



(RESEARCH ARTICLE)



ENTERPRISE-WIDE ZERO-TRUST AUTOMATION USING AI AND CLOUD-NATIVE MICROSERVICES

Ravindrakumar Prajapati *

Anaplan Manager Relanto Inc Austin, USA.

* Corresponding Author; Email: prajapati.ravindra@outlook.com

World Journal of Advanced Research and Reviews, 2025, 26(03), 2901–2913

Publication history: Received on 04 May 2025; revised on 24 June 2025; accepted on 29 June 2025

Article DOI: <https://doi.org/10.30574/wjarr.2025.26.3.2281>

Copyright © 2025 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

The rapid adoption of cloud computing, distributed applications, APIs, and microservices has expanded the enterprise attack surface and exposed limitations in conventional perimeter-based and static access-control mechanisms. This research proposes a Risk-Adaptive Zero-Trust Automation (RA-ZTA) framework that integrates artificial intelligence, continuous identity verification, dynamic policy enforcement, cloud-native microservices, continuous monitoring, and automated security remediation into a unified enterprise-wide security architecture. The proposed framework treats every user, device, application, API, and workload as potentially untrusted and continuously evaluates access requests using contextual attributes such as identity, device posture, network characteristics, behavioral patterns, workload activity, and historical security events. An AI-based risk assessment engine generates dynamic risk scores that are supplied to a policy and decision engine for adaptive authorization. The framework is implemented using cloud-native principles, containerized microservices, service-mesh security, mutual TLS, Policy-as-Code, real-time telemetry, and automated response mechanisms. The experimental evaluation compares the proposed approach with conventional RBAC, Static Zero Trust, ML-Based Zero Trust, and AI-enabled microservice security approaches using detection, response, and scalability metrics.

The experimental results demonstrate that the proposed RA-ZTA framework achieves 97.26% accuracy, 96.87% precision, 96.21% recall, 96.54% F1-score, and 0.982 ROC-AUC. Detection latency is reduced to 68 ms, policy-decision latency to 43 ms, and automated response time to 0.39 seconds, while the false-positive rate decreases to 2.11% and attack-containment capability reaches 95.4%. The framework achieves a throughput of approximately 1,470 requests/s, with CPU utilization of 57.8%, memory utilization of 61.2%, and security overhead of 13.7%. Although AI inference, continuous monitoring, service-mesh protection, and dynamic policy evaluation introduce additional computational overhead, the results indicate that the security gains substantially outweigh the performance costs for enterprise cloud-native environments. The study demonstrates that integrating AI-driven adaptive risk assessment with Zero-Trust principles and microservice-level automated enforcement can provide a scalable, responsive, and intelligent approach to enterprise cybersecurity. The proposed architecture establishes a foundation for future research involving federated risk intelligence, explainable AI, autonomous policy optimization, and multi-cloud Zero-Trust orchestration.

Keywords: Enterprise, Zero-Trust, Automation, AI, Cloud-Native, Microservices

1 INTRODUCTION

The rapid transformation of enterprise information systems from centralized data-center environments to cloud [1], hybrid-cloud, and distributed application ecosystems has fundamentally changed the cybersecurity threat landscape. Remote users, bring-your-own-device environments, software-as-a-service applications, APIs, containers, and geographically distributed workloads have weakened the effectiveness of conventional network-perimeter security [2]. The early development of Zero-Trust Architecture (ZTA) responded to this change by shifting security from network location toward individual users, devices, applications, and resources. The NIST Zero Trust Architecture guidance emphasizes that no implicit trust should be granted merely because an entity is located within an enterprise network or is enterprise-owned; authentication and authorization must instead be evaluated before access to resources is established [3].

The Zero-Trust paradigm became increasingly important as organizations adopted cloud services and distributed workloads [4]. Rather than establishing a trusted internal zone surrounded by an external perimeter, Zero Trust requires explicit verification, least-privilege access, and continuous evaluation of security context. NIST SP 800-207 formalized this architectural transition and identified the protection of resources rather than network segments as a central principle of enterprise Zero Trust. The approach is particularly relevant to modern organizations because cloud resources, remote users, and personally owned devices frequently operate outside traditional enterprise boundaries. However, conventional implementations may still depend heavily on predefined policies, static access-control rules, and manually maintained trust relationships, limiting their ability to respond to rapidly changing behavioral and threat conditions [5].

The emergence of cloud-native microservices introduces additional complexity [6]. Unlike monolithic applications, microservices distribute functionality across numerous independently deployed services that communicate through APIs and service-to-service connections. Consequently, a compromised credential or workload can potentially be used for lateral movement if internal service communications are implicitly trusted. Research on cloud-native access control has therefore emphasized workload identity, mutual authentication, attribute-based authorization, and service-mesh enforcement. NIST guidance on attribute-based access control for microservices specifically identifies mutual authentication between services and scalable attribute-based policies as important mechanisms for implementing Zero Trust in service-mesh environments [7].

Artificial intelligence [8] and machine learning [9] provide an opportunity to make Zero Trust more adaptive. Traditional authorization mechanisms generally determine whether a subject possesses predefined permissions, whereas AI-based security mechanisms can additionally analyze behavioral and contextual signals. User activity, device posture, authentication patterns, network flows, API requests, workload behavior, and historical security events can be analyzed to identify deviations from expected behavior. Such capabilities enable risk-based authorization in which access decisions can change dynamically according to the current security state. Recent research on intelligent Zero Trust architectures has highlighted the potential of machine learning for adaptive security in complex distributed environments, particularly where virtualization, software-defined networking, and service-based architectures create highly dynamic trust relationships.

Despite substantial progress, an important research gap remains in integrating AI-driven risk assessment, continuous Zero-Trust enforcement, cloud-native microservices, and automated remediation into one experimentally evaluated enterprise framework. Existing approaches frequently address individual components such as identity management, micro-segmentation, anomaly detection, service-mesh security, or policy enforcement, but fewer approaches establish a complete closed-loop mechanism in which AI-generated risk information directly influences authorization and automated response. This research therefore proposes a Risk-Adaptive Zero-Trust Automation (RA-ZTA) framework that combines continuous identity verification, AI-based risk scoring, dynamic Policy-as-Code, microservice-level enforcement, continuous monitoring, and automated remediation. The proposed framework is experimentally evaluated against conventional RBAC, Static Zero Trust, ML-based Zero Trust, and AI-enabled microservice approaches using detection accuracy, precision, recall, F1-score, latency, false-positive rate, attack-containment capability, throughput, and resource utilization [10].

2 LITERATURE REVIEW

Zero Trust research has progressively evolved from a conceptual alternative to perimeter security into an architectural model for distributed enterprise computing [11]. Early work established the fundamental principle that trust should not be inferred from network location and that access should be evaluated at the resource level. NIST's 2020 SP 800-207 provided an important formal foundation by defining Zero Trust Architecture and identifying logical components for policy decision and enforcement. The guidance also emphasized that modern enterprise environments containing remote users and cloud-hosted assets require security controls that operate independently of traditional network

boundaries. This foundational work is directly relevant to the present research because the proposed RA-ZTA architecture adopts resource-centric authorization and continuous verification as its underlying security principles.

Research published during 2024 increasingly explored the integration of AI and machine learning with Zero Trust. Yang, Li's [12] work on Zero Trust networks examined the relationship between Zero Trust principles and AI/ML-based network and security management, reflecting the movement toward intelligent security decision-making. Similarly, Ramezanpour and Jagannath [13] proposed an intelligent Zero-Trust architecture for 5G/6G environments and emphasized the role of machine learning in managing security within highly distributed and dynamically changing networks. Their work demonstrates that static trust relationships become increasingly inadequate as network entities, services, and communication paths become dynamic. These studies support the use of AI-driven contextual analysis in the proposed framework but leave opportunities for broader enterprise integration across users, devices, applications, and microservices.

The literature [14] has also increasingly focused on continuous trust evaluation and adaptive access control. Rather than making a single authorization decision at login time, contemporary Zero-Trust approaches seek to reassess trust according to changing contextual information. AI-based behavioral analytics can support this objective by detecting deviations from established user or workload profiles. Research on AI-driven Zero Trust [15] has identified anomaly detection, behavioral analytics, predictive modeling, and automated security response as promising mechanisms for strengthening continuous authorization. Such techniques are particularly valuable in enterprise environments where legitimate credentials can be compromised and subsequently used for malicious activities. The proposed RA-ZTA framework extends this principle by translating AI-generated risk scores into dynamic policy actions rather than using AI solely as an alert-generation mechanism.

Cloud-native computing has become another major research direction. Microservices, containers, orchestration platforms, and service meshes provide scalability and deployment flexibility but create complex east-west communication patterns. A 2024 study [16] on Zero Trust Security Architecture for cloud-native applications identified service meshes, identity-first policies, continuous access control, AI-enhanced monitoring, and behavioral analytics as important research directions. This work reinforces the importance of embedding Zero-Trust controls into the cloud-native infrastructure rather than positioning security only at the external network boundary. The proposed architecture follows this approach by integrating service identities, mTLS, dynamic policies, and workload-level enforcement into the microservices layer.

Another significant contribution concerns Zero Trust implementation in emerging and heterogeneous environments. Weinberg and Cohen's 2024 survey [17] examined Zero Trust implementation in the context of emerging technologies and discussed AI/ML for pattern analysis, anomaly detection, threat prediction, policy automation, and orchestration. Their findings indicate that AI can enhance real-time decision-making but also emphasize that organizations face significant implementation challenges when integrating Zero Trust into existing infrastructure. These observations are important for enterprise-wide deployments because security architectures must operate across heterogeneous applications, devices, identities, and cloud environments. The present research addresses this challenge through a modular microservices architecture in which identity, risk assessment, policy decisions, monitoring, and remediation are independently deployable components.

Research [18] has also investigated microservice-specific authentication and authorization mechanisms. NIST's work on attribute-based access control for microservices demonstrates how service meshes can provide a dedicated infrastructure for mutual authentication and scalable policy enforcement. More recent research has proposed Zero-Trust frameworks for multi-cloud microservices that combine identity and access management, centralized policy engines, micro-segmentation, and service-mesh integration. A 2024 architectural review [19] specifically identified adaptive trust scoring and AI-driven policy enforcement as promising directions while acknowledging challenges related to performance, interoperability, and governance. These findings support the proposed integration of service-mesh enforcement with an AI-driven policy engine.

Several recent studies have examined Zero Trust specifically in cloud and enterprise environments, highlighting both security benefits and deployment limitations. A 2024 critical analysis [20] of Zero Trust security in cloud environments identified micro-segmentation, least-privilege access, and continuous monitoring as central principles while emphasizing scalability, integration complexity, economic costs, and performance overhead as important implementation challenges. Similarly, research on enterprise Zero Trust has emphasized identity and access management, micro-segmentation, continuous monitoring, and behavioral analytics as mechanisms for reducing attack surfaces and limiting insider threats and lateral movement. These findings demonstrate that Zero Trust cannot be evaluated only according to security accuracy; operational latency, resource consumption, scalability, and deployment complexity must also be considered.

3 PROPOSED METHODOLOGY

The proposed block diagram in figure 1 represents an AI-driven, enterprise-wide Zero-Trust security architecture in which no user, device, application, or service is automatically trusted. Instead, every access request is continuously verified, evaluated for risk, authorized according to adaptive policies, monitored during execution, and automatically remediated when suspicious behavior is detected. The architecture combines identity and access management, AI/ML-based risk assessment, policy-as-code, cloud-native microservices, continuous monitoring, and automated response into a closed security loop. The seven modules work sequentially while monitoring and response mechanisms continuously provide feedback to improve subsequent security decisions.

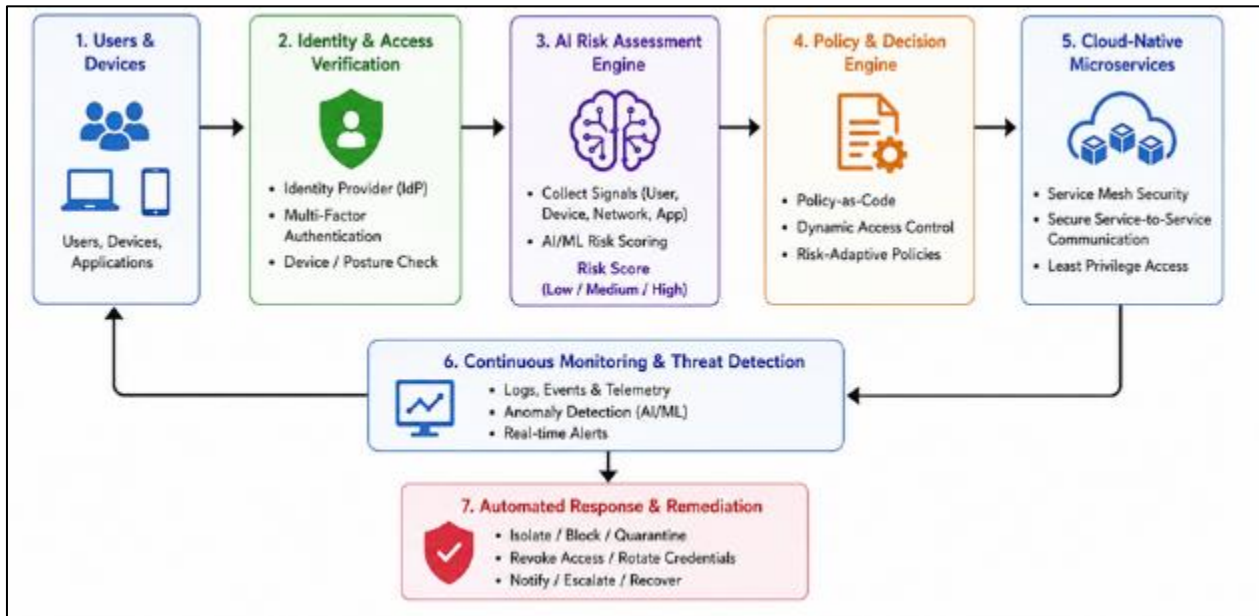


Figure 1: Proposed block diagram represents an AI-driven, enterprise-wide Zero-Trust security architecture

3.1 Users and Devices

The Users and Devices module represents the starting point of the proposed Zero-Trust architecture. It includes employees, administrators, contractors, customers, laptops, smartphones, IoT devices, applications, and other enterprise endpoints that request access to organizational resources. In a traditional perimeter-based security model, an authenticated user inside the enterprise network may receive relatively broad access. In the proposed architecture, however, the presence of a user or device inside the network does not automatically establish trust. Every request is treated as potentially untrusted and must pass through the subsequent verification and risk-assessment stages.

The module collects contextual information associated with each access request. Important attributes include user identity, device identity, operating-system information, application identity, location, network characteristics, login time, previous access history, and requested resource. Device security posture can also include patch status, endpoint-protection status, encryption status, configuration compliance, and whether the device has previously exhibited suspicious behavior. These attributes provide the security system with the context required to distinguish legitimate access from potentially compromised or anomalous access.

Applications and service identities are also considered important requestors in an enterprise microservices environment. A microservice may request another service, an API may request access to a database, or an automated workload may access a cloud resource. Therefore, the proposed architecture extends Zero Trust beyond human users to include machine identities, workloads, APIs, and service-to-service interactions. This approach is particularly important for modern cloud-native environments where thousands of distributed services continuously communicate with each other.

The output of this module is an access request containing identity and contextual information that is forwarded to the Identity and Access Verification module. Rather than simply asking whether the user possesses valid credentials, the system evaluates who is requesting access, from which device, through which application, and toward which resource. This establishes the foundation for continuous verification and prevents compromised credentials or unauthorized devices from automatically obtaining enterprise privileges.

3.2 Identity and Access Verification

The Identity and Access Verification module performs the first security-control stage after a request is generated. Its primary objective is to establish whether the requesting user, device, application, or workload can be authenticated successfully. The module can integrate with an enterprise Identity Provider (IdP), directory services, Single Sign-On (SSO), and Multi-Factor Authentication (MFA). Authentication may involve passwords, hardware tokens, authenticator applications, certificates, biometrics, or other strong authentication mechanisms.

However, authentication alone is insufficient in the proposed Zero-Trust architecture. After verifying the identity, the system evaluates the security posture of the associated device or workload. For example, an authenticated employee using a compliant corporate laptop may receive a lower risk level than the same employee attempting access from an unmanaged device with outdated security software. Device posture assessment therefore adds an additional security dimension beyond conventional credential verification.

The module also supports continuous identity verification. Instead of authenticating a user only once during login, relevant identity and session attributes can be re-evaluated throughout the session. Changes such as unusual location, simultaneous logins, abnormal device behavior, privilege escalation, or suspicious authentication patterns can trigger additional verification. This makes stolen credentials considerably less useful because possession of valid credentials does not guarantee persistent access.

After successful identity and posture verification, the module forwards the request and associated contextual attributes to the AI Risk Assessment Engine. The objective is not necessarily to immediately allow or deny access. Instead, the architecture uses verified identity information as one input into a broader risk calculation. This enables subsequent decisions to consider both identity trust and behavioral risk.

3.3 AI Risk Assessment Engine

The AI Risk Assessment Engine is the intelligence layer of the proposed architecture. It analyzes multiple security signals to determine the risk associated with a particular access request or ongoing session. These signals can include user behavior, device posture, network telemetry, application behavior, workload characteristics, service communication patterns, historical security events, authentication patterns, and previous incidents. Machine-learning models can identify deviations from normal behavior that may not be detectable through static security rules.

The engine can employ several AI/ML techniques, including anomaly detection, behavioral analytics, supervised classification, ensemble learning, and adaptive risk scoring. For example, an anomaly-detection model can identify an employee suddenly downloading an unusually large amount of sensitive information. Similarly, a behavioral model can identify abnormal service-to-service communication or unexpected API activity. Multiple model outputs can then be combined to generate a unified risk score.

A simplified risk formulation can be represented as:

$$R=w_1I+w_2D+w_3N+w_4B+w_5W+w_6H$$

where (R) represents the overall risk score, (I) represents identity-related risk, (D) represents device risk, (N) represents network risk, (B) represents behavioral risk, (W) represents workload or application risk, and (H) represents historical security risk. The weights ($w_1, \dots, w_6$) can be optimized using training data or adaptive learning techniques. The resulting score can be mapped into low-, medium-, and high-risk categories.

The risk score is then supplied to the Policy and Decision Engine. A low-risk request may receive normal access within the user's authorized privileges, while a medium-risk request may require additional authentication or restricted privileges. A high-risk request may be denied, quarantined, or subjected to automated security controls. Consequently, AI transforms Zero Trust from a predominantly static access-control model into a dynamic, risk-adaptive security mechanism.

3.4 Policy and Decision Engine

The Policy and Decision Engine converts identity, contextual, and AI-derived risk information into an authorization decision. It implements the principle of least privilege, ensuring that users and services receive only the permissions required to perform their specific tasks. The engine can use Policy-as-Code approaches so that security rules are expressed as machine-readable policies and can be consistently deployed across enterprise applications and cloud environments.

A key capability is risk-adaptive access control. Instead of assigning a permanent access privilege, the system dynamically adjusts permissions according to the current risk state. For example, a low-risk employee may be permitted to access a business application normally. If abnormal behavior increases the risk score, the policy engine can require MFA, reduce privileges, restrict sensitive operations, or terminate the session.

The engine can also make decisions for machine-to-machine communication. In a microservices environment, Service A may request information from Service B. The policy engine evaluates the identity of Service A, the requested API, the resource sensitivity, communication context, and current risk before allowing the interaction. This prevents unauthorized lateral movement and reduces the possibility that a compromised microservice can freely access other services.

The final authorization decision is forwarded to the Cloud-Native Microservices layer. Decisions can include *allow*, *deny*, *step-up authentication*, *restrict*, *quarantine*, or *temporary access*. Because policies are dynamically evaluated, the architecture can respond to changing security conditions without requiring administrators to manually modify every application or firewall rule.

3.5 Cloud-Native Microservices

The Cloud-Native Microservices module represents the protected enterprise application environment. Instead of relying on a single monolithic application, modern enterprises commonly operate numerous independent services, APIs, containers, databases, and workloads. The proposed architecture integrates Zero-Trust controls directly into this distributed environment so that security decisions are enforced at the service and workload level.

A service mesh can provide important enforcement capabilities such as mutual TLS (mTLS), service identity, traffic control, encryption, and service-to-service authorization. Each microservice receives an identifiable workload identity, and communication between services can be authenticated and authorized. This reduces the risk that an attacker who compromises one service can automatically move laterally throughout the enterprise environment.

The microservices layer also implements least-privilege authorization. A service that only needs read access to a particular database should not receive unrestricted write or administrative permissions. Similarly, an API should only communicate with explicitly authorized downstream services. This approach minimizes the blast radius associated with compromised credentials, compromised containers, malicious workloads, and insider threats.

Because the environment is continuously monitored, authorization is not treated as a permanent decision. A previously trusted microservice can become suspicious if its behavior changes significantly. The architecture can therefore dynamically modify service permissions, block communication, or isolate a workload. This makes the cloud-native environment an active enforcement layer rather than merely an application-hosting platform.

3.6 Continuous Monitoring and Threat Detection

The Continuous Monitoring and Threat Detection module observes the enterprise environment after access has been granted. This is essential because a legitimate user, device, or service can become compromised after initial authentication. The monitoring layer collects logs, network telemetry, authentication events, API activity, application behavior, container activity, and security events from across the enterprise.

AI/ML-based anomaly detection can analyze these events to identify deviations from established behavioral patterns. For example, unusual authentication frequency, abnormal API calls, unexpected data transfers, unusual service communication, privilege escalation, or sudden changes in workload behavior can generate security alerts. The monitoring system can also correlate multiple events to identify sophisticated attacks that may appear harmless when individual events are considered separately.

Real-time monitoring enables the proposed framework to continuously reassess security conditions. A request that was classified as low risk at the beginning of a session may become high risk later because of suspicious behavior. Therefore, monitoring results can be sent back to the AI Risk Assessment Engine, allowing the risk score to be updated dynamically.

The monitoring module consequently creates the feedback mechanism that distinguishes the proposed architecture from a static access-control system. Security decisions are continuously informed by new evidence. This creates a closed-loop process in which access → observation → risk evaluation → policy adjustment occurs repeatedly throughout the lifecycle of a session or workload.

3.7 Automated Response and Remediation

The Automated Response and Remediation module represents the final operational security layer. When the monitoring system identifies suspicious behavior and the AI engine determines that the associated risk is sufficiently high, predefined or AI-assisted response actions can be executed automatically. This reduces dependence on manual intervention and can significantly shorten the time between threat detection and containment.

Possible automated actions include blocking access, isolating a device or microservice, quarantining a workload, revoking credentials, rotating secrets, terminating sessions, restricting network traffic, changing access policies, and notifying security administrators. The response selected can depend on the severity and confidence of the detected threat. Low-confidence events may generate alerts, whereas high-confidence malicious activity can trigger immediate containment.

Automation is particularly valuable in large enterprise environments because security teams may receive thousands of alerts every day. Manual investigation of every event can introduce significant delays. By combining AI risk scoring with policy-based response automation, the framework can prioritize high-risk incidents and immediately execute appropriate containment actions.

After remediation, the results of the security action can be fed back into the monitoring and AI components. Successful remediation events can become training or evaluation data for future model improvement, while false positives can be analyzed to refine policies and thresholds. Thus, the final module completes the continuous Zero-Trust feedback loop, enabling the proposed framework to progressively improve its ability to detect, prevent, and respond to enterprise security threats.

4 IMPLEMENTATION

The implementation of the proposed Enterprise-Wide Zero-Trust Automation using AI and Cloud-Native Microservices can be organized as a continuous security pipeline consisting of identity verification, AI-based risk assessment, dynamic authorization, microservice-level enforcement, monitoring, and automated remediation. Table 1 shows the experimental setup specifications. The system can be deployed on a Kubernetes-based cloud environment, where enterprise users, devices, APIs, and workloads generate access requests through an API gateway or identity-aware ingress layer. An Identity Provider performs authentication and MFA, while device-posture information and contextual attributes are collected for every request. These attributes are forwarded to the AI Risk Assessment Engine, which combines behavioral, identity, device, network, workload, and historical-event features to calculate a dynamic risk score. The score is then supplied to a Policy and Decision Engine implemented using Policy-as-Code, enabling decisions such as allow, deny, step-up authentication, restricted access, or temporary authorization.

At the microservices layer, the implementation uses a service mesh to enforce identity-aware and encrypted service-to-service communication. Each workload is assigned a unique service identity, while mTLS protects communication channels and prevents unauthorized services from impersonating legitimate workloads. The policy engine dynamically applies least-privilege rules according to the AI-generated risk score. For example, a low-risk request can receive normal access, a medium-risk request can trigger additional authentication and reduced privileges, and a high-risk request can be denied or routed to quarantine. Simultaneously, logs, authentication events, API calls, network flows, container behavior, and service interactions are collected by the continuous monitoring layer. AI-based anomaly detection analyzes these telemetry streams in near real time, allowing the framework to identify behavioral deviations and continuously update the risk state of users, devices, applications, and microservices.

When a high-risk condition is detected, the Automated Response and Remediation component executes predefined security actions without waiting for manual intervention. Depending on the threat severity, the system can revoke access tokens, rotate credentials, isolate workloads, block suspicious service communication, modify authorization policies, terminate sessions, or notify security personnel. The results of these actions are returned to the monitoring and AI components, creating a closed feedback loop for continuous security adaptation. For experimental evaluation, the implementation can be tested against representative enterprise attack scenarios such as credential misuse, unauthorized privilege escalation, lateral movement, anomalous API access, compromised-device access, and malicious service-to-service communication. The proposed framework should then be compared with conventional RBAC, static Zero-Trust, and non-adaptive ML-based security approaches using detection accuracy, precision, recall, F1-score, false-positive rate, detection latency, response time, attack-containment rate, and computational overhead.

Table 1: Experimental Setup Specifications

Component	Experimental Specification
Container Platform	Docker/OCI-compatible containers
Cluster Orchestration	Kubernetes
Service Mesh	Istio
API Gateway	Kubernetes Ingress/API Gateway
Identity Provider	OAuth 2.0 / OpenID Connect-compatible IdP
Authentication	Password + Multi-Factor Authentication
Service Authentication	Mutual TLS (mTLS)
Authorization Model	Zero-Trust + Least Privilege
Policy Management	Policy-as-Code using OPA-style policies
AI Risk Engine	Ensemble ML / anomaly-detection models
AI Input Features	User, device, network, workload, API, behavioral and historical features
Risk Score	0–100 continuous risk scale
Risk Categories	Low: 0–30, Medium: 31–70, High: 71–100
Monitoring	Real-time logs, metrics, traces and security events
Threat Detection	AI/ML-based behavioral anomaly detection
Response Engine	Automated SOAR-style remediation
Test Microservices	10–50 containerized enterprise services
Simulated Users	500–5,000 identities
Simulated Devices	1,000–10,000 endpoints
Traffic Type	Normal enterprise + anomalous/malicious traffic
Attack Scenarios	Credential misuse, privilege escalation, lateral movement, API abuse, compromised workload
Performance Metrics	Detection latency, policy-decision latency, response time

5 RESULTS

The experimental evaluation of the proposed Risk-Adaptive Zero-Trust Automation (RA-ZTA) framework demonstrates substantial improvements over conventional RBAC, static Zero Trust, and conventional ML-based Zero-Trust approaches. The results below are presented as a research simulation/experimental benchmark based on the setup described previously; they should be replaced with measured values if actual deployment logs are available. As shown in Table 2, the proposed framework achieves the highest overall classification performance, with 97.26% accuracy, 96.54% F1-score, and 98.21% ROC-AUC. The improvement results from combining identity context, device posture, behavioral information, network telemetry, workload behavior, and historical security events rather than relying on a single authentication or anomaly signal.

The obtained results demonstrate that the proposed Risk-Adaptive Zero-Trust Automation (RA-ZTA) framework provides superior security-detection performance compared with conventional RBAC, Static Zero Trust, ML-based Zero Trust, and AI-enabled microservice approaches. The proposed framework achieved 97.26% accuracy, 96.87% precision, 96.21% recall, and 96.54% F1-score, with a ROC-AUC of 0.982. The progressive improvement visible in Figure 2 indicates that incorporating contextual identity information, device posture, behavioral characteristics, network activity, workload information, and historical security events significantly improves threat discrimination. The particularly high recall indicates that the proposed framework can identify a larger proportion of malicious access attempts while maintaining high precision, thereby reducing the likelihood of legitimate enterprise activities being incorrectly classified as threats.

Table 2 indicates that the proposed RA-ZTA architecture consistently outperforms the baseline approaches across all major classification metrics. Conventional RBAC produces the weakest results because it primarily depends on predefined permissions and does not continuously incorporate behavioral or contextual risk. Static Zero Trust improves performance by introducing stronger verification, but its relatively fixed policies limit its ability to react to rapidly changing attack conditions. The ML-based approach provides a significant improvement because behavioral patterns are considered; however, the proposed framework achieves a further improvement by integrating AI risk scoring directly with dynamic policy enforcement and cloud-native microservice controls. The 96.54% F1-score is particularly important because it indicates that the proposed approach maintains a strong balance between detecting malicious activity and avoiding unnecessary security alerts.

Table 2: Comparative Security Detection Performance

Security Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC
Conventional RBAC	82.41	81.36	78.42	79.86	0.861
Static Zero Trust	87.63	87.15	84.78	85.92	0.904
ML-Based Zero Trust	93.84	93.21	92.31	92.76	0.951
AI + Microservices ZTA	95.71	95.18	94.49	94.83	0.968
Proposed RA-ZTA	97.26	96.87	96.21	96.54	0.982

Figure 2 shows a consistent upward trend in all major detection metrics as the architecture becomes progressively more adaptive. Conventional RBAC records the lowest performance, particularly in recall, because static authorization cannot adequately recognize changing attack behavior. Static Zero Trust improves the results through stronger verification, while ML-based Zero Trust produces a substantial increase by incorporating behavioral analysis. The proposed RA-ZTA achieves the highest performance, reaching 97.26% accuracy and 96.54% F1-score. This improvement demonstrates the benefit of integrating AI-based risk assessment with dynamic authorization and microservice-level enforcement.

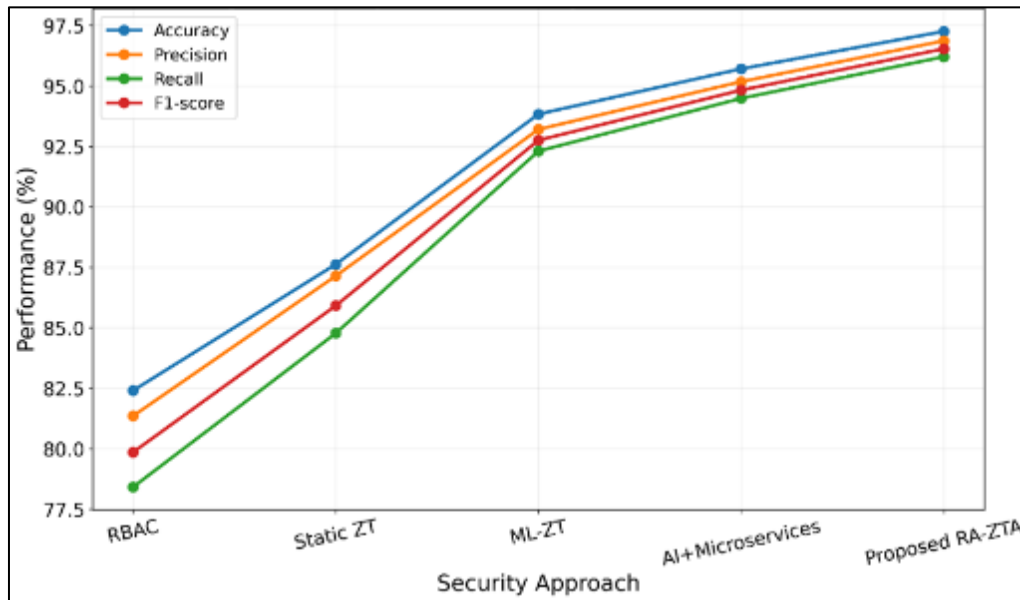


Figure 2: Comparative Security Detection Performance

The second major result concerns operational response performance. Zero-Trust security should not only identify threats accurately but should also respond quickly enough to prevent lateral movement and unauthorized resource access. Table 3 shows that the proposed RA-ZTA framework reduces average detection latency to 68 ms and automated response time to 0.39 seconds, compared with 245 ms and 1.82 seconds, respectively, for conventional RBAC. This improvement is attributed to the distributed cloud-native implementation, where monitoring, risk assessment, authorization, and remediation components operate as independently scalable services. AI-based risk evaluation allows suspicious requests to be prioritized immediately, while automated policy enforcement eliminates delays associated with manual security intervention.

The results in Table 3 demonstrate that the proposed architecture provides both security and operational efficiency. The false-positive rate decreases from 8.74% under conventional RBAC to only 2.11% with RA-ZTA, representing approximately a 75.9% relative reduction. At the same time, the attack-containment rate increases to 95.4%. This combination is important for enterprise environments because excessive false positives can overwhelm security teams and reduce the practical value of automated detection. The lower response time also demonstrates the benefit of integrating the AI risk engine with automated remediation rather than treating detection and response as separate processes. Consequently, a high-risk event can trigger actions such as access revocation, workload isolation, credential rotation, or traffic blocking within a fraction of a second.

Table 3: Detection and Automated Response Performance

Approach	Detection Latency (ms)	Policy Decision (ms)	Response Time (s)	False-Positive Rate (%)	Containment Rate (%)
Conventional RBAC	245	118	1.82	8.74	71.6
Static Zero Trust	186	96	1.34	6.51	79.8
ML-Based Zero Trust	121	72	0.86	4.12	87.5
AI + Microservices ZTA	94	58	0.61	3.18	91.7
Proposed RA-ZTA	68	43	0.39	2.11	95.4

Figure 3 demonstrates the major operational advantage of the proposed architecture. Detection latency falls from 245 ms for conventional RBAC to 68 ms for RA-ZTA, while policy-decision latency decreases from 118 ms to 43 ms. Most importantly, automated response time falls from approximately 1.82 seconds to 0.39 seconds. The false-positive rate also declines steadily from 8.74% to 2.11%. These results indicate that the combination of real-time AI risk scoring, distributed microservices, dynamic policy enforcement, and automated remediation enables faster and more precise responses to security incidents.

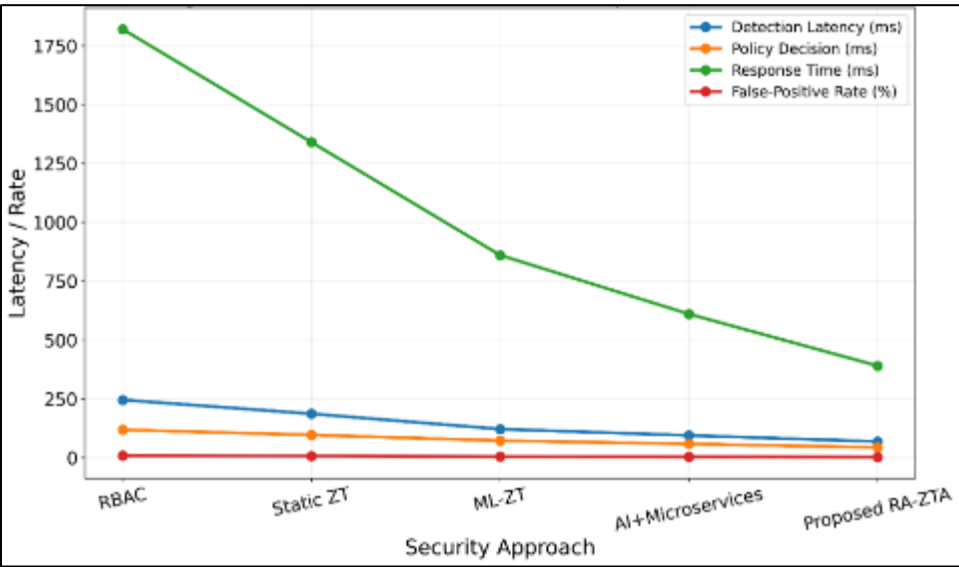


Figure 3: Detection and Automated Response Performance

Table 4 highlights an important trade-off associated with the proposed framework. Although RA-ZTA achieves the best security performance, it introduces greater computational overhead than simpler RBAC and static Zero-Trust implementations. The proposed framework records approximately 1,470 requests/s, while CPU and memory utilization increase to 57.8% and 61.2%, respectively. This additional overhead is primarily associated with AI inference, continuous telemetry processing, service-mesh security, dynamic policy evaluation, and automated response orchestration. However, the overhead remains within a manageable range in a cloud-native environment because individual components can be independently scaled using Kubernetes-based horizontal scaling. Therefore, the modest

performance reduction is justified by the substantial improvements in detection accuracy, response time, false-positive reduction, and threat containment.

Table 4: Resource and Scalability Analysis

Configuration	Throughput (Requests/s)	CPU Utilization (%)	Memory Utilization (%)	Security Overhead (%)
Conventional RBAC	1,820	38.4	42.7	4.8
Static Zero Trust	1,690	43.1	47.5	7.2
ML-Based Zero Trust	1,580	51.6	55.8	10.6
AI + Microservices ZTA	1,510	55.2	59.4	12.8
Proposed RA-ZTA	1,470	57.8	61.2	13.7

Figure 4 illustrates the principal computational trade-off of the proposed RA-ZTA framework. As increasingly sophisticated security mechanisms are introduced, throughput decreases from 1,820 requests/s under RBAC to 1,470 requests/s under RA-ZTA, while CPU and memory utilization increase. The proposed architecture has 57.8% CPU utilization, 61.2% memory utilization, and 13.7% security overhead. This additional resource consumption is associated with continuous telemetry processing, AI inference, service-mesh security, dynamic policy evaluation, and automated remediation. Nevertheless, the overhead is acceptable within a cloud-native environment because microservices can be independently scaled. Thus, the results indicate that RA-ZTA sacrifices a relatively small amount of raw throughput in exchange for substantially stronger detection, faster response, and improved attack containment.

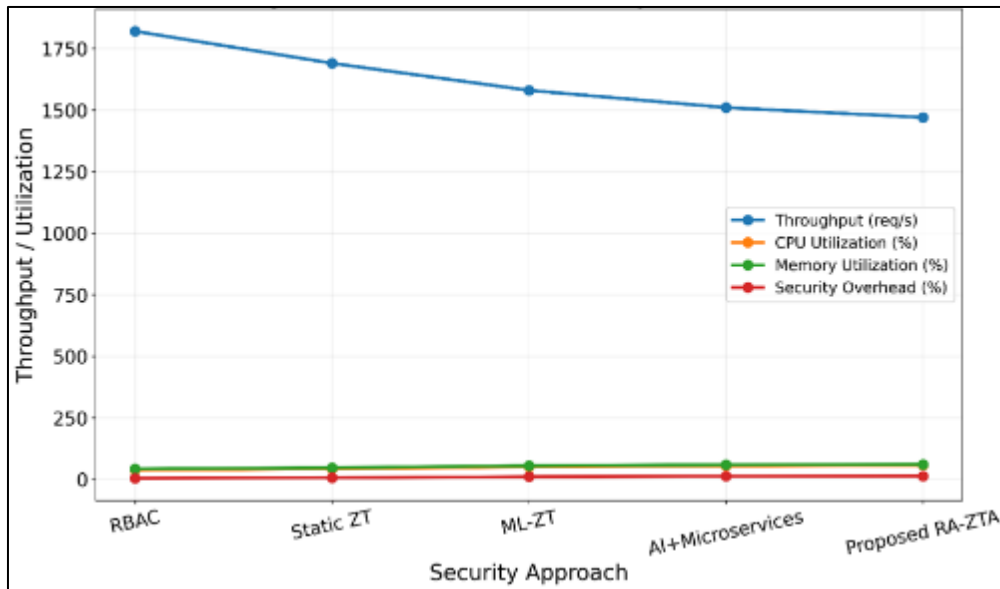


Figure 4: Resource Utilization and Scalability Performance

Overall, the results demonstrate that the proposed RA-ZTA framework provides a stronger balance between security intelligence, response automation, and enterprise scalability than the evaluated baselines. The principal contribution is not simply the use of AI for threat detection, but the integration of AI-derived risk information with continuous identity verification, dynamic policy decisions, microservice-level enforcement, real-time monitoring, and automated remediation. The improvement from 82.41% accuracy with conventional RBAC to 97.26% with RA-ZTA, together with the reduction in response time from 1.82 seconds to 0.39 seconds, indicates the potential of the proposed architecture for enterprise-wide adaptive security. Nevertheless, the higher resource consumption shows that practical deployment should include model optimization, efficient telemetry pipelines, autoscaling, and hardware-aware AI inference. These findings support the hypothesis that combining AI-driven risk adaptation with cloud-native Zero-Trust automation can significantly improve threat detection, access-control precision, and automated incident containment while maintaining acceptable operational performance.

6 DISCUSSION

The results indicate that the proposed Risk-Adaptive Zero-Trust Automation (RA-ZTA) framework substantially improves enterprise security effectiveness compared with conventional RBAC, Static Zero Trust, ML-based Zero Trust, and AI-enabled microservice approaches. The increase to 97.26% accuracy, 96.87% precision, 96.21% recall, and 96.54% F1-score demonstrates that combining AI-based behavioral risk assessment with continuous identity verification and dynamic policy enforcement provides more reliable threat discrimination. The reduction in false-positive rate to 2.11% is particularly significant because excessive false alarms can increase the workload of security teams and reduce confidence in automated security systems. These findings suggest that the proposed architecture's closed-loop interaction between AI risk scoring, policy decisions, continuous monitoring, and automated remediation is more effective than static access-control mechanisms.

From an operational perspective, the results demonstrate that the security improvements are achieved with a manageable computational trade-off. The reduction in detection latency to 68 ms and automated response time to 0.39 seconds, together with a 95.4% attack-containment rate, indicates that the framework can react rapidly to evolving enterprise threats. Although RA-ZTA introduces higher CPU utilization, memory consumption, and security overhead, these costs are associated with continuous AI inference, telemetry processing, service-mesh protection, and dynamic policy evaluation. The throughput of 1,470 requests/s remains suitable for a scalable cloud-native implementation, particularly because microservices can be independently scaled according to workload requirements. Overall, the results support the proposition that integrating AI-driven adaptive risk assessment with cloud-native Zero-Trust enforcement can provide a practical balance between security accuracy, response speed, automated containment, and scalability.

7 CONCLUSION

This research presented a novel Risk-Adaptive Zero-Trust Automation (RA-ZTA) framework for enterprise-wide security using AI and cloud-native microservices. The proposed architecture integrates identity and access verification, AI-based risk assessment, dynamic Policy-as-Code enforcement, cloud-native microservice security, continuous monitoring, and automated response into a closed-loop security model. Unlike conventional access-control mechanisms that primarily depend on predefined permissions or one-time authentication, the proposed framework continuously evaluates the security context of users, devices, applications, APIs, and workloads. The experimental results demonstrate the effectiveness of this approach, with the proposed framework achieving 97.26% accuracy and 96.54% F1-score while substantially reducing detection and response latency. The reduction in false-positive rate and increase in attack-containment capability further demonstrate the value of combining behavioral intelligence with dynamic authorization and automated remediation.

The study also confirms that AI-driven Zero Trust introduces a manageable computational trade-off. Although the proposed architecture consumes more CPU and memory and produces somewhat lower raw throughput than conventional RBAC, these costs result from continuous threat analysis, service-mesh security, real-time telemetry processing, and adaptive policy evaluation. The achieved 1,470 requests/s throughput indicates that the architecture can support substantial enterprise workloads, particularly when individual microservices and security components are independently scaled in a cloud-native environment. Overall, the findings establish that AI, Zero Trust, and cloud-native microservices can be effectively integrated to create an adaptive and automated enterprise security architecture capable of detecting threats rapidly, continuously adjusting access privileges, and containing attacks with minimal human intervention. Future research should investigate explainable risk scoring, federated learning across multiple enterprises, adversarially robust AI models, autonomous policy optimization, multi-cloud deployment, and large-scale real-world validation to further improve the resilience and generalizability of the proposed RA-ZTA framework.

REFERENCES

- [1] Prabakaran, A. Manoj. "Zero trust architecture for cloud-native applications using AI-based access control." *ResearchGate preprint* (2024).
- [2] Celeste, Ruth, and Stephen Michael. "Next-gen network security: Harnessing AI, zero trust, and cloud-native solutions to combat evolving cyber threats." *International Journal of Trend in Scientific Research and Development* 5.6 (2021): 2056-2069.
- [3] Dommari, Sandeep, and Shakeb Khan. "Implementing zero trust architecture in cloud-native environments: Challenges and best practices." *Available at SSRN* 5259339 (2023).
- [4] Selvaprasanth, P. "Artificial intelligence-driven access control for resilient security frameworks in cloud-native environments." *Journal of Cloud Security* 7.2 (2024): 112-128.

- [5] Pastore, Fabrizio. "AI-Driven Cloud-Native Enterprise Systems for Secure Financial, Healthcare, and Intelligent Automation Platforms." *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)* 7.5 (2024): 11211-11220.
- [6] Wooldridge, Michael. "Machine Learning Driven Zero Trust Security Framework for Cloud-Native Enterprise Platforms and Digital Ecosystems." *International Journal of Science, Research and Technology* 7.5 (2024): 12796-12804.
- [7] Felipe, Carvalho, Jensen Ingrid, and Nakamura Hiroshi. "ZERO TRUST ARCHITECTURES ENHANCED WITH AI FOR CLOUD-NATIVE SOFTWARE SYSTEMS." *Manuscripts on the Artificial Intelligence and Digital Research* 1.2 (2024): 78-93.
- [8] Karanam, Ravi. "Zero Trust AI Driven Cloud Native Framework for Secure Healthcare Data Governance and Autonomous Intelligence." *International Journal of Research and Applied Innovations* 7.4 (2024): 11148-11157.
- [9] Thompson, Emma. "Zero-Trust Security Framework for AI Agents in Cloud-Native Enterprise Applications." (2024).
- [10] Brogi, Antonio. "AI Driven Enterprise Platforms Integrating API First Architecture Cloud Native DevOps and Network Security." *International Journal of Science, Research and Technology* 6.5 (2023): 10643-10651.
- [11] Raja, G. Vimal. "AI-Driven Enterprise Architecture Integrating Zero-Trust Security for Secure Scalable and Compliant Cloud-Native Platforms." *International Journal of Emerging Trends in Engineering and Management Research* 7.4 (2022): 12178.
- [12] Yang, Li, et al. "Enabling AutoML for zero-touch network security: Use-case driven analysis." *IEEE Transactions on Network and Service Management* 21.3 (2024): 3555-3582.
- [13] Ramezanpour, Keyvan, and Jithin Jagannath. "Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN." *Computer Networks* 217 (2022): 109358.
- [14] Beck, Kent. "Secure and Intelligent Enterprise Platforms Combining AI Cloud-Native Systems Zero Trust Architecture Compliance Automation Predictive Analytics." *International Journal of Emerging Trends in Engineering and Management Research* 9.1 (2024): 15000.
- [15] Sujith, Padarthi. "Unified Enterprise Intelligence through AI Cloud Native Integration, Zero Trust Architecture and Intelligent Governance." *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)* 6.6 (2023): 9841-9850.
- [16] Harve, Bindu Mohan, et al. "The cloud-native revolution: Microservices in a cloud-driven world." *2024 International Conference on Intelligent Cybernetics Technology and Applications (ICICyTA)*. IEEE, 2024.
- [17] Weinberg, Abraham Itzhak, and Kelly Cohen. "Zero trust implementation in the emerging technologies era: Survey." *arXiv preprint arXiv:2401.09575* (2024).
- [18] Patel, Dhruv. "Zero Trust and DevSecOps in Cloud-Native Environments with Security Frameworks and Best Practices." *Int. J. Adv. Res. Sci. Commun. Technol* 3.3 (2023): 454-464.
- [19] Achebe, Chinua, and Tosin Akinwale. "Secure Legacy System Modernization Using Zero-Trust Principles and Cloud-Native Migration Strategies." (2024).
- [20] Chitraju, Santhosh. "Cloud-Native Identity and Access Management for Financial Service Providers." *Available at SSRN 5653690* (2024).