, such as homomorphic encryption, and limitations in scalability caused by resource-intensive consensus mechanisms. While Private and Consortium blockchains improve control and efficiency, Public and Hybrid models provide decentralization but introduce higher complexity.

## 4. PROPOSED SYSTEM

### 4.1 General overview

The proposed model uses a set of layers integrated with a private blockchain network to decentralize the access control and implement an immutable audit trail.

The system is built on a layered architecture that offers strong security, transparency, and scalability for managing sensitive data. It is designed to dynamically control user access, while securely storing and retrieving data from multiple cloud service providers. Before detailing the workflow, it is crucial to provide a general overview of how the framework works, highlighting the interactions between users, blockchain, and cloud storage systems.

As depicted in Figure 4, the workflow of the proposed framework consists of sequential steps ensuring secure data access, validation, and storage within a decentralized multi-cloud environment.

- **User Layer**: The data owner (DO) pushes data along with predefined access policies to the system, while the data requester (DR) submits an access request containing the user ID, resource ID, action, and context.
- **Data Gateway**: The request is intercepted and processed through the API Gateway, which forwards it to the Access Module within the Policy Validation Layer for verification.
- **Policy Validation Layer**: The Access Module verifies the request's authenticity by checking the user ID and digital signature. The request is then validated against the stored access policies using smart contracts in the Blockchain Module. The validation result (granted or denied) is immutably recorded on the blockchain.
- **Blockchain and Multi-Cloud Storage Interaction**: Upon successful validation, the request is forwarded to the Multi-Cloud Storage Layer, where encrypted data is securely stored or retrieved. Each Cloud Service Provider (CSP) enforces access control via Role-Based Access Control (RBAC) and Key Vaults for secure key management. The blockchain logs access events, ensuring traceability and security.

This integration between the Blockchain Module and Multi-Cloud Storage Layer ensures a decentralized, tamper-proof access control mechanism while maintaining data integrity and security across cloud providers.

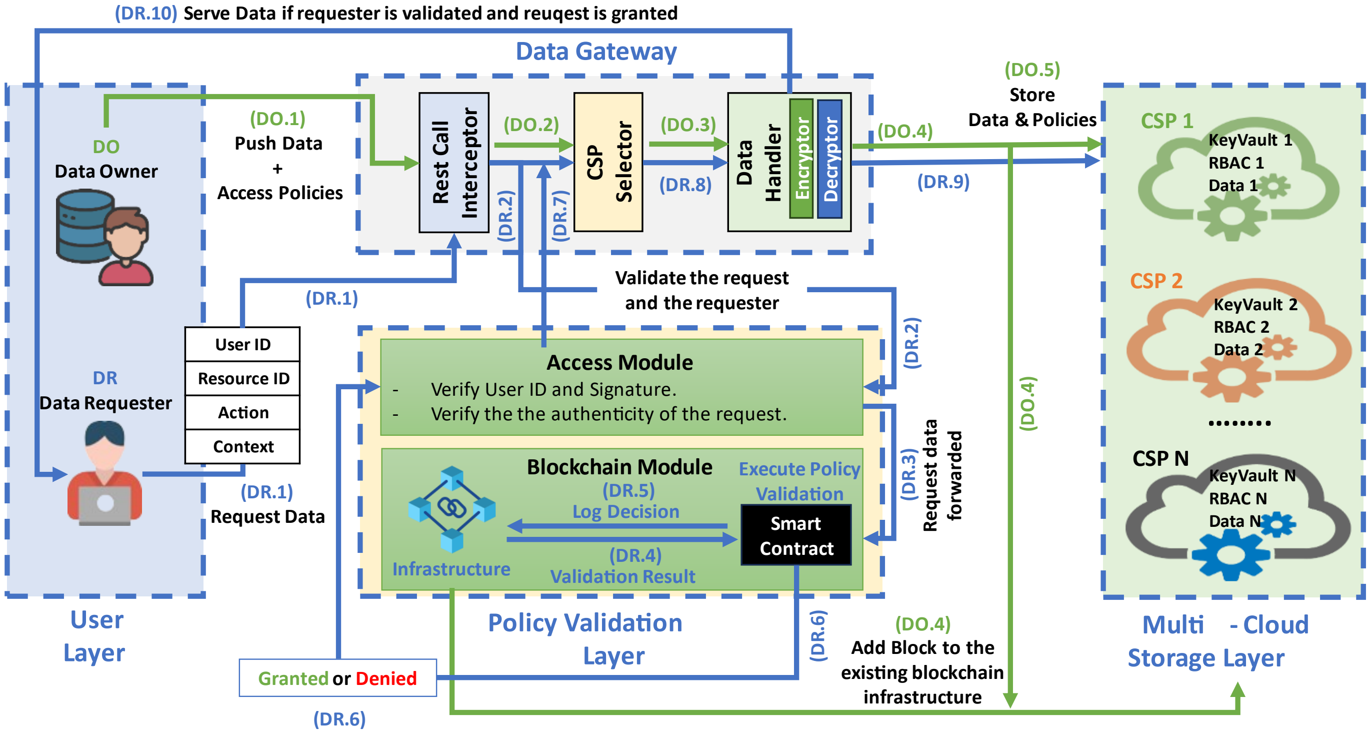


**Figure 4.** Architecture of the proposed system

## 4.2 Components

### 4.2.1 User Layer

The User Layer acts as the primary interface between users (Data Owners and Data Requesters) and the system. The Data Owner (DO) uploads data along with predefined access policies, specifying the conditions under which data can be accessed. These access policies are stored securely in the blockchain infrastructure. However, the Data Requester (DR) submits an access request that includes key attributes to rule the request.

Let $R$ represent the access request, where:

$$R = (UID, RID, Act, ctx) \tag{8}$$

$UID$ is the user identifier, $RID$ is the Resource identifier, $Act$ is the action to be performed over the resource and $ctx$ is the context of the request, especially the location of the requester and timestamp of the request. The User Layer forwards these requests to the Data Gateway for validation and processing.

### 4.2.2 Data Gateway

The Data Gateway Layer manages the flow of requests between the User Layer and the cloud storage system. It begins by intercepting all REST calls from users and validating the requests through signature and authenticity checks. Valid requests are forwarded to the CSP Selector, which identifies the optimal region for each cloud service provider (CSP) based on the proximity to the user and cost efficiency. The selected regions are then associated with the data owner for efficient storage and retrieval. Finally, the Data Handler processes data access requests, reconstructing data from the primary storage and its replicas across the selected CSP regions to ensure reliability and scalability.

Rest Calls Interceptor: The Request Interceptor is the first component of the Data Gateway Layer, which is responsible for capturing all REST API calls from the User Layer. Each access request is validated by checking the digital signature for authenticity and ensuring that the request is fresh and not replayed. If the request passes these validations, it is forwarded to the CSP Selector for further processing; otherwise, it is rejected.

***Algorithm 1:** Optimal CSP Selection Algorithm*

***Input:*** CSPs (list of cloud service providers), Regions(CSP_i) (regions for CSP_i), L (Data Owner's location), α (proximity weight), β (cost weight)

***Output:*** Selected_Regions (optimal region for each CSP)

```
Initialize Selected_Regions ← ∅
for each CSP_i in CSPs do
    Initialize Min_Scoring ← ∞
    Initialize Best_Region ← NULL
    for each region r in Regions(CSP_i) do
        Cost ← Cost(r)
        Proximity ← Distance(L, r)
        Scoring ← $\alpha$ * Proximity + $\beta$ * Cost
        if Scoring < Min_Scoring then
            Min_Scoring ← Scoring
            Best_Region ← r
        end if
    end for
    Add Best_Region to Selected_Regions
end for
return Selected_Regions
```

CSP selector: The CSP Selector determines the optimal storage or retrieval regions across multiple cloud service providers (CSPs). For data storage, regions were selected based on proximity to reduce latency and cost efficiency to minimize expenses. For data retrieval, the appropriate CSPs and regions in which the data are stored are identified.

The CSP Selector identifies the most suitable region from each cloud service provider (CSP) based on two key criteria:

- The nearest regions over all CSPs are selected based on the Data's location to reduce latency.
- Choosing the region with the cheapest storage service to ensure cost efficiency.

Using these filters, it selects one region per CSP to ensure a low latency for data access and minimal storage costs. The selected regions are then linked to the data owner for efficient future storage and retrieval operations.

For each CSP $CSP_i \in CSPs$ , the selected region $Rn_s(CSP_i)$ must satisfy:

$$Rn_s(CSP_i) = arg_{\,r \in Rn(CSP_i)} \min\ (\alpha . Dc(L, r) + \beta . Ct(r)) \tag{9}$$

where, $Rn(CSP_i)$ is the set of regions composing $CSP_i$, $L$ is the location of the Data Owner, $Dc(L, r)$ is the distance between $L$ and region $r$, $Cost(r)$ storage cost is region $r$, $\alpha$ and $\beta$ are weights for prioritizing proximity and cost efficiency. The weights α and β in Eq. (9) help balance speed and cost when selecting a cloud storage region. A higher α prioritizes lower latency for faster data access, while a higher β focuses on reducing storage costs. These values can be adjusted based on user needs or Service Level Agreements (SLAs). Experimental evaluations demonstrated that higher α values reduce retrieval time but may increase costs, whereas higher β values lower expenses but can introduce latency. This balance allows the model to adapt to different multi-cloud storage requirements.

The combined CSP selection $CSP_{filtered}$ includes the selected regions for all CSPs:

$$CSP_f = \{Rn_s(CSP_1), Rn_s(CSP_2) \ldots, Rn_s(CSP_n)\} \tag{10}$$

### 4.2.3 Data Handler

The Data Handler Module securely manages data storage and retrieval. For uploads, data are encrypted using a symmetric key derived from the Data Owner ID and Resource, distributing the encrypted data and policies across the selected CSP regions. The symmetric key is also encrypted with an asymmetric public key and stored in regional Key Vaults for redundancy. During retrieval, the module verifies the access permissions and, if granted, retrieves and decrypts the key to reconstruct the data, ensuring secure and compliant access.

As depicted in Figure 5, during the encryption phase, a symmetric key $K_s$ is dynamically generated using the secure key derivation function HKDF. This key is derived from immutable attributes, Data Owner ID and Resource ID, ensuring a unique key for each resource. The data is then encrypted with $K_s$ using AES, which is efficient and suitable for handling large datasets. To further secure the encryption key, $K_s$ is encrypted with an asymmetric public key $K_{pub}$, and the resulting $K_{s\_encrypted}$ is stored in redundant Key Vaults across the selected regions. The encrypted data $D_{encrypted}$ is distributed to the selected regions.

The decryption phase begins only after the access request is authorized by the Access Validation Layer. The Data Handler retrieves the encrypted key $K_{s\_encrypted}$ from the Key Vault in the region hosting the data. This key is then decrypted using the corresponding private key $K_{priv}$, restoring the symmetric key $K_s$. Finally, $K_s$ is used to decrypt the data, and reconstruct the original dataset for the authorized requester.

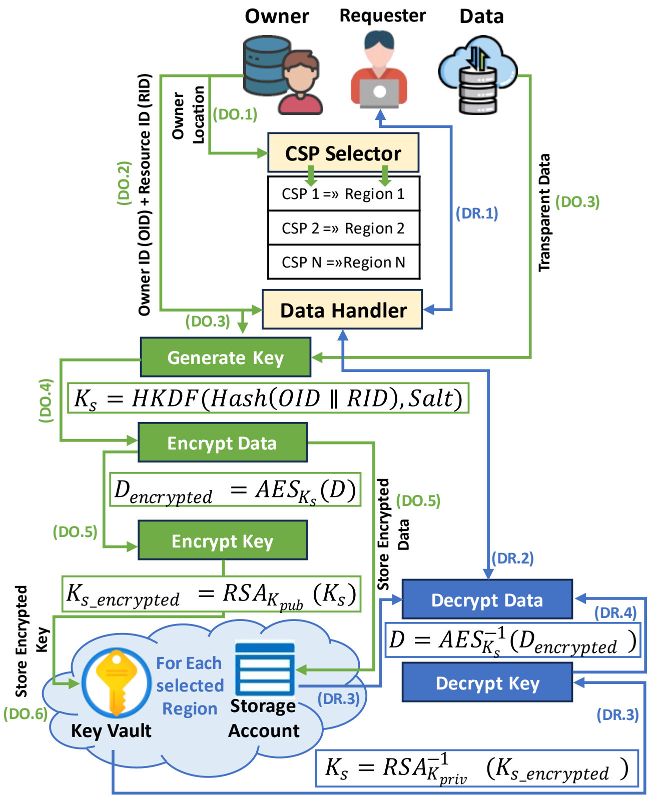


**Figure 5.** Workflow of data handler module

By distributing $D_{encrypted}$ and $K_{s\_encrypted}$ across multiple regions, the system ensures redundancy. Encryption and decryption operations provide robust confidentiality, leveraging symmetric encryption for performance and asymmetric encryption for secure key management.

### 4.2.4 Access Validation Layer

The Policy Validation Layer is the core of the framework and ensures secure and accurate access control for all the requests. It acts as the decision-making hub, validating access requests against predefined policies through a two-step process involving the Access Module and Blockchain Module. This layer guarantees that only legitimate and authorized requests are granted access, leveraging the blockchain for transparency and immutability.

Access Module: The Access Module performs the initial validation of requests by verifying the identity and attributes of the user. It ensures the request's legitimacy by checking its format, authenticity, and compliance with basic access requirements. Validated requests are then forwarded to the Blockchain Module for further policy evaluation.

Blockchain Module: The Blockchain Module evaluates the access requests against smart contract-defined policies stored in the blockchain. It executes these contracts to determine whether the request complies with the policies, logs the decision immutably on the blockchain, and returns the access decision to the Access Module for further action. This ensures transparency and secure policy enforcement.

As illustrated in Figure 6, the proposed topology functions as a private blockchain network with Ganache serving as the centralized coordination hub. Data owners are represented as nodes that connect to the central ledger by using the Web3.py Python library to send transactions and receive updates. Ganache manages the blockchain ledger to ensure synchronization across all nodes. This setup simulates decentralization while maintaining simplicity and efficiency for experimental purposes.

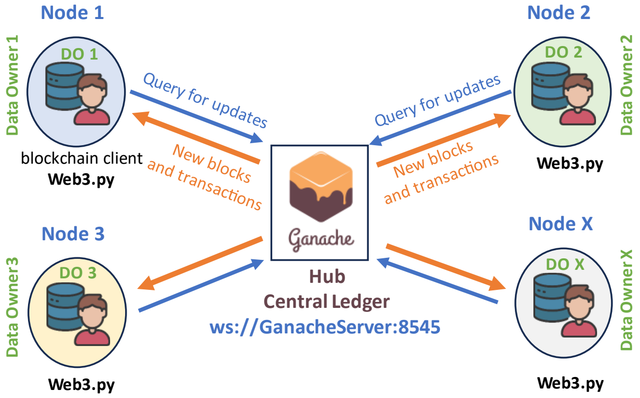


**Figure 6.** Interactions of the proposed private blockchain

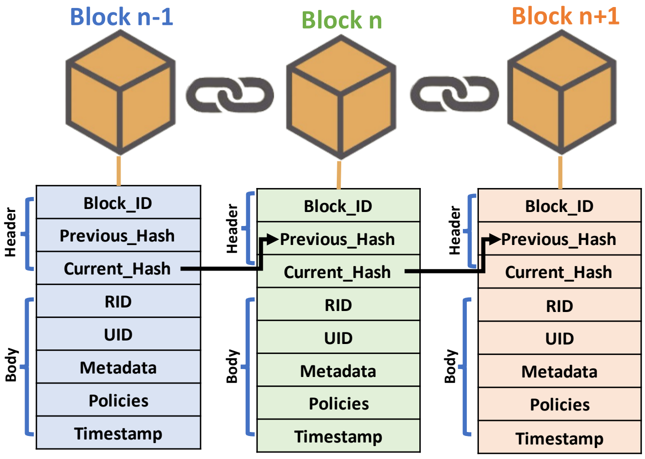


**Figure 7.** Structure of the proposed transaction block

**Table 4.** Block attributes

| Field | Description |
|---|---|
| Block_ID | A unique identifier for the block. |
| Previous_Hash | The hash of the previous block. |
| RID | The Resource identifier. |
| UID | The unique identifier of the Data Owner. |
| Metadata | Details about the data. |
| Policies | Access control rules. |
| Timestamp | The time when the block was created. |
| Current_Hash | The hash of the current block. |

In the proposed framework, data metadata and access policies are stored in the blockchain as structured blocks, as depicted in Figure 7. Each block contains key details, including the Resource ID (RID), metadata regarding data storage and replication across CSPs, and programmatically defined access policies. The metadata include information such as the primary CSP, replication locations, encryption keys, and timestamps, while the policies specify rules such as user roles, conditions, and allowed actions.

Additionally, as depicted in Table 4, the User ID (UID) is stored in the block to link the data and policies to the Data Owner, ensuring accountability and traceability. The blocks are formatted in a JSON structure for flexibility and seamless execution using smart contracts.

***Algorithm 2:*** *Smart Contract Algorithm*

***Input:*** R = (UID, RID, Act, ctx) (Access Request), P (Access Policies)

***Output:*** Decision ("Granted" or "Denied")

```
P $\leftarrow$ GetPolicies(R.RID)
ConflictingPolicies $\leftarrow$ DetectConflicts(P)
if ConflictingPolicies $\neq \emptyset$ then
    ResolvedPolicy $\leftarrow$ ResolveConflicts(ConflictingPolicies)
else
    ResolvedPolicy $\leftarrow$ P
end if
if (R.UID $\in$ ResolvedPolicy.Allowed_Users) AND
(R.Action $\in$ ResolvedPolicy.Allowed_Actions) AND
(R.Context satisfies ResolvedPolicy.Conditions) then
    Decision $\leftarrow$ "Granted"
else
    Decision $\leftarrow$ "Denied"
end if
Log $\leftarrow$ {UID: R.UID, RID: R.RID, Action: R.Action,
Decision: Decision, Timestamp: CurrentTime()}
Blockchain.Append(Log)
return Decision
```

The proposed smart contract algorithm automates access control decisions in multi-cloud storage by evaluating requests against policies stored on the blockchain. It first retrieves relevant policies and detects any conflicts. If conflicts exist, a resolution mechanism prioritizes the appropriate policy. The algorithm then verifies user permissions, requested actions, and contextual conditions before granting or denying access. The final decision is immutably logged on the blockchain, ensuring transparency and accountability.

4.2.5 Multi-Cloud Data Storage Layer

This layer manages data securely across multiple cloud service providers. For storage, the data are encrypted, replicated, and distributed to regions selected by the CSP Selector. Policies are defined in each CSP's Identity and Access Management engine, and are linked to the data. During retrieval, the Data Handler reconstructs the data using metadata from the blockchain.

## 5. EXPERIMENTAL DESIGN AND RESULTS

### 5.1 Experimental design

5.1.1 Setup

The experimental setup was conducted on a Windows 11 laptop equipped with an Intel Core i7 processor, 32GB of RAM, and 500GB of storage. The system was connected to the internet with a 400 Mbps download speed and 50 Mbps upload speed.

As illustrated in Figure 8, the User Layer employed Postman to generate and send API requests on behalf of both data owners and requesters. The Data Gateway Layer leveraged an API Gateway cloud service to intercept REST calls, while Serverless Functions facilitated Cloud Service Provider (CSP) selection and data processing.

The Policy Validation Layer was implemented using Ganache as a private Ethereum blockchain, configured with the following parameters: Network ID: 5777, Gas Limit: 8,000,000, Gas Price: 20 Gwei, and Block Gas Limit: 10,000,000. The blockchain network included 10 pre-funded accounts, each allocated 100 ETH, and was operated in deterministic mode to ensure the reproducibility of access control transactions. This layer also incorporated Serverless Functions for policy enforcement and validation.

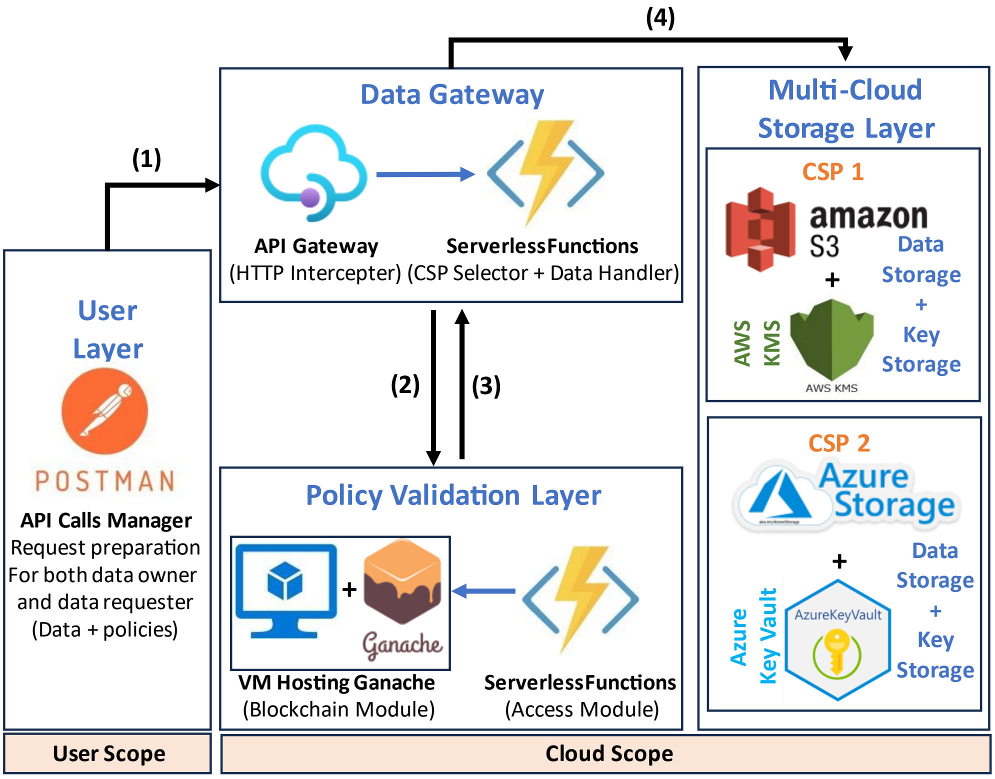


**Figure 8.** Design of the experimental setup

The Multi-Cloud Storage Layer was simulated using Amazon S3, integrated with AWS KMS for secure key management, and Azure Blob Storage, which utilized Azure Key Vault for enhanced key security. This setup ensured a robust and scalable environment for evaluating the access control framework within a multi-cloud storage system.

### 5.1.2 Model evaluation

Performance: This section evaluates key performance metrics to assess the efficiency of the proposed model in a multi-cloud environment. The analysis includes encryption time, which quantifies the computational overhead required to secure data before storage, and upload time, which measures the duration needed to transfer encrypted data and keys to cloud storage providers. Finally, blockchain overhead is examined to assess the impact of policy validation and access logging on transaction processing time.

As illustrated in Figure 9, the encryption time analysis shows that data encryption increases steadily with data size, as expected, owing to the computational demand for larger files. In contrast, key encryption remains consistently low and stable because it involves a fixed-size symmetric key regardless of the data size. This demonstrates the efficiency of RSA for key management in this scenario.

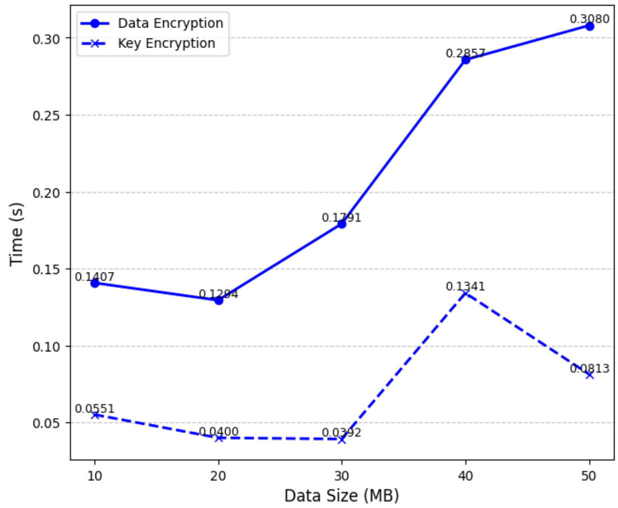


**Figure 9.** Encryption time per data and keys size

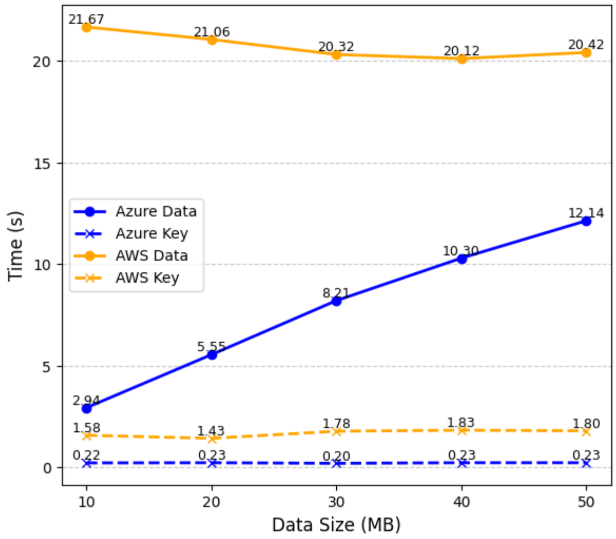


**Figure 10.** Data and keys upload time per CSP

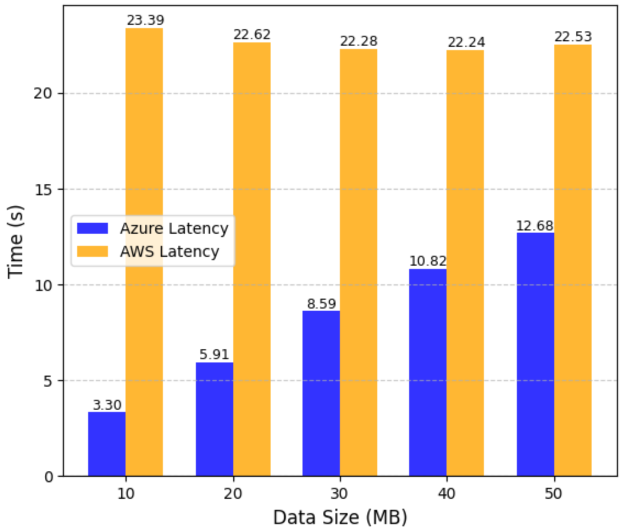


**Figure 11.** Latency per CSP

Second, regarding the upload performance shown in Figure 10, Azure outperformed AWS in terms of upload times for both data and keys, with noticeably lower latency for larger data sizes.

Latency $L_{total}$ depicted in Figure 11, represents the total time for securely storing data provided by data owner in the cloud, it combines encryption time $L_{enc}$, upload time $L_{upload}$, and network latency. Network latency includes round-trip time $L_{RTT}$ and data transfer time, which are determined by data size $S$ and upload speed $B$.

$$L_{total} = L_{enc} + L_{upload} + (L_{RTT} + \frac{S}{B}) \tag{11}$$

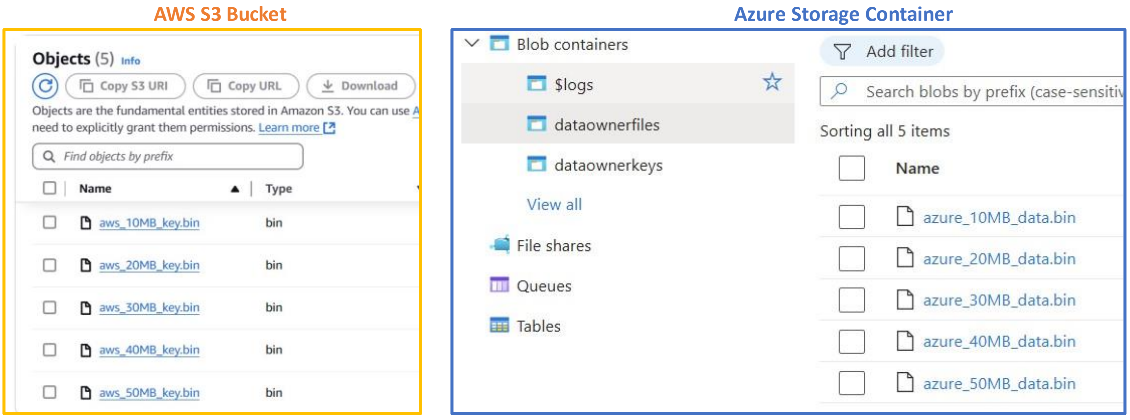
It is significantly lower for Azure compared to AWS across all data sizes. Azure latency increased moderately with data size, whereas AWS demonstrated consistently higher latency with minimal variation.

Figure 12 illustrates how data is distributed across different Cloud Service Providers (CSPs) within the multi-cloud storage layer. The left section displays an AWS S3 Bucket, where encrypted key files are stored, while the right section shows an Azure Storage Container, containing corresponding data files.

As shown in Figure 13, CPU usage remained relatively stable, ranging from 13% to 35%, while memory usage increased gradually, reaching 0.6% of the available memory. Disk utilization showed a more noticeable rise, peaking at 16.2%. These results were observed while processing multiple concurrent access requests, with the system handling five simultaneous transactions on Ganache's local blockchain environment. The experiment was designed to evaluate how resource usage changes as transaction volume increases, providing insight into the system's scalability and performance under load.

Cost: Following the deployment of our system in Azure, a cost analysis was conducted using the official Azure pricing calculator. The multi-cloud storage layer was designed by integrating Microsoft Azure and Amazon AWS for both data and key storage. As shown in Table 5, the design remains cost-effective, with a total monthly expense of 91.17$. This demonstrates that the proposed design offers a practical balance between performance and cost.

The cost estimated in Table 5 assumes a fixed resource usage and doesn't account for dynamic scaling, which can affect real-world costs. Cloud services like Azure Functions and AWS Lambda adjust pricing based on usage, meaning costs can rise with increased demand. Similarly, Azure Virtual Machines may incur extra charges when scaling to handle peak loads. Storage services like AWS S3 and Azure Blob can also become more expensive with higher access requests. Considering these factors in a dynamic cost model would provide a more realistic estimate.



AWS S3 Bucket

Objects (5) Info

Objects are the fundamental entities stored in Amazon S3. You can use A
need to explicitly grant them permissions. Learn more

Find objects by prefix

| □ | Name ▲ | Type |
|---|---|---|
| □ | aws_10MB_key.bin | bin |
| □ | aws_20MB_key.bin | bin |
| □ | aws_30MB_key.bin | bin |
| □ | aws_40MB_key.bin | bin |
| □ | aws_50MB_key.bin | bin |

Azure Storage Container

Search blobs by prefix (case-sensitiv

Sorting all 5 items

| □ | Name |
|---|---|
| □ | azure_10MB_data.bin |
| □ | azure_20MB_data.bin |
| □ | azure_30MB_data.bin |
| □ | azure_40MB_data.bin |
| □ | azure_50MB_data.bin |

**Figure 12.** Persistence of encrypted data and keys

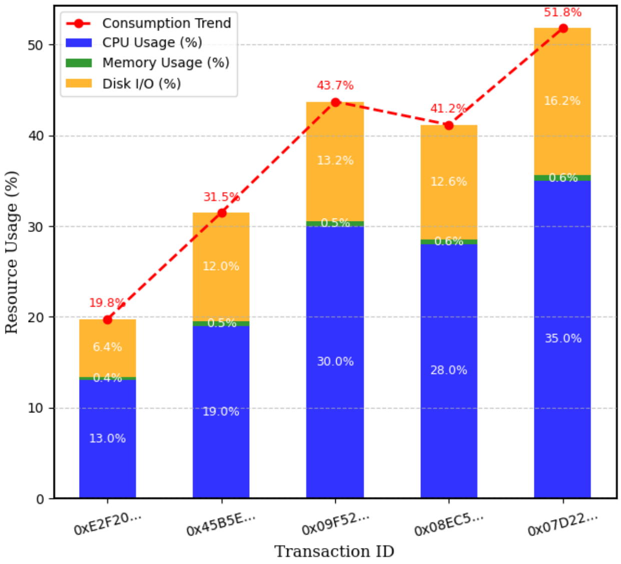


**Figure 13.** Resource consumption by Ganache process

**Table 5.** Model cost estimation

| Layer | Model Component | Cloud Service | Configuration | Cost (USD) |
|---|---|---|---|---|
| **Data Gateway** | Rest Call Interceptor | Azure API Management | Region: Central US, Tier: Consumption. | First 1 million calls **free** |
| | CSP Selector | Azure Functions | Region: Central US, Tier: Consumption, Memory size: 128 Mb, Execution time: 1000 ms, Executions per month: 2000. | First 1,000,000 executions are **free**. |
| | Data Handler | Azure Functions | | |
| **Policy Validation** | Access Module | Azure Functions | | |
| | Blockchain Infrastructure | Azure Virtual Machine | Region: Central USA East, OS: Windows, Tier: BasicStandard, Category: Compute Optimized, Instance: 1 vCPUs, 2 GB RAM, 16 GB Temporary storage, $0.105/hour. | **76.95** |
| | Smart Contract | Azure Functions | Region: Central US, Tier: Consumption, Memory size: 128 Mb, Execution time: 1000 ms, Executions per month: 2000. | First 1,000,000 executions are **free** |
| **Multi Cloud Storage** | CSP1 / Data | Azure Blob | Region: Central US, Access tier:Hot, Redundancy:LRS, Capacity: 100 Gb. | **1.84** |
| | CSP1 / Secret Key | Azure Key Vault | Region: Central US, Operations: 10000. | **0.030** |
| | CSP2 / Data | AWS S3 Object | Region: USA East, Tier: Standard, Operations: 10000. | **2.35** |
| | CSP2 / Secret Key | AWS KMS Secret | Region: USA East, Managed CMK: 10, Symmetric requests: 1000. | **10.00** |
| | | **TOTAL PER MONTH:** | | **91.17** |

### 5.2 Findings

This evaluation highlights the efficiency of the proposed model. Data encryption times increased proportionally with data size, as expected, while RSA-based key encryption maintained consistently low times due to the fixed key size. Azure demonstrated better upload performance than AWS, particularly for larger data sizes, with lower latency. The access validation module exhibited stable performance, with validation times between 2.11 and 2.33 seconds. Resource usage was efficient, increasing logically with transactions. Although integrating Azure and AWS in the multi-cloud storage layer incurs slightly higher costs, the model achieves an optimal balance between performance, cost, and scalability.

The proposed approach has several important advantages over existing models. It reduces the computational load and delays seen in the models [23-27] and makes key management easier. It also ensures scalability in multi-cloud environments, solving the issues highlighted in previous study [26-29]. Additionally, the system is designed to take full advantage of the scalability offered by cloud platforms, allowing its components to be deployed as cloud-native services using a pay-as-you-go pricing model. Its transparent design makes it easy to deploy using an Infrastructure as Code (IaC) approach.

## 6. CONCLUSION AND PERSPECTIVES

This paper outlined a private blockchain-based model that improves the security and transparency of data access in the multi-cloud storage context. The integration of decentralized access control and an immutable audit trail ensures that data is accessed only by authorized users while maintaining accountability through secure and tamper-proof logging. The experimental findings validate the model's capability to deliver robust encryption, reliable access control, and efficient data management within the multi-cloud data storage context.

Although the proposed model demonstrates strong potential, it faces challenges related to scalability and latency under high-demand conditions. As transaction volume increases, the execution time of smart contracts and blockchain validation overhead introduce processing bottlenecks, affecting system response times. Furthermore, high network traffic and inefficient resource allocation can contribute to delays in access request processing, particularly in scenarios with high concurrency.

Future work will focus on optimizing smart contract algorithms to improve access validation efficiency in dynamic scenarios. Additionally, exploring homomorphic encryption could provide enhanced security for data in its "in-use" state, allowing secure operations without exposing sensitive information. These advancements aim to address the existing limitations and improve the scalability and adaptability of the framework in diverse cloud applications.

## ACKNOWLEDGMENT

The C3S Research Laboratory generously supported this research. We sincerely thank everyone whose ideas, concepts, and methodologies, shared through academic articles and publications, have inspired and enriched our work.

## REFERENCES

[1] Idrus, M.A.Z.B., Rahman, F.D.A., Khalifa, O.O., Yusoff, N.M. (2023). Blockchain-based security for cloud data storage. In 2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Kuala Lumpur, Malaysia, pp. 73-77. https://doi.org/10.1109/ICSIMA59853.2023.10373457

[2] Ploysuayngam, W., Tangwannawit, S. (2022). Investigating the determinants of cloud storage services adoption in higher education. In 2022 6th International Conference on Information Technology (InCIT), Nonthaburi, Thailand, pp. 245-250. https://doi.org/10.1109/InCIT56086.2022.10067328

[3] Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., Ayaz, M. (2021). A systematic literature review on cloud computing security: Threats and mitigation strategies. IEEE Access, 9: 57792-57807. https://doi.org/10.1109/ACCESS.2021.3073203

[4] Joshi, A., Raturi, A., Kumar, S., Dumka, A., Singh, D.P. (2022). Improved security and privacy in cloud data security and privacy: Measures and attacks. In 2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP), Uttarakhand, India, pp. 230-233. https://doi.org/10.1109/ICFIRTP56122.2022.10063186

[5] Wan, J., Hu, K., Li, J., Su, H., Wu, Q., Li, M., Feng, L., Pan, Y. (2023). Smart contract service optimization in blockchain-cloud collaborative computing. In 2023 24th IEEE International Conference on Mobile Data Management (MDM), Singapore, Singapore, pp. 280-285. https://doi.org/10.1109/MDM58254.2023.00052

[6] Jondhale, S.D., Korde, S.K., Rakshe, D.S., Krishna, I.M., Akram, S.V., Verma, D. (2022). Blockchain in cloud computing: Design challenges. In 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 765-773. https://doi.org/10.1109/SMART55829.2022.10047424

[7] Deng, H., Fang, F., Chen, J., Zhang, Y. (2021). A cloud data storage technology for alliance blockchain technology. In 2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), NY, USA, pp. 174-179. https://doi.org/10.1109/BigDataSecurityHPSCIDS52275.2021.00041

[8] Bhari, S., Quraishi, S.J. (2022). Blockchain and cloud computing-A review. In 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, pp. 766-770. https://doi.org/10.1109/COM-IT-CON54601.2022.9850499

[9] Shaikh, R. (2022). Blockchain based cloud storage of patients health records. In 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, pp. 1-5. https://doi.org/10.1109/DELCON54057.2022.9753574

[10] Yang, F., Lei, L., Zhu, H. (2022). Overview of blockchain and cloud service integration. In 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS),

Jinan, China, pp. 35-40. https://doi.org/10.1109/BigDataSecurityHPSCIDS54978.2022.00017

[11] Bundela, R., Dhanda, N., Verma, R. (2024). Bridging blockchain with cloud computing: A blockchain-as-a-service (BaaS) approach. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, pp. 1-8. https://doi.org/10.1109/ICCCNT61001.2024.10724710

[12] Devi, S., Bharti, T.S. (2021). Study of architecture and issues in services of cloud computing. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, pp. 1578-1581. https://doi.org/10.1109/ICAC3N53548.2021.9725679

[13] Chavan, J., Patil, R., Patil, S., Gutte, V., Karande, S. (2022). A survey on security threats in cloud computing service models. In 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 574-580. https://doi.org/10.1109/ICICCS53718.2022.9788148

[14] Kaushik, P., Rao, A.M., Singh, D.P., Vashisht, S., Gupta, S. (2021). Cloud computing and comparison based on service and performance between Amazon AWS, Microsoft Azure, and Google Cloud. In 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, pp. 268-273. https://doi.org/10.1109/ICTAI53825.2021.9673425

[15] Sisodia, S., Sharma, S., Kumar, D., Biswas, L., Aluvala, S. (2024). Navigating the cloud: Choosing the right cloud computing services for your business. In 2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC), Tandojam, Pakistan, pp. 1-8. https://doi.org/10.1109/KHI-HTC60760.2024.10481894

[16] El Moudni, M., Ziyati, E. (2023). A multi-cloud and zero-trust based approach for secure and redundant data storage. In 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), Istanbul, Turkiye, pp. 1-6. https://doi.org/10.1109/WINCOM59760.2023.10323009

[17] Sahu, P., Roy, S., Gharote, M., Mondal, S., Lodha, S. (2022). Cloud storage and processing service selection considering tiered pricing and data regulations. In 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), Vancouver, WA, USA, pp. 92-101. https://doi.org/10.1109/UCC56403.2022.00020

[18] Inakollu, A., Kranthi, S., Jashua, A. (2024). A novel approach to data security in cloud storage using erasure coding and re-encryption. In 2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), Kirtipur, Nepal, pp. 972-976. https://doi.org/10.1109/I-SMAC61858.2024.10714746

[19] El Moudni, M., Ziyati, E. (2024). Securing sensitive data in multi-cloud storage using ML and homomorphic encryption. In 2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA), Balikpapan, Indonesia, pp. 754-759. https://doi.org/10.1109/ICSINTESA62455.2024.10748230

[20] Dixit, R.R., Gokulapriya, R. (2023). A comprehensive investigation of blockchain technology's role in cyber security. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 386-390. https://doi.org/10.1109/ICACCS57279.2023.10113026

[21] Wang, X., Jia, J., Cao, Y., Du, J., Hu, A., Liu, Y., Wang, Z. (2023). Application of data storage management system in blockchain-based technology. In 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, pp. 1437-1440. https://doi.org/10.1109/EEBDA56825.2023.10090564

[22] Wang, L., Ding, W., Yan, Z., Qiu, S., Wang, M., Wan, Z. (2023). EDDAC: An efficient and decentralized data access control scheme with attribute privacy preservation. IEEE Internet of Things Journal, 11(8): 14579-14592. https://doi.org/10.1109/JIOT.2023.3342314

[23] Zhang, Y., Xiong, L., Li, F., Hao, Y., Liu, Z. (2024). Blockchain-based privacy-preserving authentication with hierarchical access control using polynomial commitment for mobile cloud computing. IEEE Internet of Things Journal, 11(10): 18266-18280. https://doi.org/10.1109/JIOT.2024.3361506

[24] Cheng, H., Lo, S.L., Lu, J. (2024). A blockchain-enabled decentralized access control scheme using multi-authority attribute-based encryption for edge-assisted Internet of Things. Internet of Things, 26: 101220. https://doi.org/10.1016/j.iot.2024.101220

[25] Yang, D., Tsai, W.T. (2024). An optimized encryption storage scheme for blockchain data based on cold and hot blocks and threshold secret sharing. Entropy, 26(8): 690. https://doi.org/10.3390/e26080690

[26] Kanakasabapathi, R.S., Judith, J.E. (2024). Enhancing cloud storage security through blockchain-integrated access control and optimized cryptographic techniques. International Journal of Advanced Technology and Engineering Exploration, 11(117): 1183. https://doi.org/10.19101/IJATEE.2023.10101660

[27] Agrawal, R., Singhal, S., Sharma, A. (2024). Blockchain and fog computing model for secure data access control mechanisms for distributed data storage and authentication using hybrid encryption algorithm. Cluster Computing, 27(6): 8015-8030. https://doi.org/10.1007/s10586-024-04411-9

[28] Rangappa, K., Ramaswamy, A.K.B., Prasad, M., Kumar, S.A. (2024). A novel method of secured data distribution using sharding zkp and zero trust architecture in blockchain multi cloud environment. Cryptography, 8(3): 39. https://doi.org/10.3390/cryptography8030039

[29] Panda, S., Sahoo, S., Halder, R., Mondal, S. (2024). Contextual attribute-based access control scheme for cloud storage using blockchain technology. Software: Practice and Experience, 54(10): 2042-2062. https://doi.org/10.1002/spe.3250

[30] Du, J., Dong, G., Ning, J., Xu, Z., Yang, R. (2024). Blockchain-based and multi-authority hierarchical access control data sharing scheme. Computers and Electrical Engineering, 119: 109547. https://doi.org/10.1016/j.compeleceng.2024.109547

[31] Dai, Y., Wu, J., Mao, S., Rao, X., Gu, B., Qu, Y., Lu, Y. (2024). Blockchain empowered access control for digital twin system with attribute-based encryption. Future Generation Computer Systems, 160: 564-576.

https://doi.org/10.1016/j.future.2024.06.037

[32] Neela, K.L. (2024). DSDOS cloud: A decentralized secure data outsourcing system with hybrid encryption, blockchain smart contract-based access control, and hash authentication codes for cloud security. Transactions on Emerging Telecommunications Technologies, 35(11): e70016. https://doi.org/10.1002/ett.70016

[33] Liu, G., Xie, H., Wang, W., Huang, H. (2024). A secure and efficient electronic medical record data sharing scheme based on blockchain and proxy re-encryption. Journal of Cloud Computing, 13(1): 44. https://doi.org/10.1186/s13677-024-00608-w

[34] Singh, I., Singh, B. (2024). LAA-D: Lightweight authentication and access control mechanism with dual-data storage in cloud-internet of things system using blockchain. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 48(4): 1483-1499. https://doi.org/10.1007/s40998-024-00748-4

[35] Ullah, Z., Husnain, G., Mohmand, M.I., Qadir, M., Alzahrani, K.J., Ghadi, Y.Y., Alkahtani, H.K. (2024). Blockchain-IoT: A revolutionary model for secure data storage and fine-grained access control in internet of things. IET Communications, 18(19): 1524-1540. https://doi.org/10.1049/cmu2.12845

[36] Roy, U., Ghosh, N. (2024). BloAC: A blockchain-based secure access control management for the Internet of Things. Journal of Information Security and Applications, 87: 103897. https://doi.org/10.1016/j.jisa.2024.103897

[37] Das, S., Mishra, M., Priyadarshini, R., Barik, R.K., Saikia, M.J. (2024). A secure, privacy-preserving, and cost-efficient decentralized cloud storage framework using blockchain. Journal of King Saud University-Computer and Information Sciences, 36(10): 102260. https://doi.org/10.1016/j.jksuci.2024.102260

[38] Singh, A., Rathee, G. (2025). Smart contract empowered dynamic consent: Decentralized storage and access control for healthcare applications. Peer-to-Peer Networking and Applications, 18(1): 1-16. https://doi.org/10.1007/s12083-024-01827-3

[39] Rajkumar, V., Devi, D., Khadirkumar, N., Jeevitha, P., Suganya, T. (2024). Enhancing access control and information sharing in cloud IoT with an effective blockchain-based authority system. In International Conference on Applications and Techniques in Information Security, pp. 288-299. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-97-9743-1_21

[40] Bhatt, S., Pham, T.K., Gupta, M., Benson, J., Park, J., Sandhu, R. (2021). Attribute-based access control for AWS internet of things and secure industries of the future. IEEE Access, 9: 107200-107223. https://doi.org/10.1109/ACCESS.2021.3101218