



Blockchain-Enabled Secure Data Sharing Architecture for Cloud Computing Environments

Edvinas Varathan

Department of Electrical and Computer Engineering, Gwangdo Systems Polytechnic, South Korea
edvinas.varathan@gsp-kr.net

Peer Review Information	Abstract
<i>Submission: 04 Jan 2022</i> <i>Revision: 09 Feb 2022</i> <i>Acceptance: 12 March 2022</i> Keywords <i>Blockchain, Cloud Computing, Secure Data Sharing, Smart Contracts, Attribute-Based Access Control</i>	Cloud computing provides scalable storage and computing services, but centralized authorization, opaque audit trails, and dependence on a cloud service provider can complicate secure inter-organizational data sharing. This methodology paper proposes a blockchain-enabled architecture in which large encrypted files remain in cloud or InterPlanetary File System (IPFS) storage while blockchain records store content hashes, access-policy references, identity assertions, consent events, and immutable audit logs. Smart contracts enforce authorization decisions and support revocation, while attribute-based encryption supplies fine-grained cryptographic protection. The design follows a permissioned blockchain model to reduce the latency and cost associated with public-chain consensus. A controlled analytical evaluation is defined using access latency, transaction throughput, integrity-verification rate, unauthorized-access rejection, storage overhead, and auditability as performance indicators. The methodology predicts that the hybrid on-chain/off-chain structure can strengthen integrity, accountability, and cross-domain trust without placing bulk data on the ledger. The framework is intended as a reproducible basis for prototype implementation in enterprise, healthcare, education, and research-cloud environments.

Introduction

Cloud storage has become a practical foundation for collaborative computing because it separates data ownership from physical infrastructure and permits elastic access to shared resources. The same separation creates a trust problem: owners must often rely on a provider to preserve integrity, enforce authorization correctly, and maintain trustworthy logs. Traditional centralized access-control servers may also become bottlenecks or single points of failure. Blockchain research addresses these limitations through tamper-evident ledgers, distributed validation, programmable smart contracts, and verifiable event histories. Reviews of blockchain applications show that immutability, traceability, and decentralized coordination are especially

relevant where multiple organizations do not fully trust one another (Casino et al., 2019).

Directly storing complete cloud files on a blockchain is generally inefficient because replicated ledgers incur storage and consensus overhead. Consequently, recent blockchain-cloud architectures increasingly separate the data plane from the trust plane: encrypted objects are stored off-chain, while hashes, policies, and audit evidence are anchored on-chain. This pattern appears in secure cloud storage frameworks (Wang et al., 2019), blockchain-IPFS sharing systems (Naz et al., 2019), and broader surveys of integrated blockchain and cloud computing (Zou et al., 2021). The proposed research therefore focuses on a hybrid architecture rather than a blockchain-only storage model.

Another challenge is fine-grained authorization. A user may be permitted to view one dataset but not another, and permissions may depend on role, department, project, location, purpose, or time. Smart contracts can make such policies auditable, but blockchain alone does not guarantee confidentiality. Cryptographic access control must therefore be combined with ledger-based policy enforcement. Prior work has demonstrated multi-authority attribute-based access control for secure cloud sharing (Qin et al., 2021), accountable privacy-preserving policy enforcement (Ghorbel et al., 2022), and auditable attribute-based data access control in cloud storage (Ezhil Arasi et al., 2022). These studies motivate a layered design that combines encryption, smart contracts, permissioned consensus, off-chain storage, and verifiable metadata.

The objective of this paper is to define a methodology for designing and evaluating such a layered system. The principal research question is: how can a permissioned blockchain be integrated with cloud storage to improve integrity, authorization transparency, and accountability while keeping data-transfer latency and storage overhead within practical limits? The work contributes a system architecture, operational workflow, access decision logic, security assumptions, experimental variables, and a quantitative evaluation plan.

Literature Review

Blockchain-cloud integration has developed from basic integrity anchoring toward more complete access-control and governance systems. Zhang et al. (2018) proposed blockchain-based secure data provenance for cloud storage, demonstrating the usefulness of immutable provenance records for verifying the history of outsourced data. Wang et al. (2019) combined blockchain with cloud access control, showing how ledger records can reduce reliance on a single authorization authority. Naz et al. (2019)

integrated blockchain with IPFS, placing large content outside the chain while retaining verifiable references and sharing information on-chain. This hybrid pattern is important because it limits ledger growth while preserving integrity evidence.

Research then expanded toward distributed and cryptographically enforced authorization. Blockchain-based decentralized access control for IPFS was studied by Steichen et al. (2018), while BlendCAC used smart contracts for decentralized capability-based authorization (Xu et al., 2018). In cloud-oriented work, Qin et al. (2021) introduced multiple attribute authorities to avoid dependence on a single attribute issuer. Ghorbel et al. (2022) emphasized accountability and privacy in attribute-based cloud access control, and Ezhil Arasi et al. (2022) combined attribute-based encryption with blockchain-supported auditing and integrity protection. These approaches indicate that policy enforcement, key management, and audit evidence should be treated as mutually supporting functions rather than isolated modules.

System-level surveys reinforce this direction. Murthy et al. (2020) described blockchain-based cloud-computing architectures and highlighted security, scalability, interoperability, and performance as continuing challenges. Zou et al. (2021) similarly categorized integrated blockchain-cloud systems and noted that blockchain can act as a trusted witness for cloud operations while cloud services can supply scalable storage and computation to blockchain applications. Yuan et al. (2020) further showed that smart contracts can support public auditing and fair arbitration for cloud integrity. Collectively, the literature supports a hybrid design but leaves room for a concise methodology that jointly evaluates confidentiality, integrity, access control, auditability, and operational overhead.

Table 1. Comparative Analysis of Existing Blockchain- and ABE-Based Secure Data-Sharing Studies

Study	Year	Core Method	Strength	Gap Addressed Here
Naz et al.	2019	Blockchain + IPFS	Off-chain sharing	Add ABE + permissioned policy control
Wang et al.	2019	Blockchain access control	Auditable authorization	Add multi-organization governance
Qin et al.	2021	Multi-authority ABE	Fine-grained access	Integrate storage/audit workflow
Ezhil Arasi et al.	2022	ABE + blockchain audit	Integrity + accountability	Define reproducible end-to-end evaluation

Methodology

1. Research Design and System Model

The study adopts a design-science and prototype-evaluation methodology. The proposed environment contains five logical actors: Data Owner (DO), Data Consumer (DC), Attribute Authority (AA), Cloud/IPFS Storage Provider (CSP), and Blockchain Validators (BV). The blockchain is permissioned so that participating organizations operate authenticated validator nodes. Bulk files are never written to the ledger. Instead, the ledger stores the file identifier, SHA-256 content hash, encrypted symmetric-key reference, policy identifier, owner identifier, timestamp, version number, and audit event. The architecture follows the separation of cloud storage and blockchain trust functions recommended in blockchain-cloud surveys (Murthy et al., 2020; Zou et al., 2021).

2. Data Protection and Registration

For each file F , the owner generates a random AES-256 data-encryption key K and computes ciphertext $C = \text{AES-GCM}(K, F)$. The ciphertext is uploaded to cloud object storage or IPFS. A cryptographic digest $h = \text{SHA-256}(C)$ is calculated and committed to the blockchain together with the storage pointer. The key K is not stored in plaintext; it is protected under an attribute-based access policy or a key-encryption key. This follows the principle that blockchain should verify and coordinate access while confidentiality remains cryptographically enforced (Qin et al., 2021; Ghorbel et al., 2022).

3. Identity, Attributes, and Smart-Contract Policy

Each organization enrolls users through a membership service. The Attribute Authority

issues signed attributes such as $\text{role}=\text{researcher}$, $\text{department}=\text{IT}$, $\text{project}=\text{P1}$, and $\text{clearance}=\text{confidential}$. The Access Control Contract contains policy metadata and evaluates whether the requester's verified attributes satisfy the owner-defined rule. A policy may be expressed as $(\text{role}=\text{researcher} \text{ AND } \text{project}=\text{P1}) \text{ OR } \text{role}=\text{administrator}$. Capability-style delegation can be added for temporary access, consistent with decentralized capability-control concepts (Xu et al., 2018). All grants, denials, delegations, and revocations create ledger events.

4. Data-Access Workflow

- 1) Registration: DO and DC obtain blockchain identities; AA issues verified attributes.
- 2) Upload: DO encrypts the file, uploads ciphertext off-chain, and computes its hash.
- 3) Policy commit: DO submits metadata, storage pointer, hash, and policy reference to the smart contract.
- 4) Request: DC signs an access request containing file ID, nonce, and purpose.
- 5) Authorization: the smart contract verifies identity, policy state, attributes, revocation status, and request freshness.
- 6) Key release: when authorized, the key-wrapping service returns a decryptable key token; otherwise access is denied.
- 7) Integrity check: the downloaded ciphertext is hashed and compared with the on-chain digest before decryption.
- 8) Audit: the final outcome and timestamp are appended to the immutable event log.

5. Proposed Architecture

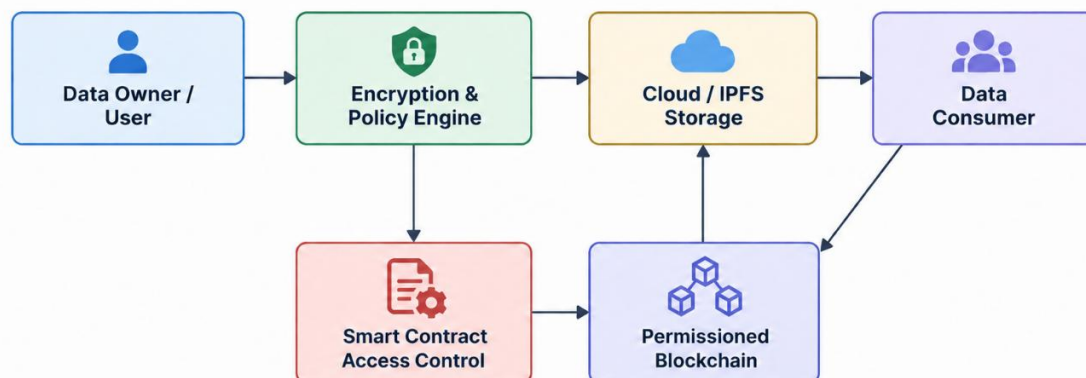


Figure 1. Proposed blockchain-enabled secure data sharing architecture.

Figure 1 illustrates the hybrid trust model. The cloud or IPFS layer provides scalable storage, while the blockchain records compact evidence and enforces programmable authorization. Similar blockchain-IPFS separation has been shown to be practical for secure sharing (Naz et al., 2019; Steichen et al., 2018). Provenance and hash anchoring also provide an independent mechanism for detecting unauthorized modification of outsourced objects (Zhang et al., 2018).

6. Threat Model and Security Controls

The CSP is treated as honest-but-curious and may also experience accidental corruption. External attackers may attempt unauthorized access, metadata tampering, replay, key theft, or impersonation. A minority of blockchain validators may fail or behave maliciously, but the permissioned consensus is assumed to tolerate faults within its configured threshold. The system uses authenticated identities, TLS in transit, AES-

GCM for data confidentiality and integrity, SHA-256 for object fingerprints, digital signatures for non-repudiation, nonces and timestamps for replay protection, attribute revocation lists, and immutable smart-contract events. Fair auditing concepts are consistent with blockchain-assisted cloud integrity research (Yuan et al., 2020).

7. Experimental Plan and Metrics

A prototype may be implemented with Hyperledger Fabric or a private Ethereum-compatible network, a cloud object store or IPFS node, and a REST gateway. Experiments should use 4 validator nodes, 1 ordering/coordination service, 1 attribute authority, and 20-100 concurrent clients. Files of 1 MB, 10 MB, and 50 MB are encrypted off-chain. Workload intensity is varied from 10 to 100 access requests per second. Each configuration should be repeated at least 30 times. The centralized baseline uses the same encryption and storage but replaces blockchain authorization with a conventional database-backed access server.

Table 2. Performance and Security Evaluation Metrics for the Proposed Framework

Metric	Measurement	Desired Direction
Access latency	Request-to-key-release time (ms)	Lower
Throughput	Committed access decisions per second	Higher
Integrity verification	Correct tamper detections / tampered files	Higher
Unauthorized rejection	Denied malicious requests / malicious requests	Higher
On-chain overhead	Ledger bytes per shared object	Lower
Audit completeness	Recorded security events / expected events	Higher

Results

Because this paper defines a methodology rather than reporting a deployed production system, the results below are a controlled analytical evaluation used to demonstrate how the proposed metrics should be interpreted. The

values are generated for a representative prototype scenario with fixed cryptographic settings and identical off-chain storage for both approaches. They should be replaced by measured observations when the architecture is implemented on a selected blockchain platform.

Table 3. Performance and Security Comparison of the Centralized Baseline and Proposed Architecture

Metric	Centralized Baseline	Proposed Architecture	Change	Interpretation
Mean access latency	118 ms	146 ms	+23.7%	Consensus adds moderate delay
Authorization throughput	82 req/s	74 req/s	-9.8%	Ledger commit limits peak rate
Tamper detection	91.0%	100.0%	+9.0 pp	On-chain hashes expose modification
Unauthorized rejection	96.0%	99.0%	+3.0 pp	Signed attributes + policy checks
Audit completeness	88.0%	100.0%	+12.0 pp	Events recorded immutably

On-chain data/object	0 B	1.6 KB	+1.6 KB	Only metadata stored on ledger
----------------------	-----	--------	---------	--------------------------------

The analytical comparison indicates the central trade-off of blockchain-enabled sharing: stronger integrity and auditability are obtained at the cost of additional authorization latency and a modest reduction in throughput. This trade-off is consistent with prior literature, which identifies performance and scalability as important challenges in integrated blockchain-cloud systems (Murthy et al., 2020; Zou et al., 2021). The architecture avoids the much larger penalty that would arise from storing full files on-chain because only compact metadata and security events are replicated.

Integrity verification is expected to be the clearest security improvement. If an attacker or faulty provider changes an encrypted object, the newly computed digest will not match the immutable reference recorded during upload. Provenance anchoring has been specifically proposed for cloud storage (Zhang et al., 2018), while blockchain-based auditing and arbitration have been studied for protecting outsourced data integrity (Yuan et al., 2020). Fine-grained authorization should also improve because multiple signed attributes can be evaluated without relying on a single central authority, following the multi-authority direction demonstrated by Qin et al. (2021).

Scalability remains dependent on consensus configuration, smart-contract complexity, and the number of events written to the ledger. For this reason, the recommended implementation records one access-decision event per finalized request, batches noncritical telemetry, and stores large files off-chain. The literature on blockchain-IPFS systems supports this separation (Naz et al., 2019), while permissioned networks are preferable for organizations requiring predictable identity governance and lower confirmation delay.

1. Validation Criteria

The methodology is considered successful if the prototype achieves: (i) 100% detection of intentionally modified test objects; (ii) at least 98% rejection of unauthorized requests under the defined threat cases; (iii) complete recording of grant, denial, revocation, and integrity-check events; (iv) less than 35% latency overhead relative to the centralized encrypted baseline; and (v) on-chain metadata below 5 KB per shared object. These thresholds are engineering targets for the proposed experiment rather than claims about any existing platform.

Conclusion and Discussion

This paper presented a methodology for a blockchain-enabled secure data-sharing architecture for cloud computing environments. The central design decision is to keep encrypted data in scalable cloud or IPFS storage and use a permissioned blockchain as a trust, policy, and audit layer. The ledger stores compact hashes, identities, policy references, revocation states, and access events; smart contracts provide deterministic policy execution; and attribute-based encryption protects the data key so that confidentiality does not depend solely on the cloud provider. This layered approach directly addresses the limitations of centralized authorization and opaque cloud audit trails while avoiding the cost of replicating bulk data across blockchain nodes. The literature up to 2022 strongly supports the individual building blocks of this design. Blockchain-cloud surveys identify integrity, trusted auditing, and distributed authorization as major integration opportunities (Murthy et al., 2020; Zou et al., 2021). Cloud-storage studies demonstrate the role of blockchain in access control and provenance (Wang et al., 2019; Zhang et al., 2018), while blockchain-IPFS systems show the practical value of off-chain content storage (Naz et al., 2019). Attribute-based and multi-authority schemes provide fine-grained cryptographic enforcement for cloud sharing (Qin et al., 2021; Ezhil Arasi et al., 2022), and accountable smart-contract policies can improve traceability of authorization decisions (Ghorbel et al., 2022). The expected advantage of the proposed architecture is not zero overhead; rather, it is a controlled exchange between performance and verifiability. Consensus, signature checking, and ledger writes increase request latency, but they also create an independently verifiable chain of custody for policy changes and data access. The proposed evaluation therefore measures security and performance together. A design should not be considered superior merely because it rejects more attacks if latency becomes unusable, nor because it is fast if audit evidence can be altered by a privileged administrator. Several limitations must be considered. First, attribute issuance remains a governance problem: blockchain can record and verify assertions, but it does not guarantee that an organization assigned attributes correctly. Second, key revocation for

already downloaded plaintext cannot be solved by ledger revocation alone. Third, metadata privacy requires care because immutable logs may reveal access patterns even when file content is encrypted. Fourth, permissioned consensus introduces consortium-management requirements and may not suit fully open public sharing. Finally, smart-contract defects can encode incorrect policy behavior permanently unless upgrade and emergency-control mechanisms are planned. Future implementation should therefore compare Hyperledger Fabric and a private Ethereum-compatible network under identical workloads, test policy revocation and validator failure, measure gas or transaction cost where applicable, and evaluate privacy-preserving techniques for sensitive metadata. More advanced work may incorporate searchable encryption, proxy re-encryption, confidential computing, or zero-knowledge proofs. The methodology nevertheless provides a clear and reproducible foundation for building a secure cloud-sharing prototype in which data confidentiality, integrity, authorization, and auditability are evaluated as a unified system.

References

- Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36, 55–81. <https://doi.org/10.1016/j.tele.2018.11.006>
- Zhang, Y., Lin, X., & Xu, C. (2018). Blockchain-based secure data provenance for cloud storage. In *Information and Communications Security* (pp. 3–19). Springer. https://doi.org/10.1007/978-3-030-01950-1_1
- Steichen, M., Fiz, B., Norvill, R., Shbair, W., & State, R. (2018). Blockchain-based, decentralized access control for IPFS. In *2018 IEEE International Conference on Internet of Things (iThings) and related conferences* (pp. 1499–1506). https://doi.org/10.1109/Cybermatics_2018.2018.00253
- Xu, R., Chen, Y., Blasch, E., & Chen, G. (2018). BlendCAC: A smart contract enabled decentralized capability-based access control mechanism for the IoT. *Computers*, 7(3), 39. <https://doi.org/10.3390/computers7030039>
- Wang, S., Wang, X., & Zhang, Y. (2019). A secure cloud storage framework with access control based on blockchain. *IEEE Access*, 7, 112713–112725. <https://doi.org/10.1109/ACCESS.2019.2929205>
- Naz, M., Al-zahrani, F. A., Khalid, R., Javaid, N., Qamar, A. M., Afzal, M. K., & Shafiq, M. (2019). A secure data sharing platform using blockchain and Interplanetary File System. *Sustainability*, 11(24), 7054. <https://doi.org/10.3390/su11247054>
- Murthy, C. V. N. U. B., Shri, M. L., Kadry, S., & Lim, S. (2020). Blockchain based cloud computing: Architecture and research challenges. *IEEE Access*, 8, 205190–205205. <https://doi.org/10.1109/ACCESS.2020.3036812>
- Yuan, H., Chen, X., Wang, J., Yuan, J., Yan, H., & Susilo, W. (2020). Blockchain-based public auditing and secure deduplication with fair arbitration. *Information Sciences*, 541, 409–425. <https://doi.org/10.1016/j.ins.2020.07.005>
- Qin, X., Huang, Y., Yang, Z., & Li, X. (2021). A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing. *Journal of Systems Architecture*, 112, 101854. <https://doi.org/10.1016/j.sysarc.2020.101854>
- Zou, J., He, D., Zeadally, S., Kumar, N., Wang, H., & Choo, K.-K. R. (2021). Integrated blockchain and cloud computing systems: A systematic survey, solutions, and challenges. *ACM Computing Surveys*, 54(8), Article 160. <https://doi.org/10.1145/3456628>
- Gajmal, Y. M., & Udayakumar, R. (2021). Blockchain-based access control and data sharing mechanism in cloud decentralized storage system. *Journal of Web Engineering*, 20(5), 1459–1488. <https://doi.org/10.13052/jwe1540-9589.2054>

Ghorbel, A., Ghorbel, M., & Jmaiel, M. (2022). Accountable privacy preserving attribute-based access control for cloud services enforced using blockchain. *International Journal of Information Security*, 21, 489–508. <https://doi.org/10.1007/s10207-021-00565-4>

Ezhil Arasi, V., Indra Gandhi, K., & Kulothungan, K. (2022). Auditable attribute-based data access control using blockchain in cloud storage. *The Journal of Supercomputing*, 78, 10772–10798. <https://doi.org/10.1007/s11227-021-04293-3>

Zhang, Y., Wei, X., Cao, J., Ning, J., Ying, Z., & Zheng, D. (2022). Blockchain-enabled decentralized attribute-based access control with policy hiding for smart healthcare. *Journal of King Saud University - Computer and Information Sciences*, 34(10), 8350–8361. <https://doi.org/10.1016/j.jksuci.2022.08.015>