ISSN: 2393-8048
International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Multilingual, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =8.152, July-December 2025, Submitted in December 2025

Blockchain - Integrated Zero-Trust Architectures for Secure Cloud Computing

Arti, Research Scholar, Dept. of Computer Science, S.K.D. University, Hanumangarh
Dr. Geetanjali, Research Supervisor, Dept. of Computer Science, S.K.D. University, Hanumangarh

Abstract

Cloud computing has emerged as the backbone of modern digital transformation, but the dynamic and distributed nature of cloud environments exposes them to insider abuse, lateral movement, and opaque auditability. Zero-Trust Architecture (ZTA)—"never trust, always verify"—addresses these issues by continuously authenticating and authorizing every request. This paper presents a blockchain-integrated ZTA that decentralizes trust brokering, anchors verifiable identity, policy state, and tamper-evident audit logs on a permissioned ledger, and automates continuous verification via smart contracts. We develop a three-layer framework (identity, access control, auditing), formalize core algorithms for on-chain verification and short-lived capability issuance, and provide analytic results on fault tolerance, latency, and throughput. Conceptual case studies (healthcare and financial cloud) indicate strengthened resistance to insider threats, fine-grained revocation, verifiable compliance, and reduced single-point-of-failure risk. Limits include scalability/latency trade-offs, interoperability, and governance. We outline migration guidance and research avenues (hybrid chains, privacy-preserving proofs, AI-assisted policy).

Keywords: Blockchain, Zero-Trust Architecture, Cloud Security, Access Control, Decentralized Identity, Auditability

1. Introduction

Cloud platforms underpin healthcare, finance, and public services, yet perimeter models implicitly trust "inside" entities (Rose et al., 2020). ZTA replaces implicit trust with continuous, context-aware verification (Kindervag, 2010). Blockchain contributes decentralized consensus, immutability, and auditability (Zhang & Jacobsen, 2018), which can externalize trust from any single provider component. We investigate how a permissioned blockchain can implement and harden ZTA principles in multi-tenant clouds. Perimeter defenses struggle with multi-cloud, BYOD, and microservices (Rose et al., 2020). ZTA enforces least privilege and continuous evaluation. Blockchain provides append-only, verifiable state and multi-party attestation (Yli-Huumo et al., 2016). Prior works explore ZTA guidelines (Rose et al., 2020) and blockchain for access control or identity (Moin et al., 2019; Mollah et al., 2021), but integrated, system-level designs remain under-evaluated at cloud scale. Centralized policy engines and identity providers can be single points of failure and high-value targets. Permissioned blockchains (e.g., Fabric) can federate trust among operators, recording policy, credentials, revocations, and audits with cryptographic integrity while keeping sensitive payloads off-chain. This satisfies regulatory needs for tamper-evident records (Mollah et al., 2021). Authoritative guidance frames ZTA as identity-, device-, and resource-centric with continuous verification at every request boundary. NIST SP 800-207 formalizes core components (PDP/PEP separation, least-privilege, continuous evaluation) and patterns for microservice and multi-cloud contexts (Rose et al., 2020). Industry implementations such as Google's BeyondCorp demonstrate how removing the "trusted intranet" in favor of device posture and user identity hardens access paths against lateral movement and credential replay (Kindervag, 2010; Zaharia et al., 2016, for the data-plane implications in distributed systems). Attribute-Based Access Control (ABAC) supports fine-grained, context-aware policy evaluation suitable for ZTA (NIST SP 800-162). Role-Based Access Control (RBAC) remains prevalent in cloud operators but is often composed with attributes for least-privilege and just-in-time elevation. In modern token ecosystems, OAuth 2.0 profiles such as UMA 2.0 enable resource-owner-managed grants, and Demonstration of Proof-of-Possession (DPoP) binds tokens to a client key to reduce bearer-token replay. For workload identity between services, SPIFFE/SPIRE issues short-lived

workload identities (SVIDs) that are rotation-friendly and cloud-agnostic, aligning well with ZTA's "always verify" principle.

Permissioned ledgers provide identity-aware, policy-governed transaction processing with pluggable consensus. Hyperledger Fabric separates endorsement from ordering and supports channels and endorsement policies for fault isolation and governance; this makes it suitable for multi-party cloud settings where several organizations must share a consistent, tamper-evident view of policy versions, revocations, and audit commitments (Androulaki et al., 2018). Practical Byzantine Fault Tolerance (PBFT) and its derivatives offer strong safety under $n \geq 3f+1n$ validators and are commonly adopted in enterprise chains (Castro & Liskov, 1999). W3C's Decentralized Identifiers (DID) and Verifiable Credentials (VC) data models enable issuers to attest attributes without central IdPs, while holders present selective proofs to verifiers. In a zero-trust cloud, on-chain registries for issuers and revocation lists allow verifiable status checks (validity, revocation) without disclosing raw attributes, supporting privacy-preserving, cross-tenant authentication (Yli-Huumo et al., 2016; Mollah et al., 2021; Kumar et al., 2022).

Studies across cloud and IoT propose smart-contract-mediated authorization, ledger-anchored revocation, and tamper-evident audit logging to remove single points of failure and strengthen insider-threat resistance. Common patterns include short-lived capabilities minted after on-chain verification, Merkle-anchored audit trails with off-chain log retention, and governed policy rotation with multi-signature approvals (Moin et al., 2019; Fernandes et al., 2022). These works consistently surface the trade-offs among throughput, confirmation latency, and governance complexity. Permissioned BFT systems achieve low-second commit latencies with batching, but high-frequency access control can stress ordering services; designs therefore amortize ledger interaction via asynchronous audit anchoring or grouped policy updates. Privacy-preserving techniques—selective disclosure in VCs, attribute-based encryption for secrets distribution, and, in some proposals, zero-knowledge proofs—reduce on-chain data exposure while maintaining auditability. For regulated sectors, append-only audit commitments with off-chain, write-once logs provide defensible compliance evidence (Mollah et al., 2021). Systematic surveys report maturing patterns for blockchain-enabled identity and access control but relatively few at-scale, multi-organization deployments with rigorous latency/throughput measurements (Fernandes et al., 2022). Open issues include interoperability across clouds, standardized credential schemas, governance of policy updates, and performance under bursty, microservice-dense workloads. These gaps motivate hybrid architectures (permissioned chains with public anchoring, asynchronous audit commits), standardized identity vocabularies, and controlled pilots in healthcare and finance.

The objective of this research is to examine the limitations of traditional cloud security and Zero Trust Architecture (ZTA), assess the promise of blockchain technology as a facilitator for ZTA, and suggest an integrated framework that embodies both approaches. It will try to plot such core applications as identity management, access control, and compliance monitoring within the same framework and identify existing limitations. In addition, it aims to define future research opportunities to tackle issues like scalability, interoperability, and privacy preservation in order to develop secure, transparent, and cost-effective cloud security solutions based on the integration of blockchain and ZTA principles.

2. Material and Methods

2.1 Research Design

We adopt a conceptual + comparative methodology grounded in ZTA design goals. We (i) derive requirements, (ii) specify a three-layer architecture with on-chain verification, (iii) define algorithms and security/performance formulas, and (iv) validate via criteria-based analysis and two domain case studies (healthcare, finance). This is appropriate where multiparty deployments and regulated data preclude easy large-scale experimentation.

2.2 Threat Model and Assumptions

- **Adversaries:** malicious insiders; compromised service accounts; lateral-movement actors; colluding administrators within a single organization.
- **Trust:** No single cloud component is fully trusted. A permissioned BFT ledger with nnn validator nodes (operated by separate orgs/tenants/regions) is assumed.
- **Byzantine fault tolerance:** The ledger tolerates up to f Byzantine faults if n≥3f+1n \ge 3f+1n≥3f+1 (Castro & Liskov, 1999).
- **Privacy:** Sensitive data remain off-chain; only hashes/commitments, pointers, and policy/credential metadata go on-chain.
- **Crypto:** ECDSA/Ed25519 for identities; SHA-256 for hashing; optional BLS aggregation for multi-party attestations

2.3 Architecture (Three Layers)

(A) Identity & Credentials

- Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) for users, devices, services; revocation lists and issuer registries anchored on-chain (W3C DID/VC).
- Optional Attribute-Based Encryption (ABE) for fine-grained secrets sharing (Sahai & Waters, 2005).

(B) Access Control & Continuous Verification

- Policy Decision Point (PDP) queries on-chain policy state and revocations; Policy Enforcement Points (PEPs) guard every microservice.
- Short-lived capability tokens (e.g., DPoP-bound JWT/macaroons) minted after on-chain verification; renewed frequently (minutes) to enforce "always verify."

(C) Auditing & Compliance

- **Append-only audit channel**: each access decision and token issuance emits an audit event hash (Merkle-linked) to the ledger; raw logs stored off-chain (e.g., WORM storage) with on-chain hash commitments for later proof.

2.4 Core Algorithms (Pseudo-code)

Algorithm 1: On-Chain-Anchored Authorization (OOZ-AA)

Input: subject S, resource R, action A, context C (device, network, time)

1. Verify DID/VC of S (issuer ∈ allowed_issuers AND revocation_status=valid).
2. Fetch current policy P* (hash h_P on-chain; policy doc off-chain by h_P).
3. Evaluate ABAC decision D = PDP.eval (P*, S, R, A, C).
4. If D=Permit:
   - 4.1 Mint short-lived capability cap = Macaroon/JWT (DPoP-bound, ttl = minutes).
   - 4.2 Emit audit_event = H(S,R,A,C,cap_id) to audit queue; anchor Merkle root on-chain.
   - 4.3 Return cap to PEP.

   Else: deny and emit audit_event (deny).

Algorithm 2: Continuous Verification on Use (CV-Use)

On each API call with token cap:

1. PEP checks signature/binding (DPoP) and expiry (ttl).
2. Query revocation cache (local + on-chain CRL); if revoked ⇒ deny.
3. Optional: re-evaluate critical attributes (device posture, geo, risk score).
4. If all pass ⇒ allow; emit audit_event; rotate/refresh cap as needed.

Algorithm 3: Policy/Key Rotation & Revocation (PKR)

1. Propose change (policy version v+1 or key set K') with multi-sig approvals.
2. Endorse per governance policy; order & commit to chain (store hash only).
3. Distribute new P*, K' to PDP/PEP; invalidate prior caps (grace ≤ TTL).
4. Anchor revocation entries; ensure caches purge within SLA.

ISSN: 2393-8048
International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Multilingual, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =8.152, July-December 2025, Submitted in December 2025

2.5 Security & Performance Formulas

- BFT tolerance: $n \geq 3f + 1 \Rightarrow$ tolerates up to $f = \lfloor (n-1)/3 \rfloor$ Byzantine validators.
- Confirmation latency: $T\_conf \approx T\_prop + T\_order + T\_commit + T\_anchor$.

For PBFT with batching (batch size B), on-chain cost is amortized $\approx T\_chain/B$ per decision.

- Throughput bound: $TPS\_end\text{-}to\text{-}end \leq \min\{ TPS\_ledger , TPS\_PDP , TPS\_PEP \}$.
- Merkle inclusion proof: proof length $\approx \lceil \log_2 N \rceil$ hashes.
- Privilege lifetime risk: $R \propto \lambda\_comp \times TTL$ (shorter TTL reduces exposure).

2.6 Evaluation Criteria & Comparative Method Governance & privacy one-paragraph

Governance & Privacy. The permissioned ledger is operated by multiple organizations with an endorsement policy (e.g., 2-of-N per channel). Only hashes/commitments and credential/policy metadata are on-chain; PHI/PII and raw logs remain off-chain in WORM storage with envelope encryption and access logging. Data minimization and selective disclosure (VCs) are enforced.

- Scalability/Latency: Tconf PEP/PDP decision time, batch effects B.
- Resilience to insider abuse: need for collusion >f to subvert logs/policy; revocation propagation time.
- Compliance readiness: audit completeness (on-chain commitments + off-chain retention).
- Overhead: storage growth $G(t) = G\_0 + r_{tx} . S_{tx} .t$ mitigation via channels/partitioning, pruning, or checkpointing.

2.7 Case Study Set-up

- **Health-care cloud:** EHR access via PEPs; clinicians carry DIDs/VCs; break-glass policies versioned on-chain.
- **Financial cloud:** high-value data rooms; trader/analyst roles; strong revocation SLAs and segregation of duties.

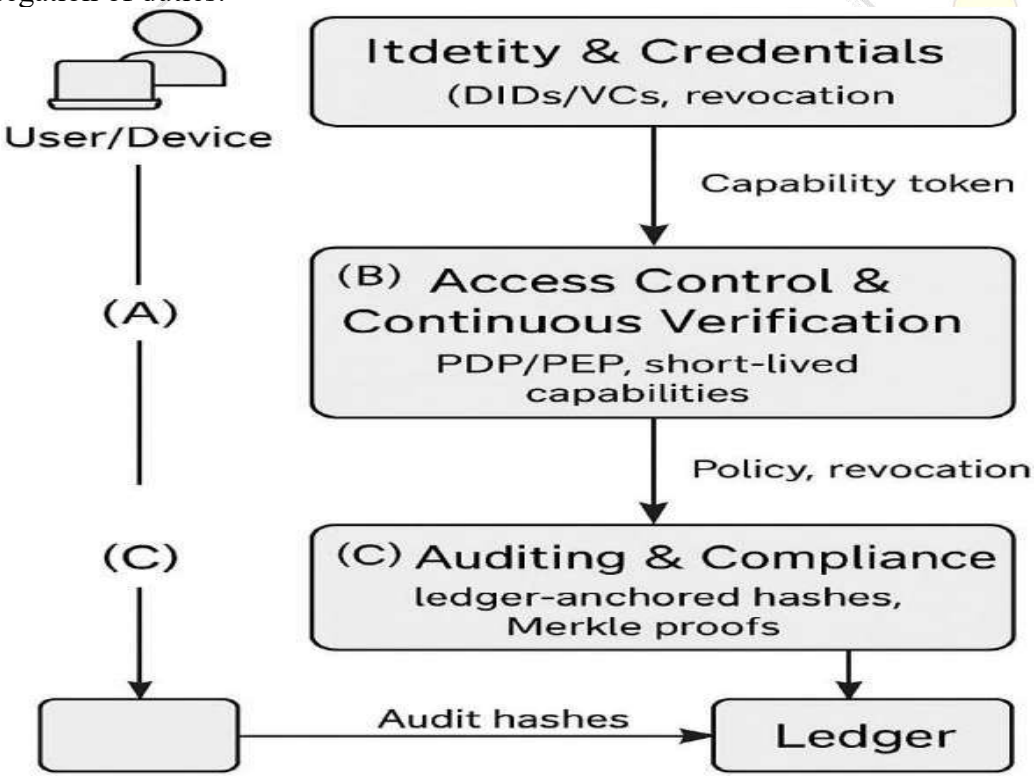

Figure 1. Blockchain-Integrated Zero-Trust Architecture

Three layers—(A) Identity & Credentials (DIDs/VCs; on-chain issuer & revocation registries), (B) Access Control & Continuous Verification (PEP/PDP; short-lived, DPoP-bound capabilities), and (C) Auditing & Compliance (Merkle-anchored audit hashes; off-chain WORM logs).

3. Results and Discussion

3.1 Analytic Results

Table 1. Plain ZTA vs Blockchain-Integrated ZTA

Criterion	Plain ZTA (central PDP/IdP)	Blockchain-Integrated ZTA
Trust broker	Single control-plane service	Federated via BFT validators (no single org can rewrite policy/audit)
Revocation	DB update; opaque to tenants	On-chain revocation list; verifiable to all parties
Audit trail	Provider-owned logs	Tamper-evident (hash/Merkle anchored) with inclusion proofs
Insider resistance	Admin can silently alter logs	Requires > f colluders; changes visible network-wide
Latency	Lower (no ledger hop)	Slightly higher; mitigated via batching/async anchoring
Ops complexity	Lower	Higher (governance, validator ops, privacy design)

- **Fault tolerance:** With n=10 validators, $f=\lfloor(10-1)/3\rfloor=3$. Thus ≥4 colluding faulty nodes are required to corrupt policy/audit consensus.
- **Latency composition:** For PBFT-style permissioned chains, T_{conf} is dominated by **ordering** and **commit**; batching requests (B) amortizes on-chain cost per decision, leaving PEP/PDP latency as the main per-request component.
- **Throughput:** End-to-end TPS is capped by the slowest element—often ledger commit rate under strict audit-every-decision, or PDP evaluation if policies include heavy external risk checks. A practical design is asynchronous audit anchoring (hash queue) with bounded staleness (e.g., ≤5 s).

3.2 Qualitative Outcomes - Case Studies Healthcare cloud

- **Identity:** DIDs/VCs remove dependence on one IdP; revocation is verifiable on-chain.
- **Access:** Short-lived capabilities prevent standing privileges; break-glass requires on-chain quorum with post-facto audit.
- **Audit:** Clinical accesses produce Merkle-anchored events enabling regulator proofs without exposing PHI on-chain.

Financial cloud

- **Segregation of duties:** Policy versions and approvals are immutable; insider attempts to widen access leave unavoidable traces.
- **Key rotation:** PKR algorithm enforces governed changes; token invalidation occurs within TTL windows (minutes).

4. Conclusion

A blockchain-integrated ZTA decentralizes trust brokering, makes revocation and policy versioning verifiable, and yields tamper-evident audits—key advantages for multi-tenant, regulated clouds. With careful engineering (permissioned BFT, short-lived capabilities, off-chain payloads, async anchoring), the design improves insider resistance and compliance posture with manageable latency overheads. This positions block chain-empowered ZTA as a strong candidate for next-generation cloud security.

References

1. Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. OSDI. Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the Weil pairing. ASIACRYPT.
2. Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. OSDI. Fernandes, E., Oliveira, R., & Silva, T. (2022). Blockchain for zero-trust cloud architectures: A

systematic review. *Journal of Cloud Computing*, 11(1), 1–15.

3. Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. Elsevier. Kairouz, P., et al. (2021). Advances and open problems in federated learning. *FnT in ML*, 14(1–2), 1–210.
4. Kindervag, J. (2010). Build security into your network's DNA: The zero trust network architecture. Forrester.
5. Kumar, R., Singh, A., & Sharma, V. (2022). Decentralized identity management for secure cloud ecosystems. *IEEE Access*, 10, 45789–45801.
6. Moin, A., Karim, A., & Rahman, M. (2019). Blockchain for secure access control in IoT. *FGCS*, 95, 27–43.
7. Mollah, M. B., Zhao, J., Niyato, D., Lam, K. Y., & Yang, L. (2021). Blockchain for future cloud computing: A survey. *Computers & Security*, 102, 102121.
8. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture*, NIST SP 800-207.
9. Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. EUROCRYPT. (ABE foundations)
10. Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology? *PLoS ONE*, 11(10), e0163477.
11. Zaharia, M., et al. (2016). Apache Spark: A unified engine for big data processing. *CACM*, 59(11), 56–65.
12. Zhang, R., & Jacobsen, H. (2018). Towards dependable, scalable, and pervasive distributed ledgers with blockchains. *PVLDB*, 10(12), 1730–1741.
13. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., … Yellick, J. (2018). *Hyperledger Fabric: A distributed operating system for permissioned blockchains*. In *Proceedings of the Thirteenth EuroSys Conference 2018*. Association for Computing Machinery.
14. Hu, V. C., Ferraiolo, D. F., Kuhn, D. R., & Chandramouli, R. (2014). *Guide to attribute based access control (ABAC)* (NIST Special Publication 800–162). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-162
15. Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., & Waite, D. (2023). *OAuth 2.0 demonstrating proof of possession (DPoP)* (RFC 9449). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc9449 r
16. Kantara Initiative. (2018). *User-Managed Access (UMA) 2.0 grant for OAuth 2.0 authorization*.
17. The SPIFFE Project. (2022). *SPIFFE Workload API (v1.0) specification*. Cloud Native Computing Foundation.
18. World Wide Web Consortium. (2022). *Decentralized identifiers (DIDs) v1.0*. W3C Recommendation.
19. World Wide Web Consortium. (2022). *Verifiable credentials data model v1.1*. W3C Recommendation.