# Query Performance Analysis of Blockchain-Based Audit Trails for Insurance Access Control

Anthony Uchenna Eneh [1], Love Allen Chijioke Ahakonye [2], Jae Min Lee [1], Dong-Seong Kim [1] [*]
[1] IT-Convergence Engineering, *Kumoh National Institute of Technology*, Gumi, South Korea
[*] NSLab Co. Ltd., Gumi, South Korea, *Kumoh National Institute of Technology*, Gumi, South Korea
[2] ICT Convergence Research Center, *Kumoh National of Technology*, Gumi, South Korea
(anthony, loveahakonye, dskim, ljmpaul)@kumoh.ac.kr

***Abstract*—Blockchain-based access control offers tamper-proof audit trails for regulatory compliance, but prior work has not quantified query performance, critical for compliance workflows. We present the first comparison of Ethereum event logs against PostgreSQL and MongoDB for audit queries, benchmarking four patterns (by subject, resource, time range, and denials) across 100-10,000 events. Results show blockchain-indexed queries are 6–8× slower than PostgreSQL, while scan-heavy queries are 35–41× slower, with significantly higher gas costs, trading latency for immutability. These findings inform hybrid architectures: routing critical decisions to blockchain while delegating routine logs to databases.**

***Index Terms*—Audit log, Access control, Benchmarking, Blockchain, Ethereum, Insurance**

## I. Introduction

Auto-insurance claims processing involves sensitive evidence across multiple stakeholders, with regulatory frameworks (GDPR, HIPAA) mandating comprehensive audit trails. Compliance officers must routinely query these trails to answer questions such as "who accessed this claim file?" or "which access requests were denied last month?" These queries demand low latency and efficient filtering. Traditional centralized systems suffer from tampering vulnerability [1], where malicious insiders can alter or delete logs without detection. ClaimGuard [2], a blockchain-based ABAC system, addresses this by emitting immutable audit events to the Ethereum blockchain, ensuring tamper-evident records that no single party can unilaterally modify. Prior work demonstrated authorization correctness but **never quantified audit log queryability**, a critical gap for production deployments where compliance workflows depend on responsive audit queries.

Related work on blockchain audit logging includes BLSQ [3], which employs Merkle tree optimization for tamper-proof log storage but does not benchmark query latency against databases; BCALS [4], a cloud log management system reporting 55–65 ms insertion latency while focusing on security rather than query patterns; hybrid blockchain-database benchmarks [5] analyzing write throughput (100–500 tps) without audit query evaluation; and VeriBench [6], which categorizes verifiable databases but omits Ethereum-to-SQL comparisons. Blockchain ABAC systems such as SmartAccess [7] (10–50 tps, 200–800 ms latency) and EHR auditing frameworks [8] lack query benchmarks, while Gürsoy et al. [9] report 35–750 ms query latencies for genomic data rather than compliance patterns.

This paper provides the first audit query performance analysis of blockchain-based access control, comparing Ethereum, PostgreSQL 16, and MongoDB 7 across four compliance patterns (100-10K events).

## II. System Methodology

### A. ClaimGuard Audit Architecture

ClaimGuard's audit mechanism consists of an on-chain `AccessAuditLog` contract that emits indexed events for each access decision, as shown in the Smart contract below.

```
event AccessChecked(
    address indexed subject,
    bytes32 indexed resourceIdHash,
    bytes32 indexed actionHash,
    bool allow,
    uint256 ts
);
```

The Policy Enforcement Gateway (PEG) calls `logAccess()` after evaluating policies. Events persist permanently in blockchain transaction logs, queryable via Web3 `get_logs()` with indexed filters and block ranges.

**Query Interface:** Ethereum's event logs support filtering on indexed fields (`subject`, `resourceIdHash`, `actionHash`) and block ranges (`from_block`, `to_block`). The `allow` flag is not indexed, so denial queries require full log scanning.

### B. Baseline Systems

**PostgreSQL 16:** B-tree indexes on `subject`, `resource_id_hash`, `timestamp`, and `allowed`; partial index optimizes denial queries. **MongoDB 7:** Indexes on all query fields with schema validation. Both were deployed in Docker on the same hardware as Ethereum (Hardhat local network).

### C. Experimental Design

**Event Generation:** 100–10K synthetic events with 200 subjects, 1000 resources, 6 action types, and 80% allowed / 20% denied ratio.

**Query Patterns:** Four compliance-driven patterns: *By Subject* (user audit), *By Resource* (forensic analysis), *Time Range* (periodic reporting), and *All Denials* (security monitoring).

**Metrics:** Query latency (median, P90, P99 over 10 repetitions) and storage costs, including network overhead.

**Environment:** Hardhat 3.0, PostgreSQL 16, MongoDB 7 in Docker (Intel i7, 16GB RAM, SSD).

## III. Results and Discussion

ClaimGuard's blockchain achieves 47 tps with 20.2 ms insertion latency, comparable to SmartAccess (10–50 tps, 200–800 ms) while consuming only 26K gas versus 50-150K. PostgreSQL and MongoDB achieve 62× and 1000× higher throughput, respectively.

### A. Query Latency Analysis

Figure 1 compares query latency across all systems at 10K events. Indexed queries are 6-8× slower than in PostgreSQL due to JSON-RPC overhead. Time-range queries are 35× slower (323.8 ms vs. 9.2 ms) because block scans cannot leverage timestamp indexes. Denial queries are 41× slower since `allow` is not indexed on-chain. The performance gap widens with scale: at 10K events, scan-heavy patterns exceed 300 ms, compared to PostgreSQL's sub-10 ms response time. These results confirm that blockchain audit logs are well-suited to low-volume, high-assurance scenarios but require off-chain indexing for larger compliance workloads.

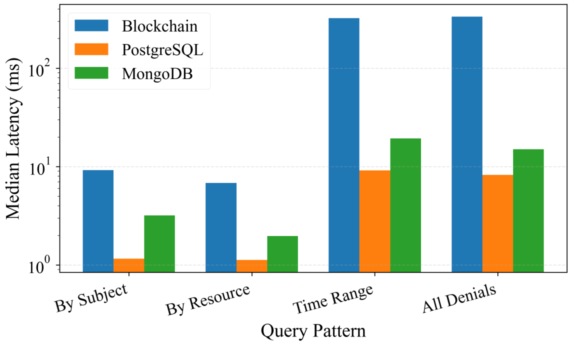

Fig. 1. Query latency comparison at 10K events. Indexed queries (By Subject, By Resource) show blockchain 6–8× slower than PostgreSQL. Scan-heavy queries (Time Range, All Denials) show blockchain 35-41× slower.

Blockchain audit logs are suitable for high-value transactions that require tamper-proof evidence; databases are preferable for high-frequency, cost-sensitive workloads. We recommend a hybrid approach: log critical decisions on-chain while routing routine logs to databases, reducing costs 10–100×.

### B. Comparison with Related Works

Table I compares ClaimGuard against related blockchain systems. While prior works focused on write throughput and access control latency, none evaluated audit log *query* performance against traditional databases. Each blockchain event costs ∼26K gas ($0.52 at 10 gwei, $2K/ETH). PostgreSQL uses ∼819 bytes/event; MongoDB ∼279 bytes. On-chain storage is orders of magnitude more expensive; 10K events cost ∼5,000 on mainnet versus pennies for databases.

TABLE I
Performance Comparison with Related Works

| System | Write Throughput (tps) | Insertion Latency (ms) | Query Latency (ms) | Gas Cost (per event) |
|---|---|---|---|---|
| SmartAccess [7] | 10–50 | 200–800 | – | 50–150K |
| Hybrid BCDB [5] | 100–500$^{\dagger}$ | – | – | – |
| Gürsoy et al. [9] | – | 400$^{\ddagger}$ | 35–750 | – |
| **This work** | 47 | 20.2 | 6.8–335 | 26K |

## IV. Conclusion

We present the first quantitative analysis of query performance on blockchain audit logs. Blockchain-indexed queries are 6–8× slower than PostgreSQL; scan-heavy queries are 35-41× slower, with gas costs orders of magnitude higher. These findings support hybrid architectures, routing critical decisions to blockchain while delegating routine logs to databases. Future work includes Layer 2 deployment and off-chain indexing.

## Acknowledgment

This work was partly supported by the Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201612, 25%), by the Priority Research Centers Program through the NRF funded by the MEST (2018R1A6A1A03024003, 25%), and by the MSIT, Korea, under the ITRC support program (IITP-2025-RS-2024-00438430, 25%) and by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (RS-2025-25431637, 25%).

## References

[1] D. Basin, S. Debois, and T. Hildebrandt, "On purpose and by necessity: Compliance under the gdpr," in *Financial Cryptography and Data Security*, 2018.

[2] A. U. Eneh, L. A. C. Ahakonye, J. M. Lee, and D.-S. Kim, "ClaimGuard: A Blockchain-Backed Access Control Gateway for Privacy-Preservation in Auto-Insurance Claims," in *2025 16th International Conference on Information and Communication Technology Convergence (ICTC)*, 2025.

[3] W. Li, Y. Feng, N. Liu, Y. Li, X. Fu, and Y. Yu, "A secure and efficient log storage and query framework based on blockchain," *Computer Networks*, vol. 252, p. 110683, 2024.

[4] A. Ali, A. Khan, M. Ahmed, and G. Jeon, "Bcals: Blockchain-based secure log management system for cloud computing," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, p. e4272, 2022.

[5] Z. Ge, D. Loghin, B. C. Ooi, P. Ruan, and T. Wang, "Hybrid blockchain database systems: design and performance," in *Proceedings of the VLDB Endowment*, vol. 15, no. 5, 2022, pp. 1092–1104.

[6] C. Yue, M. Zhang, C. Zhu, G. Chen, D. Loghin, and B. C. Ooi, "Veribench: Analyzing the performance of database systems with verifiability," in *Proceedings of the VLDB Endowment*, vol. 16, no. 9, 2023, pp. 2145–2158.

[7] M. T. De Oliveira, L. H. Reis, Y. Verginadis *et al.*, "Smartaccess: attribute-based access control system for medical records based on smart contracts," *IEEE Access*, vol. 10, pp. 117 836–117 854, 2022.

[8] F. Ullah, J. He, N. Zhu, A. Wajahat, A. Nazir, and S. Qureshi, "Blockchain-enabled ehr access auditing: Enhancing healthcare data security," *Heliyon*, vol. 10, no. 14, p. e34407, 2024.

[9] G. Gürsoy, C. M. Brannon, and M. Gerstein, "Using ethereum blockchain to store and query pharmacogenomics data via smart contracts," *BMC Medical Genomics*, vol. 13, no. 1, p. 74, 2020.