IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

A Framework for Zero Trust Architecture in Cloud-Native Ecosystems: Authentication, Anomaly Detection, and Integration

Pavan R. Warule[1], Prajakta Muntode[2], Arati Wakare[3] and Prof. S. A. Wanave[4]
Department of Computer Engineering[1-4]
Adsul's Technical Campus, Ahilyanagar, India
Corresponding author: Pavan R. Warule (pavanwarule4961@gmail.com)
Savitribai Phule Pune University, Pune, India

Abstract: *The enterprise adoption of dynamic, distributed cloud-native architectures has rendered traditional perimeter-based security models obsolete. The implicit trust granted to entities "inside" the network creates a critical vulnerability to lateral movement attacks in microservice environments. Zero Trust Architecture (ZTA), founded on the principle of "never trust, always verify," emerges as the necessary security paradigm for this new ecosystem. This paper proposes a comprehensive, intelligent ZTA framework designed to address the unique challenges of microservices, containers, and serverless functions. The framework is built on three pillars:*
(1) a lightweight, scalable authentication stack utilizing automated workload identity via SPIFFE/SPIRE and performant, sidecarless service mesh enforcement planes; (2) an AI-driven continuous validation engine employing machine learning for behavioral anomaly detection and Explainable AI (XAI) for operational trust; and (3) a domain-specific integration model for applying ZTA principles to the unique constraints of 5G and IoT ecosystems. This work analyzes the critical performance trade-offs of modern ZTA implementations and presents a viable architectural path toward a secure, scalable, and observable cloud-native future..

Keywords: *Zero Trust Architecture.*

I. INTRODUCTION

The modern enterprise is undergoing a fundamental architectural transformation, migrating from static, monolithic applications to dynamic, cloud-native paradigms built on microservices, containers, and serverless functions. This shift is driven by the business demand for agility, resilience, and rapid, independent scalability. However, this architectural evolution introduces a new and severe threat model that traditional security assumptions are incapable of addressing.

Legacy security is predicated on a perimeter-based, "castle-and-moat" model. It fastidiously inspects "north-south" traffic (client-to-server) at the network edge, but implicitly trusts traffic originating from within the "trusted" local network.[3] This model fails completely in a cloud-native context for two primary reasons. First, the infrastructure is *ephemeral*; workloads are created and destroyed in seconds, rendering static, IP-based firewall rules obsolete. Second, application traffic is dominated by "east-west" (service-to-service) communication, which traditional perimeter defenses are blind to.[3]

In container orchestration platforms like Kubernetes, the network is often flat by default, allowing unrestricted pod-to-pod communication. This creates the ideal environment for lateral movement. An attacker who compromises a single, low-privilege microservice (e.g., Service A) gains an internal foothold from which they can attack other services (e.g., Service B) that may be misconfigured to implicitly trust all internal traffic. This simple misconfiguration can lead to a catastrophic breach. The architectural choice to adopt microservices, therefore, *causally creates* the security gap that invalidates the perimeter model.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

The logical and necessary successor for this new paradigm is the Zero Trust Architecture (ZTA). Defined formally in NIST Special Publication 800-207, ZTA moves defenses from static, network-based perimeters to focus on users, assets, and resources.[7] The core tenets of ZTA are:

- **Never Trust, Always Verify:** All requests are treated as potential threats, and implicit trust is eliminated.
- **Assume Breach:** The network is assumed to be hostile.
- **Enforce Least Privilege:** Access is granted on a per-session, per-resource basis.
- **Secure All Communication:** All communication is secured regardless of network location.

This represents a fundamental shift from *network-based* controls (i.e., "what is your IP address?") to *identity-based* controls (i.e., "who are you, and are you authorized for this specific request?"). This paper's primary contribution is a unified, performant ZTA framework that addresses three critical research gaps. We present (1) a lightweight authentication model that solves the performance overhead problem, (2) an AI/XAI-based continuous validation loop, and (3) an analysis of ZTA integration into the adjacent domains of 5G and IoT.

TABLE I: TRADITIONAL PERIMETER VS. ZTA IN CLOUD-NATIVE ENVIRONMENTS

Attribute	Traditional Perimeter Model	Zero Trust Architecture (ZTA)
Trust Assumption	Implicitly trust "inside" network	Explicitly verify all ("assume breach")
Enforcement Point	Network Edge (e.g., Firewall)	Per-resource / Per-session
Primary Traffic	North-South (Client-to-Server)	East-West and North-South
Key Vulnerability	Lateral Movement	Compromised Identity
Identity Basis	Network Location (IP Address)	Cryptographic, Attestable Identity

II. LIGHTWEIGHT AND SCALABLE WORKLOAD AUTHENTICATION

The foundation of ZTA is strong, verifiable identity. In a static world, this was handled with credentials like API keys, passwords, and long-lived certificates. In a dynamic cloud-native environment, this static credential model is an untenable liability. A single compromised key can function as a "skeleton key," granting an attacker vast access to cloud resources and enabling widespread lateral movement. Furthermore, the operational burden of managing, rotating, and auditing these secrets at scale is immense, often leading to insecure practices that violate ZTA principles. This necessitates a "secretless" paradigm based on automated, short-lived, and verifiable workload identity.

A modern ZTA authentication stack can be architected in three distinct, decoupled layers:

A. Identity Plane: SPIFFE/SPIRE

The foundation for identity-based ZTA is a universal, platform-agnostic, and cryptographically verifiable identity. The **Secure Production Identity Framework for Everyone (SPIFFE)** provides an open-source standard for this. Its software implementation, **SPIRE**, acts as the identity provider. The SPIRE server and agents work by:

Attesting a workload's identity at runtime. For example, it can verify a workload's process ID, kernel attributes, or its Kubernetes ServiceAccount token.

Issuing a **SPIFFE Verifiable Identity Document (SVID)** upon successful attestation. SVIDs are short-lived, automatically rotated cryptographic documents that represent the workload's identity.

Providing SVIDs to workloads, which can take two forms: **X.509 certificates** or **JWT-SVIDs** (JSON Web Tokens).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

B. Authentication Mechanism: mTLS and JWT

Workloads use their SVIDs to authenticate with other services.

Mutual TLS (mTLS): This is the primary mechanism for securing the *channel* between two workloads (e.g., microservice-to-microservice). Using X.509-SVIDs, both the client and server cryptographically prove their identity to each other, and all traffic in transit is encrypted.[14]

JWT-SVIDs: This mechanism is used to authenticate a *message* or API call. A workload presents a signed JWT-SVID to a resource, which validates its signature and identity. This is critical for scenarios where mTLS is not feasible, such as requests from external users, through L7 load balancers, or for stateless authentication.[14]

C. Enforcement Plane: Service Mesh

While SPIFFE/SPIRE provides the *identity*, a **service mesh** (such as Istio or Linkerd) acts as the *enforcement plane*.[15] The mesh injects lightweight proxies (either as sidecars or using a node-based agent) that intercept all network traffic to and from a workload. These proxies, often integrated with SPIRE, transparently handle mTLS establishment and termination using the workload SVIDs. This achieves two core ZTA goals:

Secure Communication: All service-to-service traffic is authenticated and encrypted by default, transparently to the application.

Micro-segmentation: The mesh can enforce fine-grained, identity-based authorization policies, such as "Service 'checkout' is allowed to call Service 'payment', but Service 'logging' is not".[9]

D. Addressing the Performance vs. Security Trade-off

A significant barrier to ZTA adoption has been the perceived performance overhead of these enforcement proxies.[20] Encrypting all traffic and performing identity validation on every call introduces latency. However, this performance cost is not a fundamental flaw of ZTA, but rather an *implementation artifact* of first-generation, sidecar-based service meshes.

Recent performance analysis demonstrates a clear architectural evolution in ZTA enforcement, as shown in Table II. While enabling mTLS on a baseline Kubernetes cluster adds only 3% latency, a standard (Gen 1) Istio sidecar implementation can add a prohibitive 166% latency, largely due to extra features like HTTP parsing. In contrast, lightweight (Gen 2) sidecars like Linkerd, and sidecarless (Gen 3) architectures like Istio Ambient, have drastically reduced this overhead to 33% and 8%, respectively.[21]

This data demonstrates that the ZTA enforcement plane is rapidly evolving to solve the overhead problem, making it viable for production, at-scale deployments. Furthermore, the decoupling of the *identity plane* (SPIFFE) from the *enforcement plane* (mesh) is a critical architectural pattern. It allows a single, universal identity provider to serve heterogeneous enforcement points (e.g., Istio, Linkerd, API gateways) across a multi-cloud and hybrid-cloud environment, enabling a truly federated and scalable ZTA.[11]

TABLE II: PERFORMANCE OVERHEAD OF MTLS IN SERVICE MESH (AT 3,200 RPS)

Architecture	Proxy Model	P99 Latency Increase (vs. baseline)	Memory Overhead (per pod/node)
Baseline-mTLS	N/A	$\sim$3%	$\sim$100%
Istio (Standard)	Sidecar	$\sim$166%	$\sim$255MB (client)
Linkerd	Lightweight Sidecar	$\sim$33%	$\sim$60MB (client)
Istio Ambient	Sidecarless (Ztunnel)	$\sim$8%	$\sim$26MB (node-level)

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

III. AI-DRIVEN CONTINUOUS VALIDATION AND ANOMALY DETECTION

The ZTA tenet of "never trust, always verify" implies that authentication cannot be a static, one-time event. Even if a workload (Service A) authenticates successfully to Service B via mTLS, that trust must be *continuously reassessed*.[9] Service A could be compromised *after* authentication and begin behaving maliciously. ZTA therefore demands continuous validation based on real-time behavior.[25]

A. The Role of AI/ML in Continuous Validation

At cloud-native scale, with thousands of ephemeral workloads generating millions of events per second, manual or rule-based validation is impossible. This continuous validation loop can only be achieved by applying Artificial Intelligence (AI) and Machine Learning (ML).[27] The service mesh, in addition to being an *enforcement* plane, is also a rich *telemetry source*, providing detailed logs, traces, and metrics on all east-west traffic. This data is the ideal input for ML models.

The primary ML paradigms used for this task include :

Supervised Learning: Models (e.g., Random Forests, SVMs) are trained on pre-labeled datasets containing examples of both normal and anomalous (attack) traffic. This is effective for detecting known threats but struggles with novel, zero-day attacks.

Unsupervised Learning: Models (e.g., K-Means clustering, PCA) are fed unlabeled data to build a *baseline* of normal behavior. This is highly effective at detecting *anomalies*—deviations from the norm—which may indicate a compromised workload, an insider threat, or lateral movement.[27]

Reinforcement Learning (RL): An RL agent can be trained to dynamically adjust security policies (e.g., block a connection) based on environmental feedback, learning optimal threat responses over time in a trial-and-error process.

This creates a powerful, closed-loop framework: (1) The service mesh *observes* workload behavior (telemetry). (2) An AI/ML engine *analyzes* this behavior in real-time to generate a dynamic trust score. (3) This trust score is fed *back* to the mesh's Policy Engine, which can then take adaptive action—such as revoking access, forcing re-authentication, or quarantining the suspicious workload.

B. Bridging the Trust Gap with Explainable AI (XAI)

A significant barrier to adopting AI in security operations is the "black box" problem.[31] If an AI model flags a critical production service as "anomalous" and revokes its access, security operators *must* understand *why* that decision was made. Without this transparency, the AI-driven system is operationally untrustworthy.[32]

Explainable AI (XAI) is the critical component that "bridges this trust gap" by providing interpretability to the AI's decisions.[32] For ZTA, XAI is not a "nice-to-have" but an *essential operational requirement*. Specific XAI techniques used to make anomaly detection models transparent include [33]:

SHAP (SHapley Additive exPlanations): Provides *global* explanations by calculating the contribution of each feature to the model's overall prediction. It can identify which factors (e.g., unusual request time, anomalous data payload size, novel port usage) are the strongest indicators of malicious behavior across the system.[33]

LIME (Local Interpretable Model-agnostic Explanations): Provides *local* explanations, justifying the model's decision for a *single, specific request* that was flagged as anomalous. This allows an operator to immediately see, for example, "This request was denied because it originated from a new geographic location *and* attempted to access a sensitive database endpoint, a combination never seen before".[33]

By integrating ZTA + AI + XAI, the framework moves from static authentication to a truly adaptive, continuously validated, and *operationally transparent* security posture.

IV. ZTA INTEGRATION FOR 5G AND IOT ECOSYSTEMS

The principles of the ZTA framework—decoupled identity, fine-grained enforcement, and continuous validation—are not limited to traditional cloud data centers. They form a *meta-framework* that can be adapted to other complex, distributed domains, most notably 5G and the Internet of Things (IoT).

A. ZTA for 5G Networks

The 5G architecture itself is cloud-native, featuring a disaggregated control/user plane, Software-Defined Networking (SDN), Network Function Virtualization (NFV), Multi-Access Edge Computing (MEC), and network slicing.[34] This

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

"open architecture" [35] exposes a vast new attack surface, rendering perimeter security as ineffective for 5G as it is for microservices.[34]

The solution is to apply ZTA principles *within* the 5G core. The enforcement plane is adapted from a service mesh to the native 5G functions:

A proposed layered ZTA framework integrates **Policy Decision Points (PDPs)** and **Policy Enforcement Points (PEPs)** directly into key 3GPP network functions, such as the Access and Mobility Management Function (AMF), the Session Management Function (SMF), and the User Plane Function (UPF).[34]

This allows for continuous, identity-aware authentication and micro-segmentation *of the 5G network functions themselves*.

Furthermore, AI/ML is used to create an **intelligent ZTA (i-ZTA)**, enabling real-time anomaly detection and automated responses to threats within the tactical 5G network.[36]

B. Lightweight ZTA for IoT Environments

IoT ecosystems present a unique challenge: applying ZTA to billions of physically vulnerable and *resource-constrained* devices that lack the power and computational capacity for heavyweight cryptographic operations or security agents.[39]

For this domain, a *lightweight* ZTA framework is required, modifying the enforcement and validation mechanisms:

Lightweight Trust Validation: Instead of requiring complex computations on the device, trust can be validated by a sensor coordinator using efficient primitives like **Merkle Trees**. The device's trust score is stored and continuously updated in the cloud, and access is revoked if it falls below a threshold.[40]

Offloaded, Fine-Grained Access Control: Computationally intensive tasks are outsourced to the cloud without sacrificing confidentiality. This is achieved using **Attribute-Based Encryption (ABE)**, such as **Key-Policy ABE (KP-ABE)**. A data owner (e.g., a sensor coordinator) can encrypt data with a fine-grained policy (e.g., "Role=Doctor AND Geo=Hospital"). The cloud can manage key generation and store this ciphertext, but *cannot decrypt it*. Only users whose attributes match the policy can access the data.[40]

In both 5G and IoT, the *meta-framework* holds: a universal identity is established, an enforcement point validates that identity against a policy, and trust is continuously monitored. Only the *implementation* of the enforcement plane (Service Mesh vs. 5G Core Function vs. Cloud-ABE) is adapted to the specific domain's constraints.

V. IMPLEMENTATION FRAMEWORK FOR CONTAINERIZED ENVIRONMENTS

The most common implementation of cloud-native applications is in containerized environments orchestrated by Kubernetes.[5] For this domain, a mature, standardized path to ZTA implementation exists, codified by the National Institute of Standards and Technology (NIST).

A critical distinction exists between NIST's ZTA publications:

NIST SP 800-207 defines the high-level *principles* and abstract architecture of ZTA.[7] It is the "what" and "why."

The NIST SP 800-204 Series (including 204, 204A, 204B, 204C) provides the specific, practical *implementation strategy* for securing microservice-based applications.[43] It is the "how."

Crucially, **NIST SP 800-204A** designates an official *reference platform* for implementing ZTA for microservices: **Kubernetes as the orchestrator and the Istio service mesh as the security kernel**.[42] This validates the approach of using a service mesh as the core ZTA enforcement plane, moving it from a "good idea" to a government-backed standard. This reference platform addresses gaps in native Kubernetes, such as its lack of mTLS, default-insecure communication, and L3-only network policies.[42]

Based on this reference platform and the concepts in this paper, a phased implementation model for ZTA in Kubernetes emerges. ZTA is not a single product to be installed, but a *journey* of architectural maturity.[46] This 5-layer model provides a clear roadmap for that journey:

Layer 1 (Baseline Identity): Utilize Kubernetes ServiceAccounts as the foundational, platform-level identity for pods.

Layer 2 (Cryptographic Identity): Deploy SPIRE to attest these ServiceAccounts and automatically issue short-lived, rotated SVIDs, establishing a strong, universal cryptographic identity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

Layer 3 (Enforcement & Micro-segmentation): Deploy a service mesh (e.g., lightweight Linkerd or sidecarless Istio Ambient). Configure it to consume the SVIDs to enforce cluster-wide mTLS (strict mode) and apply a "deny-by-default" AuthorizationPolicy.

Layer 4 (Continuous Validation): Feed the service mesh's rich observability data (logs, traces) into an AI/ML anomaly detection engine (as described in Section III). Use XAI tools like SHAP and LIME for operator visibility and trust.[33]

Layer 5 (Adaptive Policy): Close the loop. Use the AI's dynamic trust score to programmatically update the mesh's AuthorizationPolicy via its API, creating a fully adaptive, self-healing security posture.

VI. OPEN CHALLENGES AND FUTURE RESEARCH

Despite the maturation of ZTA frameworks, significant challenges and open research questions remain. Adopting this architecture introduces new complexities that must be addressed.

A. Key Challenges

Performance Overhead: While lightweight (Gen 3) architectures show promise, the 8-33% latency overhead may still be unacceptable for high-frequency trading, real-time industrial controls, or other ultra-low-latency applications.[22]

Policy Management at Scale: The promise of "fine-grained" authorization becomes an operational nightmare at scale. Managing, auditing, and debugging authorization policies for thousands of rapidly-evolving microservices is exponentially complex, creating a high risk of "policy drift" and misconfiguration.[20]

Legacy and Hybrid Cloud Integration: Most enterprises are not 100% cloud-native. Integrating ZTA principles with legacy monolithic systems (which cannot easily run a proxy) and maintaining a consistent identity and policy framework across heterogeneous multi-cloud and on-premise environments remains a significant hurdle.[47]

B. Future Research Directions

This paper identifies several critical areas for future research to address these challenges:

AI-Driven Policy Orchestration: Moving beyond AI for *detection* to using AI for *generation*. Future systems could use ML to analyze observed traffic patterns and automatically *recommend* or *generate* the true, least-privilege AuthorizationPolicy for a service, solving the policy management problem.[47]

eBPF and Hardware Acceleration: Researching novel data planes that move enforcement out of the proxy and into the Linux kernel using eBPF. This could drastically reduce latency by bypassing the user-space proxy entirely.[50] Further research into offloading cryptographic operations to smartNICs could approach zero-latency enforcement.

Federated Trust and Identity: While SPIFFE supports federation, more research is needed on the standards and protocols for securely federating trust and identity across organizational boundaries and different cloud providers, enabling true multi-cloud ZTA.[24]

Lightweight, Explainable AI: Developing new XAI-enabled ML models that are computationally *lightweight* enough to run at the network edge, enabling real-time, explainable anomaly detection *on* resource-constrained IoT devices themselves.[52]

VII. CONCLUSION

Traditional, perimeter-based security models are fundamentally incompatible with the dynamic, ephemeral, and distributed nature of cloud-native applications.[3] The implicit trust at the core of this legacy model is the direct vector for the lateral movement attacks that plague modern systems.

This paper has proposed a comprehensive, intelligent Zero Trust Architecture framework designed for this new reality. This framework is:

Performant: It solves the critical adoption barrier of performance overhead by leveraging lightweight identity (SPIFFE) and modern, sidecarless (Gen 3) enforcement planes, which reduce latency by over 95% compared to first-generation sidecars.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

Intelligent: It moves beyond static authentication by creating a closed-loop system where AI/ML provides continuous behavioral validation , and XAI provides the operational trust and transparency necessary for adoption.[32]

Extensible: It provides a universal *meta-framework* (Identity, Enforcement, Validation) that can be adapted to the specific constraints of diverse ecosystems, including 5G [34] and IoT.[40]

The validation of this approach by the NIST SP 800-204 series, which specifies Kubernetes and service mesh as the reference platform for ZTA [42], signals an industry-wide consensus. Zero Trust is not a single product, but an architectural *strategy*. When implemented as a performant, intelligent, and extensible system, it provides the only viable and resilient security posture for the next generation of distributed computing.

APPENDIX

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments. Avoid expressions such as "One of us (S.B.A.) would like to thank...." Instead, write "F. A. Author thanks...." Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page.

REFERENCES

[1]. U.S. Department of Commerce, National Institute of Standards and Technology, "Zero Trust Architecture," *NIST Special Publication 800-207*, Aug. 2020.

[2]. U.S. Department of Commerce, National Institute of Standards and Technology, "Security Strategies for Microservices-based Application Systems," *NIST Special Publication 800-204*, Aug. 2019. [45]

[3]. U.S. Department of Commerce, National Institute of Standards and Technology, "Attribute-based Access Control for Microservices-Based Applications Using a Service Mesh," *NIST Special Publication 800-204B*, Oct. 2020. [42]

[4]. Z. Butcher, "NIST Standards for Zero Trust: The SP 800-204 Series," Tetrate, 2023. [42]

[5]. Al-Rubaie, A. Al-Ali, and A. Al-Ali, "Investigating the Performance Overhead of mTLS in Service Mesh Architectures," *arXiv:2411.02267 [cs.NI]*, Nov. 2024. [21]

[6]. SPIFFE.io, "SPIFFE and SPIRE Use Cases: Workload Authentication," SPIFFE Documentation.

[7]. P. P. de L. Fraga, "A Secure Lightweight Data Access Control Protocol for Cloud-Centric IoT Sensor Networks," *arXiv:2309.01293*, Sep. 2023. [40]

[8]. K. Ramezanpour, "A Layered Zero Trust Architecture for 5G Networks," *International Journal of Innovative Research in Engineering & Management*, vol. 11, no. 6, 2024. [34]

[9]. W. Leister, "Zero Trust Architecture: A Systematic Literature Review," *arXiv:2503.11659*, Mar. 2025. [39]

[10]. S. A. A. Shah, "Explainable AI (XAI) for Transparency in ZTA-based UAV Detection," *arXiv:2403.17093*, Mar. 2024. [33]

[11]. K. Sandhu, "AI-Powered Anomaly Detection in Zero Trust Network Architecture (ZTNA)," *Nanotechnology Perceptions*, vol. 20, no. S16, 2024.

[12]. R. K. Y. See, "Explainable AI in Anomaly Detection for Zero Trust Security: Bridging the Trust Gap," *ResearchGate*, 2024. [32]

[13]. T. M. Fernandez, "A Survey of Privacy-Enhancing and Trust-Centric Cloud-Native Security Techniques," *Electronics*, vol. 13, no. 12, 2024.

[14]. K. Ramezanpour, R. J. J. T. J. F. R. E. A. P. L. T. S. T., "Intelligent Zero Trust Architecture (i-ZTA) for 5G/6G Networks," *arXiv:2210.01739*, Oct. 2022. [36]

[15]. J. D. S. T. M. C. J. E. S., "Zero Trust Security for Multi-Cloud Microservices Environments," *International Journal of Multidisciplinary Research*, 2024. [49]

[16]. Palo Alto Networks, "What is Microsegmentation?" Cyberpedia, 2024. [4]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

[17]. B. Smith, "Cloud-Native Threats and Legacy Defenses," *DevOps Digest*, 2024.
[18]. A Jones, "10 Ways Microservices Create New Security Challenges," KongHQ Blog, 2024.
[19]. J. D. L. Cruz, "Securing Kubernetes: A Roadmap to NIST Compliance," *ResearchGate*, 2024.
[20]. H. Cabouly, "Implementing Zero Trust in Kubernetes with Istio Service Mesh," 2024.
[21]. W. Barker and W. T. Polk, "Ballot Resolution for SP 800-207, Zero Trust Architecture," *NIST CSRC*, Aug. 2020.
[22]. T. Larsson, "Performance Analysis of Istio Framework," *arXiv:2105.02334*, May 2021.
[23]. B. M. C. B. S., "Security of microservice applications: A systematic literature review," *PeerJ Computer Science*, vol. 8, 2022. [54]
[24]. Cloud-Native Threats, Legacy Defenses: The Hybrid Security Dilemma | DEVOPSdigest, accessed on November 4, 2025, https://www.devopsdigest.com/cloud-native-threats-legacy-defenses-the-hybrid-security-dilemma
[25]. What Is Microsegmentation? - Palo Alto Networks, accessed on November 4, 2025, https://www.paloaltonetworks.com/cyberpedia/what-is-microsegmentation
[26]. (PDF) Applying Zero Trust to Kubernetes Clusters - ResearchGate, accessed on November 4, 2025, https://www.researchgate.net/publication/392002631_Applying_Zero_Trust_to_Kubernetes_Clusters
[27]. Zero Trust Architecture - NIST Technical Series Publications, accessed on November 4, 2025, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
[28]. Zero Trust Maturity Model Version 2.0 - CISA, accessed on November 4, 2025, https://www.cisa.gov/sites/default/files/2023-04/CISA_Zero_Trust_Maturity_Model_Version_2_508c.pdf
[29]. What Is Zero Trust Architecture? Key Elements and Use Cases - Palo Alto Networks, accessed on November 4, 2025, https://www.paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-architecture
[30]. A Systematic Literature Review on the Implementation and Challenges of Zero Trust Architecture Across Domains - MDPI, accessed on November 4, 2025, https://www.mdpi.com/1424-8220/25/19/6118
[31]. SP 800-207A, A Zero Trust Architecture Model for Access Control in ..., accessed on November 4, 2025, https://csrc.nist.gov/pubs/sp/800/207/a/final
[32]. SPIRE Use Cases - SPIFFE, accessed on November 4, 2025, https://spiffe.io/docs/latest/spire-about/use-cases/
[33]. Zero trust network security in Kubernetes with the service mesh, accessed on November 4, 2025, https://www.buoyant.io/zero-trust-in-kubernetes-with-linkerd
[34]. Extend ZTNA with external authorization and serverless computing - Cloudflare Docs, accessed on November 4, 2025, https://developers.cloudflare.com/reference-architecture/diagrams/sase/augment-access-with-serverless/
[35]. Enhancing Security in ASP.NET Core Applications: Implementing Oauth, JWT, and Zero-Trust Models - | International Journal of Innovative Science and Research Technology, accessed on November 4, 2025, https://www.ijisrt.com/assets/upload/files/IJISRT25MAR1677.pdf
[36]. The Istio service mesh, accessed on November 4, 2025, https://istio.io/latest/about/service-mesh/
[37]. [Guide] Implementing Zero Trust in Kubernetes with Istio Service ..., accessed on November 4, 2025, https://www.reddit.com/r/kubernetes/comments/1o7d4kh/guide_implementing_zero_trust_in_kubernetes_with/
[38]. Implementing Zero Trust Architecture in Microservices: An In-Depth Guide - Medium, accessed on November 4, 2025, https://medium.com/@platform.engineers/implementing-zero-trust-architecture-in-microservices-an-in-depth-guide-a66417447621
[39]. Technical Report: Performance Comparison of Service Mesh Frameworks: the MTLS Test Case - arXiv, accessed on November 4, 2025, https://arxiv.org/html/2411.02267v1
[40]. Performance Analysis of Zero-Trust multi-cloud - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2105.02334

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

[41]. Technical Report: Performance Comparison of Service Mesh ... - arXiv, accessed on November 4, 2025, https://arxiv.org/abs/2411.02267

[42]. Establishing Workload Identity for Zero Trust CI/CD: From Secrets to SPIFFE-Based Authentication - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2504.14760

[43]. [2410.18291] Enhancing Enterprise Security with Zero Trust Architecture - arXiv, accessed on November 4, 2025, https://arxiv.org/abs/2410.18291

[44]. Enterprise-Grade Security for the Model Context Protocol (MCP): Frameworks and Mitigation Strategies - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2504.08623

[45]. AI-Powered Anomaly Detection in Zero Trust Environments: A ..., accessed on November 4, 2025, https://nano-ntp.com/index.php/nano/article/download/5083/4027/9921

[46]. Dynamic Trust in Cloud Environments : Transforming Enterprise Security Models Through Zero Trust - IEEE Chicago Section, accessed on November 4, 2025, https://ieeechicago.org/dynamic-trust-in-cloud-environments-transforming-enterprise-security-models-through-zero-trust/

[47]. A Comprehensive Survey of Privacy-Enhancing and Trust-Centric ..., accessed on November 4, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC12030732/

[48]. Bridging the Gap: Using AI to Operationalize Zero Trust in Multi-Cloud Environments, accessed on November 4, 2025, https://cloudsecurityalliance.org/blog/2025/05/02/bridging-the-gap-using-ai-to-operationalize-zero-trust-in-multi-cloud-environments

[49]. [2509.00069] AnomalyExplainer Explainable AI for LLM-based anomaly detection using BERTViz and Captum - arXiv, accessed on November 4, 2025, https://arxiv.org/abs/2509.00069

[50]. Explainable AI in Anomaly Detection for Zero Trust Security ..., accessed on November 4, 2025, https://www.researchgate.net/publication/391327263_Explainable_AI_in_Anomaly_Detection_for_Zero_Trust_Security_Bridging_the_Trust_Gap

[51]. Enhancing UAV Security Through Zero Trust Architecture: An ... - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2403.17093

[52]. Zero Trust Architecture for 5G Networks - IJIRMPS, accessed on November 4, 2025, https://www.ijirmps.org/papers/2024/6/232707.pdf

[53]. (PDF) Toward Zero Trust Security IN 5G OPEN ARCHITECTURE NETWORK SLICES, accessed on November 4, 2025, https://www.researchgate.net/publication/377334836_Toward_Zero_Trust_Security_IN_5G_OPEN_ARCHITECTURE_NETWORK_SLICES

[54]. Intelligent zero trust architecture for 5G/6G networks - ANDRO Computational Solutions, accessed on November 4, 2025, https://www.androcs.com/wp/wp-content/uploads/2023/08/RamezanpourZTA22.pdf

[55]. Enabling a Zero Trust Architecture in a 5G-enabled Smart Grid - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2210.01739

[56]. (PDF) Intelligent Zero Trust Architecture for 5G/6G Tactical Networks: Principles, Challenges, and the Role of Machine Learning - ResearchGate, accessed on November 4, 2025, https://www.researchgate.net/publication/351342318_Intelligent_Zero_Trust_Architecture_for_5G6G_Tactical_Networks_Principles_Challenges_and_the_Role_of_Machine_Learning

[57]. Zero Trust Architecture: A Systematic Literature Review - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2503.11659

[58]. Zero Trust Real-Time Lightweight Access Control Protocol for ... - arXiv, accessed on November 4, 2025, https://arxiv.org/pdf/2309.01293

[59]. Zero Trust Architecture: A Systematic Literature Review - arXiv, accessed on November 4, 2025, https://arxiv.org/html/2503.11659v2

[60]. NIST Standards for Zero Trust: the SP 800-204 Series - Tetrate, accessed on November 4, 2025, https://tetrate.io/blog/nist-standards-for-zero-trust-the-sp-800-204-series

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

[61]. SP 800-204, Security Strategies for Microservices-based Application Systems | CSRC, accessed on November 4, 2025, https://csrc.nist.gov/pubs/sp/800/204/ipd

[62]. SP 800-204, Security Strategies for Microservices-based Application Systems | CSRC, accessed on November 4, 2025, https://csrc.nist.gov/pubs/sp/800/204/final

[63]. Security Strategies for Microservices-based Application Systems, accessed on November 4, 2025, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-204.pdf

[64]. Zero trust security model implementation in Kubernetes-based cloud infrastructure - International Journal of Science and Research Archive, accessed on November 4, 2025, https://ijsra.net/sites/default/files/IJSRA-2021-0007.pdf

[65]. A Systematic Literature Review on the Implementation and Challenges of Zero Trust Architecture Across Domains - PubMed Central, accessed on November 4, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC12526847/

[66]. Challenges in Implementing Scalable Zero Trust with Micro-Segmentation in Hybrid Multi-Cloud Environments : r/cybersecurity - Reddit, accessed on November 4, 2025, https://www.reddit.com/r/cybersecurity/comments/1gegxsi/challenges_in_implementing_scalable_zero_trust/

[67]. Implementing Zero Trust Security in Multi-Cloud Microservices Platforms: A Review and Architectural Framework - International Journal of Advanced Multidisciplinary