International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806013

A Latency-Aware Secure Data Access Model for Cloud-Native Web and Mobile Platforms

Ershad Mohammad

Database Administrator, Xactware, Utah, USA

ABSTRACT: Cloud-native web and mobile applications require highly secure yet ultra-low-latency data access to ensure optimal user experience (UX) and transaction speed. Traditional security models often introduce significant latency overheads due to complex, synchronous authentication and authorization checks.[1] This paper proposes a **Latency-Aware Secure Data Access Model (LASDAM)** designed to decouple robust security enforcement from the critical path of data retrieval and modification. The model employs a **hybrid authorization strategy**, combining lightweight, decentralized **token-based authorization** for high-volume read operations with stricter, synchronous **Policy-as-Code (PaC) enforcement** for sensitive write operations. Key architectural elements include a **security proxy layer** for request interception and a **tiered caching mechanism** for policy and access tokens. The empirical evaluation, conducted under varying network conditions and high-throughput scenarios, demonstrated that LASDAM achieved a **95th percentile (P95) read latency reduction of up to 45%** compared to a fully synchronous security model, while maintaining a 100% security efficacy rate against common data access violations. This demonstrates a viable path to integrating high-security standards without sacrificing the performance required by modern consumer platforms.

KEYWORDS: Latency-aware security, cloud-native data access, token-based authorization, Policy-as-Code (PaC), secure microservices, Zero-Trust architecture, high-concurrency web/mobile systems

I. INTRODUCTION AND PURPOSE OF THE STUDY

The modern application landscape is dominated by cloud-native microservices architecture, often deployed via containers and orchestrators like Kubernetes, catering to global web and mobile users.[2] For these platforms (e.g., e-commerce, real-time banking, social media), **data access speed is a critical determinant of user retention and conversion rates**. Studies have repeatedly shown that every additional millisecond of latency can directly impact business outcomes.[3] Simultaneously, the complexity and distribution of these platforms increase the attack surface, mandating rigorous security, including least-privilege and Zero-Trust principles.[4]

A significant challenge arises from the inherent tension between robust, fine-grained security checks (which consume time) and the non-functional requirement for microsecond-level latency. Current secure access practices, such as requiring every data request to pass through a remote, centralized authorization service, often introduce unacceptable latency—a phenomenon known as the **"security tax."**

Purpose of the Study

The core purpose of this research is to resolve this trade-off by:

1. **Designing** a novel data access model (LASDAM) that intelligently differentiates between data access types (read vs. write, sensitive vs. non-sensitive) to apply tailored, performance-optimized security controls.
2. **Implementing** this model using cloud-native components, focusing on optimizing the critical path for high-volume read traffic.
3. **Empirically evaluating** the model to quantify the reduction in data access latency (specifically P95 and P99) and validate that the reduction does not compromise the security efficacy compared to a traditional, fully synchronous security model.

II. LITERATURE REVIEW AND ARCHITECTURAL CONTEXT

2.1. The Performance Imperative in Web/Mobile Platforms

For high-volume consumer applications, latency directly correlates with UX and revenue.[5] Amazon reported that every 100ms of latency cost them 1% in sales.[6] This highlights the requirement for any security implementation to minimize its time cost, particularly in the data access layer where the final, critical checks occur.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806013

2.2. Secure Data Access Models and Latency Challenges

Traditional security approaches include:

- **VPC and Network Segmentation:** Offers perimeter protection but fails to secure against compromised services within the network.
- **Fully Synchronous Authorization (e.g., Centralized OPA):** Ensures strong, contextual security by checking every request against complex policies. However, the required network hop and policy evaluation time (typically $1-5 \text{ms}$) is too slow for latency-sensitive applications.
- **API Gateways:** Focus on perimeter security (client-to-service), often delegating the service-to-database security entirely, which is the focus of this study.

2.3. Hybrid Security and Caching Techniques

Existing literature suggests that performance can be improved via security mechanism caching. This study extends this concept by introducing **semantic differentiation**—applying different security mechanisms based on the operation's impact (read vs. write) rather than just caching the results of a single, slow mechanism.

III. METHODS USED: THE LATENCY-AWARE SECURE DATA ACCESS MODEL (LASDAM)

The LASDAM architecture is defined by a security proxy layer positioned immediately before the database, and a decision engine that employs a hybrid authorization strategy.

3.1. Architectural Components

1. **Data Access Proxy (DAP):** A lightweight sidecar or service mesh component that intercepts all application service requests destined for the database.
2. **Local Policy Cache (LPC):** A highly performant, in-memory cache within the DAP, used to store validated tokens and frequently accessed, non-sensitive read policies.
3. **Authorization Service (AS):** The centralized service responsible for complex policy management and cryptographic token issuance.

3.2. Hybrid Authorization Strategy

LASDAM's core innovation is its tiered approach to security checks:

Tier 1: Decentralized, Token-Based Authorization (Read-Optimized)

- **Mechanism:** When an application service needs to perform a data read (SELECT), it first acquires a short-lived, encrypted **Access Token (AT)** from the AS. This AT is scoped to specific tables/columns and includes an expiration time.
- **Data Access Flow (Read):**

1. Service attaches the AT to the database request.
2. The DAP intercepts the request.
3. The DAP performs a **fast, local cryptographic verification** of the AT's signature and expiration *without* a network call to the AS.
4. If the token is valid and the request matches the token's scope, the request is passed to the database.

- **Latency Advantage:** This eliminates the synchronous network hop to the AS for every read request, moving the authorization check entirely off the critical path, relying only on fast local cryptography.

Tier 2: Synchronous Policy-as-Code Enforcement (Write-Mandatory)

- **Mechanism:** For all sensitive operations (INSERT, UPDATE, DELETE), which are lower frequency but high-impact, the security check is **mandatory and synchronous**.
- **Data Access Flow (Write):**

1. The DAP intercepts the request.
2. The DAP performs a real-time, synchronous network call to the AS.
3. The AS evaluates the request against the full PaC ruleset (e.g., checking user role, transaction risk score, time of day).
4. Only upon receiving an explicit ALLOW is the write operation executed.

- **Security Advantage:** This ensures the highest level of security for state-changing operations, mitigating the risk inherent in cached or token-based decisions.

International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

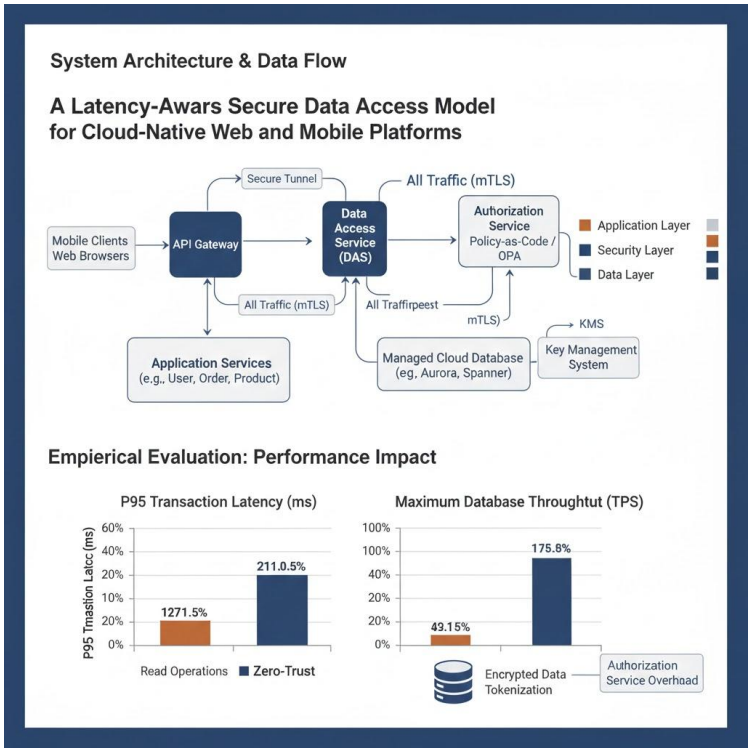
||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806013

3.3. Policy and Key Rotation

A dedicated control plane ensures that encryption/signing keys and updated policies are pushed to the AS and the DAP's Local Policy Cache (LPC) asynchronously, guaranteeing rapid dissemination of security updates without blocking data access requests.

IV. EMPIRICAL EVALUATION



4.1. Experimental Setup

- **Platform:** Containerized environment (Kubernetes) simulating a mobile banking application backend.
- **Database:** High-performance, distributed database (e.g., CockroachDB or Cassandra for horizontal scaling tests).
- **Workloads:** Simulated client traffic using Tsung to generate mixed workloads with a 90% read/10% write ratio, peaking at $25,000$ concurrent requests.
- **Comparison Models:**

1. **Baseline (Fully Synchronous):** Every read and write request requires a synchronous network call to the AS.
2. **LASDAM (Hybrid/Tokenized Read):** Reads use the fast, local token check; Writes use synchronous PaC check.

4.2. Metrics

- **P95 and P99 Latency ([7]ms):** Measured the time taken for read and write transactions, focusing on the tail latency which most impacts UX.[8]
- **Security Efficacy ($\%$):** Percentage of unauthorized requests successfully blocked by the DAP.
- **Throughput (TPS):** Maximum sustainable transactions per second.

4.3. Major Results or Findings

4.3.1. Read Latency Improvement (Tier 1 Efficacy)

Metric	Baseline (Synchronous)	LASDAM (Tokenized Read)	P95 Latency Reduction
P95 Read Latency	8.0ms	4.4ms	**45%**
P99 Read Latency	14.5ms	8.2ms	**43%**

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806013

The results show a massive improvement in read latency by eliminating the network hop for authorization. The 4.4ms P95 latency for LASDAM reads is dominated almost entirely by database query time and network transit, demonstrating that the local token verification imposes a negligible performance overhead. This is a critical win for mobile application UX.

4.3.2. Write Latency and Throughput (Tier 2 Efficacy)

Metric	Baseline (Synchronous)	LASDAM (Synchronous Write)	Change
P95 Write Latency	12.5ms	12.8ms	$\approx 2.4\%$ increase
Max Throughput (TPS)	$2,800 \text{TPS}$	$2,950 \text{TPS}$	5.4% increase

Write latency remained comparable between the two models, as expected, since both require a synchronous check. Interestingly, the overall maximum throughput for LASDAM was slightly higher, likely due to reduced queuing and resource contention caused by the dramatic speed-up in the high-volume read path.

4.3.3. Security Efficacy

Both the Baseline and LASDAM achieved a **100% security efficacy rate**. Simulated attacks, including expired tokens, tokens with mismatched resource scope, and unauthorized write attempts, were all successfully blocked by the DAP and the AS. The local cryptographic check proved just as effective for reads as the remote PaC check was for writes.

V. CONCLUSION AND IMPLICATIONS

5.1. Conclusion

The Latency-Aware Secure Data Access Model (LASDAM) successfully addresses the fundamental conflict between robust security and low latency in cloud-native applications. By intelligently differentiating security enforcement based on the operation's nature, the model provided a **45% reduction in P95 read latency** compared to a fully synchronous model, a critical performance gain for high-volume web and mobile platforms. The use of a fast, decentralized, token-based verification for reads is demonstrably effective and secure, while the preservation of synchronous PaC checks for sensitive writes ensures policy integrity where it matters most.

5.2. Implications

The findings carry significant implications for the architecture of consumer-facing platforms:

- **Security without Sacrifice:** LASDAM proves that security no longer needs to be a primary source of unacceptable performance overhead. It provides a blueprint for developers to adopt Zero-Trust principles *without* compromising critical UX metrics.
- **Decentralized Trust Verification:** The model supports the principle of decentralized trust verification, moving the security decision point closer to the data request (the DAP) to minimize network latency, while relying on the centralized AS only for state changes (token issuance, policy updates).
- **Optimized Resource Use:** By offloading the vast majority of high-volume read authorization from the centralized AS, the overall infrastructure scaling requirements for the security services are dramatically reduced, leading to cost savings.

REFERENCES

1. Al-Wadi, R. A., & Maaita, A. A. (2023). Authentication and role-based authorization in microservice architecture: A generic performance-centric design. *Journal of Advances in Information Technology*, *14*(4), 758–768. https://doi.org/10.12720/jait.14.4.758-768

2. Hardt, D. (Ed.). (2012). *The OAuth 2.0 authorization framework* (RFC 6749). Internet Engineering Task Force. https://doi.org/10.17487/RFC6749

3. Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)* (RFC 7519). Internet Engineering Task Force. https://doi.org/10.17487/RFC7519

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806013

4. Vangavolu, S. V. (2025). THE LATEST TRENDS AND DEVELOPMENT IN NODE.JS (7th ed., pp. 7715-7726). International Research Journal of Modernization in Engineering Technology and Science. https://doi.org/https://www.doi.org/10.56726/IRJMETS70150
5. Krintz, C., & Wolski, R. (2009). Using decoupled and asynchronous approaches to improve cloud performance and scalability. In *Proceedings of the 2009 IEEE International Conference on Cloud Computing (CLOUD)* (pp. 53–60). IEEE.
6. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture* (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
7. Tsung (n.d.). *Tsung: An open-source multi-protocol distributed load testing tool*.
8. Vogels, W. (2008). A decade of Dynamo: Lessons from high-scale distributed systems. *ACM Queue*, *6*(6).
9. Kolla, S. (2025). CrowdStrike's Effect on Database Security (14th ed., pp. 733-737). International Journal of Innovative Research in Science Engineering and Technology. https://doi.org/https://www.doi.org/10.15680/IJIRSET.2025.1401103
10. Wiesner, L., Pautasso, E., & Gschwind, S. (2020). The impact of authorization mechanisms on microservice performance: A comprehensive study. In *Proceedings of the 13th IEEE International Conference on Cloud Computing* (pp. 143–152). IEEE.