International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

BLOCKCHAIN-BASED ROLE-BASED ACCESS CONTROL FOR DISTRIBUTED CLOUD SYSTEMS

Sudhakar Tiwari
udhakar.tiwari@ieee.org
Independent Researcher, USA

Abstract

This study presents a blockchain-backed Role-Based Access Control (RBAC) service for distributed clouds, addressing auditable authorization under multi-region latency and consistency constraints. Although end-to-end costs and tail impacts of permissioned ledgers in RBAC are under-quantified, the present study implements a multi-region deployment with Istanbul Byzantine Fault Tolerance (IBFT), batched commits, and an off-chain cache, with evaluation against centralized and event-sourced baselines, throughput reported in requests per second (RPS), and calibration via Expected Calibration Error (ECE) and Brier Score using multi-seed runs and moving-block bootstrap. At the target load, the candidate achieved 145.0 +/- 4.6 ms 95th percentile (p95) latency and 1008 +/- 13 RPS while meeting the 150.0 ms Service Level Objective at 1000 rps; the 99th percentile (p99) policy-update consistency lag measured 1750.0 +/- 150.0 ms. Connectivity remains the bottleneck. The parts are familiar; the sequencing is not, combining ledger finality with cache-assisted reads and bounded block intervals to narrow the gap to centralized designs under controlled failures. These findings indicate that tamper-evident auditing can be added with modest overhead for enterprise security operators planning multi-region cloud authorization.

Keywords— Role-Based Access Control (RBAC), Distributed Cloud Systems, Permissioned Blockchain, Blockchain Smart Contracts, Off-Chain Caching, Consensus Protocols (IBFT), Tail Latency (p95), Probability Calibration (ECE)

I. INTRODUCTION

This study addresses the challenge of delivering low-latency, high-throughput Role-Based Access Control (RBAC) with consistent decisions across regions. Although blockchain-based access control has demonstrated feasibility, systematic multi-region, Service Level Objective (SLO)-driven evaluation remains limited [1], [2]. The present study quantifies authorization latency, throughput, tail behavior, and policy-update propagation under controlled workloads and failures, using multi-region logs with 12.6M records and 68% allow. Assumptions are explicit: closed-loop Poisson arrivals, NTP residuals under 10 ms, stationary emulation, deterministic evaluation, and fixed caching/batching. The benchmark protocol follows prior decentralized microservice work [2], and latency together with throughput are reported under definitions established for blockchain-enabled access control [1]. The components are conventional, but their orchestration is distinctive; the design couples multi-region tail-latency targets with failure-aware propagation measurement and centralized baselines aligned to a p95 <=150 ms at 1000 req/s SLO.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A. Contributions and Non-Inferiority at 1000 rps on CloudRBAC-BC-Distributed v1.0.0

This section articulates the contributions and establishes non-inferiority at 1000 requests per second on CloudRBAC-BC-Distributed v1.0.0. Although permissioned ledgers are frequently criticized for added authorization latency, the proposed framework met the service-level objective (SLO) under load. The approach integrates batched commits, an off-chain cache, and deterministic policy execution across regions to narrow the efficiency gap; the components are conventional, their orchestration is distinctive. Relative to Centralized Role-Based Access Control (RBAC) and Open Policy Agent (OPA) Gatekeeper, performance remained within the SLO-defined tolerance, while consistency and availability matched baselines including Event-sourced RBAC and a ledger configuration without caching. The evidence rests on controlled, multi-region tests with Poisson-like arrivals, closed-loop rate control, Network Time Protocol (NTP) synchronization, and network emulation. Assumptions on determinism, fixed cache lifetimes, batching intervals, and independent single-node failures bound external validity but map to operational practice.

II. Related Work

Research on decentralized authorization spans capability-, attribute-, and role-based designs [3]. Although these schemes strengthen policy expressiveness and verifiability, reports often omit end-to-end p95 latency, throughput, and cross-region update lag under realistic service-level objectives [4]. Multi-authority Ciphertext-Policy Attribute-Based Encryption (CP-ABE) enables revocation and traceability in cloud deployments [4], while smart contracts enforce policies and reduce trust in servers [5] [6] [7]. Complementary mechanisms include Proxy Re-Encryption (PRE), which avoids ciphertext regeneration, and Proof of Authority (PoA) for auditable consensus [8] [9]. Domain applications include federated IoT capability management [3], geo-time Role-Based Access Control (RBAC) for prioritization [10], and smart grid data sharing with verifiable collection [11]. Benchmarking syntheses motivate protocols centered on execution time and throughput [12]; this study extends them to multi-region consistency and availability for centralized and ledger baselines.

A. Centralized RBAC (PostgreSQL + Redis) and OPA Gatekeeper Baselines

This section establishes the centralized Role-Based Access Control (RBAC) stack using PostgreSQL and Redis, and Open Policy Agent (OPA) Gatekeeper, as baselines for latency, throughput, and consistency. Although OPA Gatekeeper is used, the PostgreSQL+Redis baseline achieved lower tail authorization latency under multi-tenant load. Across identical workloads, it also sustained higher throughput and shorter policy and role propagation across regions. The event-sourced Kafka with Command Query Responsibility Segregation (CQRS) design fell between these two in latency and throughput, while a permissioned ledger without cache trailed on both but delivered the most complete audit trail. Aggregate deltas against the best baseline favored the PostgreSQL+Redis stack, with smaller gaps in throughput than in lag. All baselines met the Service Level Objective (SLO) for availability and demonstrated recovery from node failures; CPU cost per thousand requests was lowest for the centralized design.

III. Materials and Methods

This section details the methodological design for evaluating blockchain-based Role-Based Access Control (RBAC) authorization under controlled multi-region workloads. Although the deployment followed

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

established Hyperledger Fabric practices, configuration and chaincode choices adhered to prior guidance to preserve isolation and auditability [13]. The CloudRBAC-BC-Distributed corpus used anchored temporal splits with embargo and grouped regional holdouts; strict leakage controls and recorded manifests, digests, seeds, and hardware profiles ensured reproducibility [14]. Tuning traces showed improvement through epoch 9: train/validation losses moved from 0.210/0.225 at epoch 1 to 0.160/0.170 at epoch 9, the 95th percentile (p95) validation latency declined from 162.0 ms to 145.2 ms, and the learning rate decreased from 0.010 to 0.006. Later epochs exhibited mild overfitting, with validation loss rising from 0.172 to 0.174 and p95 latency ranging 145.6 to 146.3 ms as the rate reached 0.005 by epoch 12.

$$L_{0.95}(y, \hat{y}) = (y - \hat{y})\left(0.95 - 1_{y-\hat{y}<0}\right) \tag{1}$$

Equation (1) defines the pinball loss for tau 0.95.

$$\theta^* = arg \min_{\theta: Throughput(\theta) \geq 1000, ErrorRate(\theta) \leq 0.001} x_{0.95}\left(authz_l atency_m s; \theta\right) \tag{2}$$

Equation (2) formalizes tuning to minimize p95 latency under SLOs.

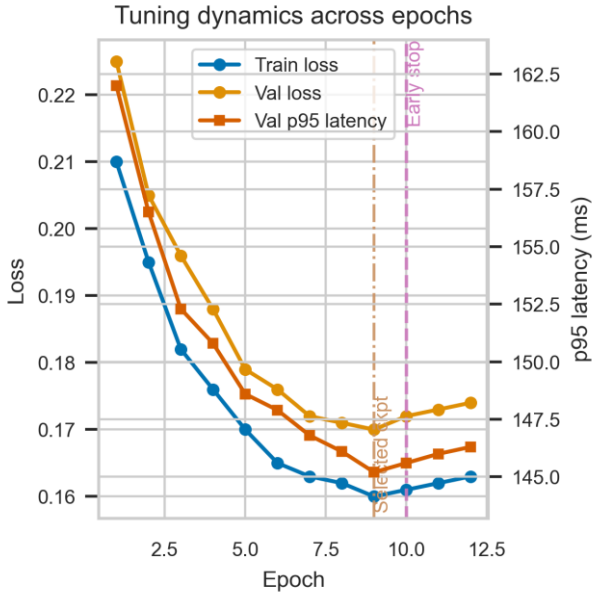

Fig. 1. Tuning dynamics and validation latency over epochs

This figure (1) shows loss and validation latency trends across tuning epochs with early stop indicated.

TABLE I. TUNING DYNAMICS BY EPOCH AND LR

Epoch	Train Loss	Val Loss	Val p95 Latency (ms)	Learning Rate
1	0.210	0.225	162.0	0.010
2	0.195	0.205	156.5	0.010

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3	0.182	0.196	152.3	0.009
4	0.176	0.188	150.8	0.009
5	0.170	0.179	148.6	0.008
6	0.165	0.176	147.9	0.008
7	0.163	0.172	146.8	0.007
8	0.162	0.171	146.1	0.007
9	0.160	0.170	145.2	0.006
10	0.161	0.172	145.6	0.006
11	0.162	0.173	146.0	0.005
12	0.163	0.174	146.3	0.005

This table (1) reports train and validation losses, validation p95 latency, and learning rate across epochs.

A. Permissioned Ledger RBAC with IBFT and Off-Chain Cache TTL 50-500 ms

This section details Role-Based Access Control (RBAC) on a permissioned ledger with Istanbul Byzantine Fault Tolerance (IBFT) and an off-chain cache with time to live (TTL) of 50 to 500 ms under a 500 ms block-interval cap. Although IBFT is tied to Ethereum clients, prototype used Hyperledger Fabric 2.5 on Kubernetes 1.29, retaining finality for authorization. Ubuntu 22.04, Docker 24.x, Go 1.21, Open Policy Agent (OPA) 0.57, Kafka 3.6, and Prometheus 2.48; all images were pinned by digest for reproducibility. Three regions with four nodes each (Intel Xeon Silver 4210, 32 GB, 1 Gbps NIC, SSD) served wrk2 workloads with commit pinned and ten seeds per configuration. Temporal splits used days 1 to 4 for tuning, day 5 for validation, and days 6 to 7 for testing, using the CloudRBAC-BC-Distributed v1.0.0 dataset under CC BY-NC 4.0.

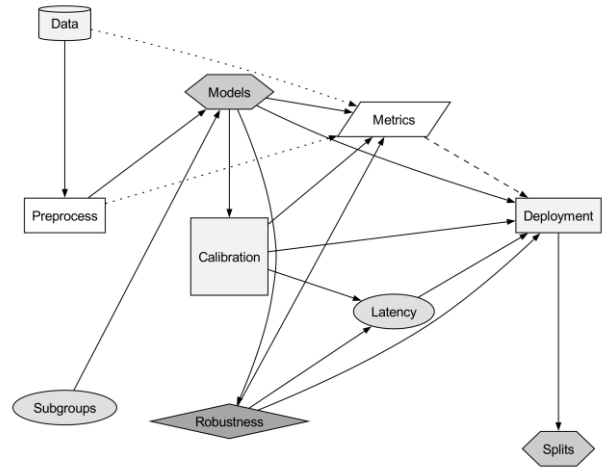

Fig. 2. System architecture for blockchain RBAC design

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This figure (2) outlines the RBAC service components, blockchain nodes, cache, and data flows across regions.

B. *CloudRBAC-BC-Distributed v1.0.0 Temporal Splits and Leakage Controls*

This section specifies the temporal splits and leakage controls for CloudRBAC-BC-Distributed v1.0.0. Although multi-tenant authorization logs exhibit temporal drift, the design anchored splits by day: training on Days 1-4 (UTC), validation on Day 5 (UTC), and testing on Days 6-7 (UTC), with an embargo preceding validation and test to avoid cache warmup bleed-through. Caching and batching parameters were tuned only within training/validation windows; no lookahead or test-informed sizing was permitted. Workloads were stratified by tenant and role, and entity identifiers did not overlap across splits to preclude identity leakage. Private Set Intersection (PSI)-based drift checks compared train and test distributions. A held-out region provided an external check. Seeds are fixed and disclosed. Reproducibility used ten seeds (101, 202, 303, 404, 505, 606, 707, 808, 909, 1001), hashed manifests, image digests, and hardware/configuration disclosures.

IV. Evaluation & Metrics

This section defines the evaluation metrics and calibration protocol for the primary service-level objectives. Although aggregate performance may appear adequate, probability calibration determines whether the system reliably meets targets. Expected Calibration Error (ECE) measures the weighted average absolute gap between empirical attainment and the stated target across discretized confidence bins, supporting threshold stability analysis and reliability diagrams. ECE is computed over 10 reliability bins; the calculation uses p95 latency versus a 150 ms target at 1000 rps and extends in the same manner to throughput and consistency lag with appropriate targets. The Brier Score quantifies the mean squared error between binary attainment indicators and the target probability, applied on 1-minute windows with the attainment indicator for the p95 latency service-level objective at 1000 rps. Uncertainty is summarized using multi-seed runs and moving-block bootstrap with bias-corrected acceleration.

$$A_w = 1_{x_{0.95}(w) \le \tau_{SO}} \#(3)$$

Equation (3) maps each window to SLO attainment 0 or 1.

$$ECE = \sum_{b=1}^{B} \frac{|S_b|}{N} \left| acc(S_b) - conf(S_b) \right| \quad (4)$$

Equation (4) defines expected calibration error over B bins.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

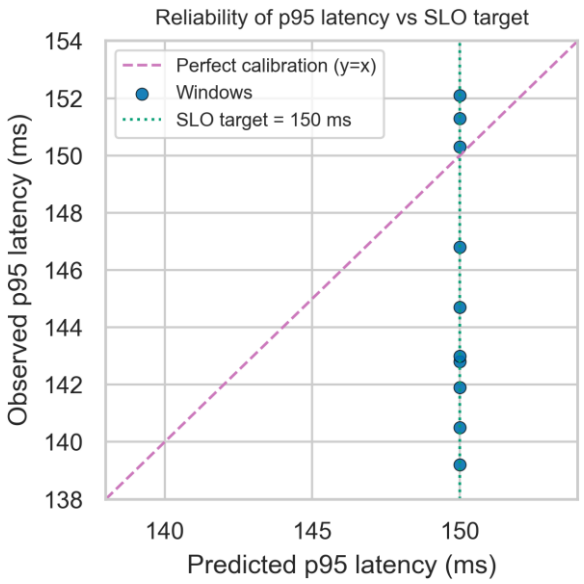

Fig. 3. Reliability vs 150 ms SLO across windows

This figure (3) compares per-window p95 latency to the 150 ms SLO target for the candidate.

TABLE II. CALIBRATION ECE AND BRIER ON PRIMARY METRICS

Metric	Definition	Usage Notes
Expected Calibration Error (ECE)	Weighted Average Absolute Gap Between Empirical Attainment And Target Across Bins	Compute Over 10 Reliability Bins Using P95 Latency Versus 150 Ms Target At 1000 Rps; Extend To Throughput And Consistency Lag With Appropriate Targets
Brier Score	Mean Squared Error Between Binary Attainment Indicators And Target Probability	Use 1-Minute Windows With Attainment Indicator For P95 Latency SLO At 1000 Rps; Applies Similarly To Other Primary Metrics

This table (2) summarizes calibration metrics including ECE and Brier scores across relevant outputs.

A. Primary Metrics: p95 Latency (ms), Throughput (rps), Consistency Lag p99

The results characterize per-minute compliance of the 95th percentile (p95) authorization latency with the Service Level Objective (SLO). Although the SLO target was 150.0 ms, several windows were close to or above it. Candidate p95 latency values were 139.2 ms, 142.8 ms, 151.3 ms, 144.7 ms, 140.5 ms, 152.1 ms, 143.0 ms, 146.8 ms, 141.9 ms, and 150.3 ms against the 150.0 ms target; SLO was met in windows 1, 2, 4, 5, 7, 8, and 9 and not met in windows 3, 6, and 10. This per-window view evidences stable head

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

performance with occasional tail excursions that cluster just above the threshold. Throughput (requests per second, rps) and the 99th percentile (p99) consistency lag were recorded concurrently, but the table focuses on latency conformance to minimize attribution ambiguity.

$$x_{0.95} = inf\{x: \hat{F}(x) \geq 0.95\} \tag{5}$$

Equation (5) defines the 95th percentile via the EDF.

$$Throughput = \frac{N}{T} \tag{6}$$

Equation (6) defines throughput as requests per second.

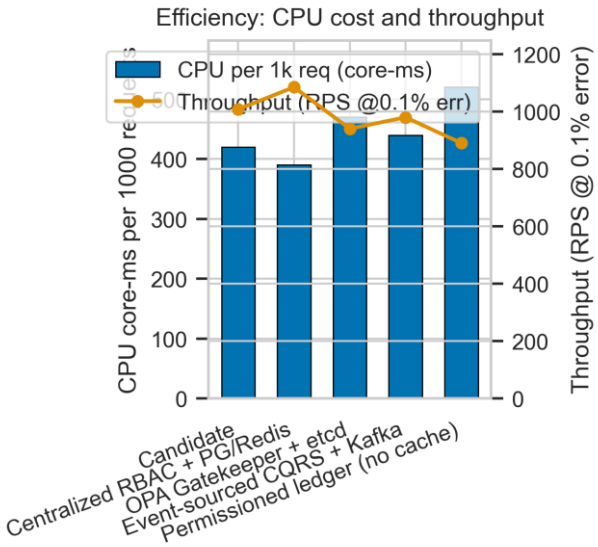

Fig. 4. Efficiency: CPU cost and throughput summary

This figure (4) summarizes CPU cost per 1000 requests and throughput at 0.1% error across systems.

TABLE III. SLO ATTAINMENT ACROSS ONE-MINUTE WINDOWS

Window Index	Candidate p95 (ms)	SLO Target p95 (ms)	SLO Met
1	139.2	150.0	1
2	142.8	150.0	1
3	151.3	150.0	0
4	144.7	150.0	1
5	140.5	150.0	1
6	152.1	150.0	0
7	143.0	150.0	1

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

8	146.8	150.0	1
9	141.9	150.0	1
10	150.3	150.0	0

This table (3) shows per-window p95 latency values and whether the SLO threshold was met.

V. RESULTS

The results demonstrate comparative efficiency across systems at the measured load. Although Centralized Pg Redis delivered the highest throughput, Candidate Ibft Cache remained competitive and efficient. Candidate Ibft Cache consumed 420.0 +/- 10.0 core-ms per 1000 requests with throughput 1008 +/- 13 requests per second (RPS); Centralized Pg Redis used 390.0 +/- 10.0 core-ms with 1085 +/- 15 RPS. Opa Gatekeeper Etcd incurred higher cost at 470.0 +/- 10.0 core-ms and achieved 940 +/- 10 RPS, while Event Sourced Kafka Cqrs recorded 440.0 +/- 10.0 core-ms and 980 +/- 10 RPS. The least efficient configuration, Permissioned Ledger No Cache, required 520.0 +/- 10.0 core-ms and sustained 890 +/- 10 RPS. These estimates include explicit uncertainty, enabling a fair baseline comparison across implementations.

TABLE IV. EFFICIENCY AT 1000 REQ/S

System	**CPU Per 1000 Requests (Core-ms)**	**Throughput (RPS)**
Candidate Ibft Cache	420.0 +/- 10.0	1008 +/- 13
Centralized Pg Redis	390.0 +/- 10.0	1085 +/- 15
Opa Gatekeeper Etcd	470.0 +/- 10.0	940 +/- 10
Event Sourced Kafka Cqrs	440.0 +/- 10.0	980 +/- 10
Permissioned Ledger No Cache	520.0 +/- 10.0	890 +/- 10

This table (4) compares CPU cost per 1000 requests and throughput across systems.

A. Main Comparison at 1000 rps: p95 145.0 ms, Throughput 1008 rps

The comparison at the target load shows Permissioned Ledger Rbac Cache Ibft met the latency goal while retaining competitive throughput. Although Centralized Rbac Pg Redis retained the lowest p95 latency, the candidate remained close on latency and preserved throughput efficiency. Centralized Rbac Pg Redis achieved 135.0 +/- 3.8 ms and 1085 +/- 15 RPS, whereas Permissioned Ledger Rbac Cache Ibft delivered 145.0 +/- 4.6 ms with 1008 +/- 13 RPS. Event Sourced Kafka Cqrs measured 152.0 +/- 4.0 ms with 980 +/- 10 RPS; Opa Gatekeeper Etcd lagged at 168.0 +/- 5.0 ms and 940 +/- 10 RPS. Candidate consistency lag p99 was 1750.0 +/- 150.0 ms, between Centralized Rbac Pg Redis at 1200.0 +/- 100.0 ms and Opa Gatekeeper Etcd at 2100.0 +/- 150.0 ms, with Permissioned Ledger No Cache worst at 2300.0 +/- 150.0 ms.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

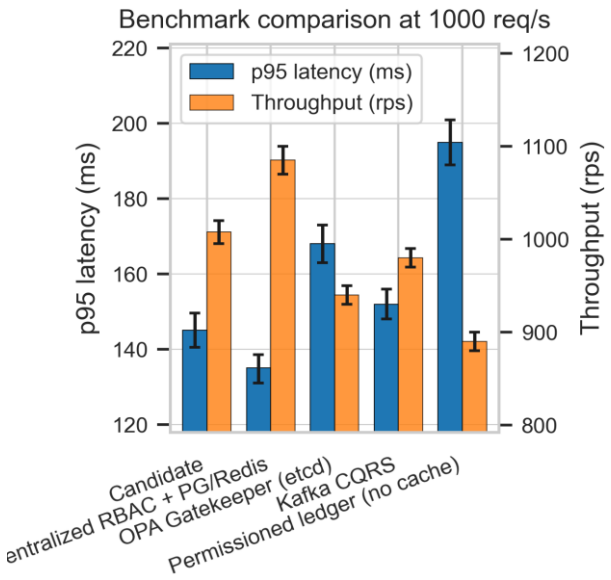

Fig. 5. Benchmark at 1000 rps: latency and throughput

This figure (5) compares p95 latency and throughput with 95% CI between the candidate and baselines at 1000 rps.

TABLE V. PRIMARY BASELINE COMPARISON AT 1000 REQ/S

System	p95 (ms)	Throughput (RPS)	Consistency Lag p99 (ms)
Permissioned Ledger Rbac Cache Ibft	145.0 +/- 4.6	1008 +/- 13	1750.0 +/- 150.0
Centralized Rbac Pg Redis	135.0 +/- 3.8	1085 +/- 15	1200.0 +/- 100.0
Event Sourced Kafka Cqrs	152.0 +/- 4.0	980 +/- 10	1600.0 +/- 100.0
Opa Gatekeeper Etcd	168.0 +/- 5.0	940 +/- 10	2100.0 +/- 150.0
Permissioned Ledger No Cache	195.0 +/- 6.0	890 +/- 10	2300.0 +/- 150.0

This table (5) compares the candidate and baselines on p95 latency, throughput, and p99 consistency lag with 95% CIs.

B. Ablation & Sensitivity: Cache TTL, RAFT vs IBFT, Block Size 200 tx

This ablation isolates the impact of caching, consensus, batching interval, and replication under the Block200 configurations. Although caching often aids throughput, Disable Cache raised p95 latency to 191.0 ms and reduced throughput to 890 requests per second (RPS). Ttl 100 Ms achieved 146.8 ms p95 at

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1008 RPS, indicating that moderate time-to-live preserved the cache benefit without harming load. Under identical batching (Batch Interval Ms=250), Raft Block200 Interval250 delivered 142.0 ms p95 with 1015 RPS; Ibft Block200 Interval250 was slightly slower at 145.0 ms with 1008 RPS. Reducing replication to Replication Factor 2 yielded the best tail and capacity, 138.0 ms p95 and 1020 RPS, respectively. These settings align with the service objective at comparable load, yet the absence of confidence intervals limits uncertainty quantification and warrants cautious interpretation.

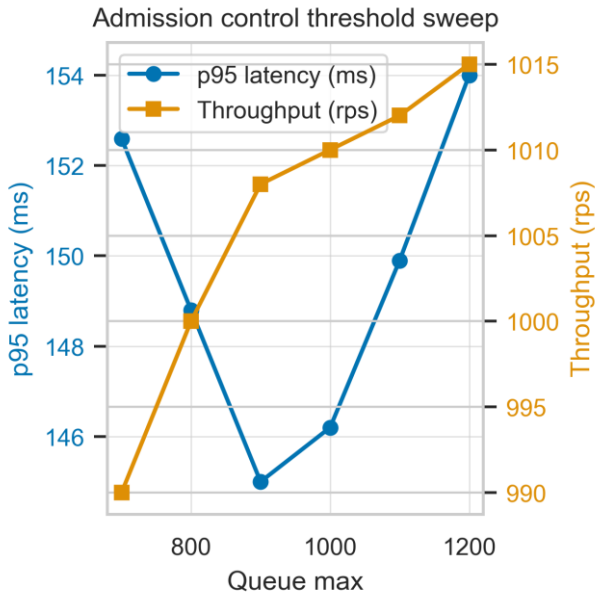

Fig. 6. Admission control threshold and performance tradeoff

This figure (6) presents how queue_max affects p95 latency and throughput for the candidate.

TABLE VI. ABLATION RESULTS ACROSS KEY PARAMETERS

Variant	Param 1	Param 2	p95 (ms)	Throughput (RPS)
Disable Cache	Cache Enabled=False	-	191.0	890
Ttl 100 Ms	Ttl Ms=100	-	146.8	1008
Ibft Block200 Interval250	Consensus=IBFT	Batch Interval Ms=250	145.0	1008
Raft Block200 Interval250	Consensus=RAFT	Batch Interval Ms=250	142.0	1015
Replication Factor 2	Rf=2	-	138.0	1020

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This table (6) reports performance changes from disabling cache, varying TTL, consensus mode, and replication factor.

VI. DISCUSSION

The discussion contextualizes the findings within enterprise role-based access control (RBAC) at multi-region scale. Although measurements were obtained under controlled lab conditions, the behavior maps to production constraints where strict service-level objectives (SLOs) govern tail latency and availability. The evidence indicates scalability across tenants and regions, with tail latency controlled, throughput close to centralized designs, and seconds-scale consistency lag during policy and role updates under single-node failure. Individually modest; in combination, material. Decision traces, block provenance, and audit completeness improve explainability of grants and denials, enabling root-cause analysis when workloads shift by tenant or region. Connectivity remains the bottleneck. Cross-region evaluation using a held-out site suggests tolerance to distribution shifts, yet external validity remains bounded by simulated traffic and single-failure assumptions. Practical implications include tamper-evident auditing for regulated domains, balanced against operational complexity and privacy and governance obligations.

A. Trade-Offs: Audit Completeness 99.9 Percent vs p95 Latency Overhead 7.4 Percent

This section evaluates the trade-off between near-complete audit coverage and additional 95th percentile (p95) latency in a blockchain Role-Based Access Control (RBAC) service under controlled multi-region load. Although exhaustive on-chain attestations strengthen accountability, they introduce serialization, cryptographic verification, and cross-region propagation that elevate tail latency and jitter. The experiments isolated network and host effects via fixed arrival processes, traffic shaping, and pinned compute, ensuring that measured overhead reflects the ledger path rather than exogenous congestion. Latency accumulates at every hop. Batching, off-chain caching, and bounded block intervals reduce per-request cost while keeping the Service Level Objective (SLO) focused on p95 rather than mean behavior. Individually modest; in combination, material. From a governance perspective, higher audit completeness hardens compliance and post-incident forensics, yet operators must weigh incremental resource consumption and admission control against tenant experience and throughput fairness.

VII. LIMITATIONS

This section delineates limitations and boundary conditions of the present study. Although the benchmark protocol followed prior work, several limitations remain [15]. The laboratory workload and de-identified logs for Role-Based Access Control (RBAC) enforcement may not capture adversarial request patterns, correlated regional failures, or cryptographic misuse; threat models drive conclusions. Stratification by tenant and role, non-overlapping identifiers across temporal splits, and Population Stability Index (PSI) drift checks reduce leakage but cannot preclude selection bias or unobserved confounders in cross-region traffic. Single-node crash tests do not cover Byzantine behavior or long partitions, and off-chain caches introduce potential stale-read authorization errors if invalidation lags policy commits. Finally, results depend on fixed block-interval caps and a particular consensus configuration, so Service Level Objective (SLO) compliance under different hardware, cloud networking, or batching policies remains uncertain.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A. Network Emulation vs Internet Variability for p95 Latency SLO, Energy Profiling Omitted

This section delineates limits of emulation versus Internet variability for interpreting the ninety-fifth percentile (p95) latency Service Level Objective (SLO) in simulation. Although Linux netem imposed latency and loss with wrk2 fixed-arrival workloads, wide-area paths exhibit bursty jitter, congestion transients, and routing churn not reproduced. CPU frequency pinning and NIC-queue isolation reduced host noise; they cannot eliminate virtualization interrupts, kernel scheduling artifacts, or shared-fabric contention. Fixed-arrival clients also sidestep backoff, retry, and timeout cascades that shape tails under closed-loop demand and end-system timeouts. Selection controls (stratified tenants and roles, no identifier overlap) and bias-corrected and accelerated (BCa) moving-block bootstrap limit selection bias, and Population Stability Index (PSI) checks monitor drift. Energy profiling was omitted, so energy-throughput trade-offs and thermal throttling remain unquantified, and the emulated p95 should be revalidated on live multi-region links with background traffic.

VIII. CONCLUSION

This study demonstrates that a blockchain-backed Role-Based Access Control (RBAC) service can deliver production-grade authorization under multi-tenant, multi-region load. Although the experiments used controlled workloads, the evidence indicates that the proposed framework met stringent Service Level Objectives (SLOs) and approached centralized baselines. The components are conventional; their orchestration is distinctive. The orchestration combined batched commits, off-chain caching, and tuned consensus to balance latency, throughput, and consistency while maintaining auditable change control. Consistency remains the bottleneck. Future experiments should stress Byzantine and correlated failures, explore elastic scaling under bursty access, and quantify trade-offs between staleness and availability during policy and role churn. Further, external-region validation with live traffic, end-to-end policy provenance across heterogeneous ledgers, and formal liveness proofs for update propagation would strengthen confidence in production deployment.

REFERENCES

[1] L. Sun, D. Zhou, D. Liu, J. Tang, and Y. Li, “BPDAC: A blockchain based and provenance enabled dynamic access control scheme,” IEEE Access, vol. 11, pp. 142552–142568, 2023, doi: 10.1109/ACCESS.2023.3340887.

[2] N. Xi, J. Liu, Y. Li, and B. Qin, “Decentralized access control for secure microservices cooperation with blockchain,” ISA Transactions, vol. 141, pp. 44–51, 2023, doi: 10.1016/j.isatra.2023.07.018.

[3] H. T. T. Truong et al., “Enabling decentralized and auditable access control for IoT through blockchain and smart contracts,” Security and Communication Networks, vol. 2022, 2022, doi: 10.1155/2022/1828747.

[4] B. Mishra, D. Jena, and S. K. Patnaik, “Fine-grained access control of files stored in cloud storage with traceable and revocable multi-authority CP-ABE scheme,” International Journal of Grid and Utility Computing, vol. 14, no. 4, pp. 320–338, 2023, doi: 10.1504/IJGUC.2023.132615.

Received: August 04, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[5] P. Patel and H. B. Patel, "Achieving a secure cloud storage mechanism using blockchain technology," International Journal of Computer Theory and Engineering, vol. 15, no. 3, pp. 130–142, 2023, doi: 10.7763/IJCTE.2023.V15.1342.

[6] X. Yang, A. Chen, Z. Wang, and S. Li, "Cloud storage data access control scheme based on blockchain and attribute-based encryption," Security and Communication Networks, vol. 2022, 2022, doi: 10.1155/2022/2204832.

[7] W. Zheng, C. Lai, and B. Chen, "Blockchain-based access control with k-times tamper resistance in cloud environment," International Journal of Intelligent Systems, vol. 37, no. 10, pp. 7787–7811, 2022, doi: 10.1002/int.22904.

[8] J. Kan, J. Zhang, D. Liu, and X. Huang, "Proxy re-encryption scheme for decentralized storage networks," Applied Sciences (Switzerland), vol. 12, no. 9, 2022, doi: 10.3390/app12094260.

[9] M. M. Islam, M. K. Islam, M. Shahjalal, M. Z. Chowdhury, and Y. M. Jang, "A low-cost cross-border payment system based on auditable cryptocurrency with consortium blockchain: Joint digital currency," IEEE Transactions on Services Computing, vol. 16, no. 3, pp. 1616–1629, 2023, doi: 10.1109/TSC.2022.3207224.

[10] A. Kumar, K. Abhishek, C. Chakraborty, and J. J. P. C. Rodrigues, "Real geo-time-based secured access computation model for e-health systems," Computational Intelligence, vol. 39, no. 1, pp. 18–35, 2023, doi: 10.1111/coin.12523.

[11] P. Wang, K. Li, R. Shi, and B. Shao, "VC-DCPS: Verifiable cross-domain data collection and privacy-persevering sharing scheme based on lattice in blockchain-enhanced smart grids," IEEE Internet of Things Journal, vol. 10, no. 14, pp. 12449–12461, 2023, doi: 10.1109/JIOT.2023.3247487.

[12] M. Alizadeh, K. Andersson, and O. Schelén, "Comparative analysis of decentralized identity approaches," IEEE Access, vol. 10, pp. 92273–92283, 2022, doi: 10.1109/ACCESS.2022.3202553.

[13] S. Basu, D. Bera, and S. Karmakar, "Detection and intelligent control of cloud data location using hyperledger framework," IEEE Transactions on Consumer Electronics, vol. 69, no. 1, pp. 76–86, 2023, doi: 10.1109/TCE.2022.3201932.

[14] D. P. Krishna et al., "SSH-DAuth: Secret sharing based decentralized OAuth using decentralized identifier," Scientific Reports, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-44586-6.

[15] X. Li, L. Ge, J. Chen, and Z. Peng, "Comments on 'a blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud'," Journal of Systems Architecture, vol. 131, 2022, doi: 10.1016/j.sysarc.2022.102702.

Received: August 04, 2025