JOURNAL OF ARTIFICIAL INTELLIGENCE AND ROBOTICS
Jan-Mar 2026, VOL. 3, ISSUE 1, pp. 1-11
https://doi.org/10.61577/jaiar.2026.100001

RESEAPRO JOURNALS

ORIGINAL ARTICLE

OPEN ACCESS

Intelligent adaptive authentication for zero-trust microservice architectures using AI-driven behavioral analytics

Aditya Rautaray

Department of Cloud Security Service, CVS Healthcare, Rhode Island, USA

ABSTRACT

Securing microservices within enterprise settings is difficult due to each microservice exposing its own API, the attack surface increasing with the number of services, and the current Zero Trust Architecture (ZTA) frameworks not incorporating any behavioral analysis when evaluating requests. In this paper, we introduce an AI-enabled adaptive authentication framework that uses a hybrid machine learning model based on a random forest classifier and isolation forest anomaly detector to compute a continuous risk score for each authentication request. Depending on the risk score, a three-stage policy decision is made either allowing seamless access, performing multi-factor authentication, or denying access to the requested resource. The policy decision process takes place via the Policy Decision Point (PDP) and Policy Enforcement Point (PEP) in the service mesh architecture. On a synthetic dataset of 10,000 authentication events, our solution reaches a classification accuracy of 94.2%, a 4.8% False Positive Rate (FPR), and an end-to-end decision latency of 35-50 ms, outperforming a static ZTA baseline by 7 percentage points in terms of accuracy. Only 21% of all authentication requests require multi-factor authentication, while under a uniform authentication scheme, 100% of them would need it. All performance numbers reported were obtained using simulation to evaluate our solution on actual Kubernetes clusters is the next research direction.

KEY WORDS

Zero Trust Architecture (ZTA); Adaptive authentication; Microservices security; AI risk engine; Risk-based access control

ARTICLE HISTORY

Received 27 January 2026;
Revised 18 February 2026;
Accepted 25 February 2026

Introduction

Most enterprise software these days is not a big monolithic application. It has shifted toward a world where many little, autonomously deployable services, each with its own logical network boundary and doing some fraction of the workload, communicate through simple APIs. This new world has many advantages: faster development cycles, huge loads living in a scaled-out manner and isolation of system failures. But it also brings with it a host of security issues.

Traditional authentication mechanisms work well in legacy, monolithic architectures where components reside within a predefined boundary of control. In a microservice architecture that uses services that are ephemeral, dynamically elastic, and have no defined boundaries of control, this sort of identity check is significantly more complex. During this service-to-service communication, a user in need of access to a given resource in a service needs to be verified at each step before proceeding, even if that user has acquired access via a prior resource. An attacker can exploit this invocation by getting access to the token or gaining control of a given service; then invocations can span many services without being detected. As a safeguard against this, the ZTA was established. The concept at its core (never trust and verify), is quite straightforward (yet not entirely clear in implementation): all requests must be independently authorized and deemed worthy of access based on the user and request alone, regardless of where a like request may originate from or if access has already been granted [1,2].

This is a good baseline for sure, but nine times out of ten, practically everything I've seen has been so rigid as it exists today, you can't use it as the default policy for every situation involving you, the user, or the client's environment. The key element here is that it allows you to firewall your security based on the real-time context of authentication.

In this paper, I implemented a secure microservices authentication system, which I called Zero Trust, i.e., an intelligent firewall. On the "back" of it is running the AI Risk Engine, which defines an access authentication risk of each login according to behaviors and contexts. A low-risk request can be authorized using the default authentication criteria. A medium-risk request needs a second factor of authentication and a high-risk request is denied. The same engine makes the decisions, allowing it to reach the same security level across the whole microservice, executing those decisions through a PEP/PDP with uniform service mesh deployment of each microservice.

The principal contributions of this work are:

An AI-powered risk assessment engine for authentication that continuously evaluates the behavior of access attempts across microservices.

An adaptive authentication system adjusts the level of verification required based on a calculated risk score. The entire research process, including system design, data flow, model selection, and evaluation methods, is explained in detail to enable other researchers to replicate the study.

Joint PDP/PEP enforcement of Zero Trust security, preventing access to unauthorized resources, together with a thorough per-component and weight-sensitivity analysis of the risk-scoring mechanism. The remainder of this paper is organized as follows. Joint PDP/PEP enforcement of Zero Trust security, preventing

***Correspondence:** Mr. Aditya Rautray, Department of Cloud Security Service, CVS Healthcare, Rhode Island, USA, e-mail: adisaivikram@gmail.com

access to unauthorized resources, together with a thorough per-component and weight-sensitivity analysis of the risk-scoring mechanism. The remainder of this paper is organized as follows. The literature review section reviews related work. The methodology section describes the research methodology. The proposed architecture and AI risk engine section introduces the proposed architecture and its AI risk engine. The implementation details and policy enforcement section detail the implementation and policy enforcement mechanisms. The experimental evaluation and results section present the experimental evaluation and results, including the ablation analysis. The discussion section discusses the findings and their limitations. The conclusion and future work section concludes the paper.

Literature Review

The design of the proposed framework draws on research in three converging fields: distributed systems security, behavioral analytics, and machine learning. This section reviews work across five key areas before situating the proposed system relative to that body of work.

ZTA: Foundations and evolution

Zero Trust approach was created by Kindervag at Forrester Research as a perimeter security angle: in this perspective, no user, device or service should be trusted no matter where it is in the network, NIST SP800207 architecture has now defined its own canonical set of parts: the PDP, the PEP, the Policy Administration Point (PAP) and the Realtime monitoring component [3,2].

The major operational deficiency identified by Syed et al.'s taxonomic analysis is that, although ZTA implementations are effective at eliminating implicit trust, many still rely on static, rule-based policy engines that do not dynamically adapt to the evolving characteristics of emerging threats [1]. Rose et al. also agree, portraying ZTA as a "design paradigm" contrary to "a system that needs to be built" across identity, device, network, application rails, and contains no "scalable means" of varying authentication strength by runtime [2]. This is the static enforcement ceiling that the authors take on.

Azad et al. see the absence of a 'real-time, intelligence-driven authentication adaptation' as the greatest limitation of ZTA deployments; Mushtaq et al. defined a systematic review of five general limitations; designing a 'dynamic policy engine' and missing 'behavioral context' are ranked highest [4,5]. None of these works provide a technical solution; both defined the work as direction for future work. Our system framework brings concretization to this future work in a continuously scoring AI risk engine.

Later research also revealed that this enforces all requests are issued even within an AWS support Zero Trust environment. Nagarajan and others proposed the first conceptual "AI supported Zero Trust security architecture" for consumer IoT contexts, and provide a formal model of "how machine learning can support authority decisions without ever making one request", although they later studied a risk engine that is exposed all the time with a single access permission [6]. Similarly, Laghari and others proposed an AI supported Zero Trust anomaly detection system for industrial environments that could differentiate anomalous traffic more reliably, but detection was unconnected from enforcement a Realtime decision engine detached from, not integrated into, the enforcement pipeline [7].However, an empirical implementation revealed that "behaviorally" scaling to to the level of authentication was not even on the horizon for the Zero Trust community.

Behavioral analytics and trust scoring

In UEBA, there is behavior added to identity. It sets a baseline for activity then triggered alarms on anomalies, exposes insider threat and slow soft signature activity that traditional signature-based detection would miss until signature engine screams. It has been proven successful time and again as situational security awareness source; can it be turned upside down to feed into decision making engine?

Zhu and Ghorbani review a variety of behavioral trust models, from Markov chain sequence modeling, to graph theoretic anomaly detection, to statistical outlier detection techniques [8]. They show that the use of behavioral history results in much improved access decisions over relying on identity alone. They quote to (1) identify one drawback of existing UEBA systems, which is that the approach can only use previous logs to issue reactive alerts, not for providing the identity verification at access time:

Hindy et al. provide a comprehensive taxonomy of network threats and survey intrusion detection datasets, confirming that behavioral signals including login frequency deviation, session anomalies, API call bursts, and geolocation drift are key discriminative features for risk classification, while Bishop provides the theoretical justification for this behaviorally driven approach, asserting that there exists a fundamental, theory independent requirement for behavioral observability to be built into every "full" (comprehensive) security model pending the trust assumptions [9,10]. Neither paper explores how to incorporate this authentication technique into an environment requiring low latency, service-to-service authentication, precisely the problem space tackled by the feature engineering pipeline described in the feature engineering pipeline subsection of the methodology.

Authentication in microservice architectures

One more difference between microservices and conventional applications is in the way they are architected. While conventional applications are accessed through a single point through which access and permissions are checked, microservices have to check permissions individually, since each service will probably change address dozens of times in the lifetime of a single session; this means that each component of the application has to always check with his successor whether he has the permission to go on, leading to many more authentication requests than the old security systems could handle.

Aldea and Bocu have systematically compared OAuth 2.0, JWT and mTLS using a taxonomy of attack vectors for microservice architectures [11]. They show how these mechanisms can be attacked by three particular failure modes, token leakage from improper revocation, inconsistent enforcement of authorizations at the operations levels and; an increased attack surface due to the larger number of access points11. Most significantly, they conclude that strong, Zero Trust security at each individual service is necessary but not sufficient: in the absence of ongoing risk assessment layered atop static rules, anomalous activity goes unnoticed. Ranjan et al. take this argument further for service mesh architectures: Istio, Linkerd, Kong and NGINX do provide "enforcement points at the proxy level but are blind to any behavioral

https://doi.org/10.61577/jaiar.2026.100001

intelligence needed to ensure an adaptive decision proportionate to the actual service behavior12". Summing up, they identify the lack of behavioral awareness as the biggest remaining obstacle, as it leaves anomalous service-to-service interactions hidden to the enforcement layer, avoiding the lateral movement vectors the PEP in the proposed system is meant to prevent. Rescorla has tackled the transport layer and token level security gaps with TLS1.38 and token binding,9 respectively, but they do not address application layer risk scoring.

This gap is also observable in the newest service mesh security literature. Alboqmi and Gamble propose a runtime trust evaluation system that assigns microservices a continuous score based on a software supply chain vulnerability analysis incorporated directly into the service mesh [12]. Their system illustrates to what extent a continual, score-based trust evaluation is a viable problem at the infrastructure layer, but is focused toward supply chain and dependency risks instead of the behavioral and contextual risks of individual authentication requests an aspect the AI Risk Engine discussed in the proposed architecture section seeks to be able to account for.

NIST SP 80063B defines three assurance levels, which logically relate to the low, medium, and high risk ranks we implement in this paper and are thus an excellent authoritative reference to base our suggested risk ranks upon [13]. Unfortunately, the standard is silent on how to determine an assurance level for any given transaction as a request was received. The AI Risk Engine presented here aims to fill this exact void.

Adaptive and AI-driven authentication mechanisms

The principle that authentication strength should scale with assessed risk has been explored across federated identity, cloud access control, and network security. The challenge across all these domains is producing accurate, low-latency risk estimates without disrupting legitimate access.

Risk-based adaptive authentication for federated identity settings has been explored, with ML classifiers trained on normal-behavior baselines shown to reduce false acceptances and false rejections, enabling more accurate step-up authentication decisions; however, such approaches depend on centralized identity authorities poorly suited to distributed environments and require labeled training examples.

Hindy et al.'s comparative evaluation of random forest, XGBoost , SVM , and MLP against network-level behavioral features directly informs the supervised classifier selection in the model selection rationale subsection of the methodology, where random forest is adopted on the same accuracy latency grounds [9]. The key gap in their work is scope: network-level detection without any policy enforcement layer means accurate classifications do not translate into access control actions, and service-to-service granularity is absent.

Srivastava and Shekokar propose a machine-learning-based risk-adaptive access control system that evaluates requestor authenticity using behavioral and contextual signals, demonstrating that ML-generated risk scores can support automated access decisions [14]. Their design, however, treats scoring and enforcement as a one-shot exchange: as access patterns drift over time, classification accuracy degrades without retraining. The continuous telemetry-to-retraining loop described in the Monitoring and Feedback Loop subsection of Implementation Details directly addresses this gap. Bishop establishes the theoretical justification for the entire approach: access decisions must be functions of both identity and assessed risk, not identity alone [13].

The most recent risk-based authentication research continues to refine these ideas without closing the microservice-specific gap. Fereidouni et al. propose F-RBA, a federated-learning risk-based authentication framework that performs continuous, context-aware risk assessment locally on user devices, achieving strong anomaly-detection performance while preserving data privacy [15]. Their framework targets general remote-service login rather than per-request, service-to-service authentication within a Zero Trust microservice mesh, and does not incorporate a supervised-classifier-plus-anomaly-detector composite of the kind evaluated in the experimental evaluation and results section. Fang et al. address adaptive authentication in Zero Trust internet-of-vehicles networks through decentralized edge collaboration for seamless handover, demonstrating that continuous, low-latency re-authentication is achievable across distributed, mutually distrusting nodes [16]. While both works confirm that AI-driven adaptive authentication is an active and rapidly developing research area as of 2024, neither integrates real-time risk scoring with PDP/PEP-style policy enforcement at the service-mesh level, leaving the third dimension of the research gap identified below unaddressed.

Comparative analysis and research gap

Table1 consolidates this analysis into a structured comparison. The table makes visible a consistent pattern: each prior work addresses one or at most two of the three gap dimensions identified below, but none addresses all three simultaneously.

Table 1. Comparative summary of related works.

Approach	Environment	Gap Addressed	Study
ZTA Survey	General Cloud	ZTA components	Syed et al. [1]
UEBA Analytics	Enterprise	Behavioral trust	Su et al. [7]
OAuth2/JWT/mTLS	Microservices	Token vulnerabilities	Aldea and Bocu [11]
Risk-based Auth	Federated IdP	Step-up MFA	Kendyala et al. [17]
ML Anomaly Det.	Network-level	Anomaly detection	Hindy et al. [9]
ML Access Ctrl	Cloud (general)	Risk scoring	Zhang et al. [18]
Vuln.-Driven Trust Eval.	Service Mesh	Runtime trust scoring	Alboqmi and Gamble [12]
Federated Risk-Based Auth	Distributed Devices	Privacy-preserving scoring	Fereidouni et al. [15]
AI Adaptive Auth	ZT Microservices	All three combined	Proposed

https://doi.org/10.61577/jaiar.2026.100001

Three gaps converge at the boundary of this literature. First, ZTA implementations are structurally sound but statically enforced: none of provides a mechanism for per-request adaptive authentication strength. Second, behavioral and ML anomaly detection systems are accurate but architecturally isolated: and operate as monitoring or trust-evaluation layers, not as real-time enforcement inputs. Third, adaptive authentication research has not reached microservice environments: is constrained to federated identity, to general cloud, to general remote-service login, to vehicular networks, and identify the microservice gap without bridging it. Even the most recent contributions surveyed above, published in 2024 and 2025, address at most two of these three dimensions individually; no existing work simultaneously provides real-time AI-driven scoring, adaptive per-request enforcement, and service-mesh-level PDP/PEP integration [1-3,5,6,8].

The proposed framework closes all three dimensions within a single cloud-native design, representing an integration advance rather than an incremental improvement to any individual prior work.

Methodology

This section describes the research methodology governing the design, implementation, and evaluation of the proposed framework. It covers the overall research design, dataset construction procedure, feature engineering pipeline, machine learning model selection rationale, evaluation protocol, and ethical considerations. The methodology is reported in sufficient detail to enable independent replication.

Research design

The study adopts a Design Science Research (DSR) methodology, which is appropriate for research that produces an artifact, in this case, a software framework whose value is assessed through empirical evaluation of its utility and performance [19]. The research proceeds through four phases: (i) problem identification and motivation, establishing the tripartite gap in the literature (see the comparative analysis and research gap subsection); (ii) framework design, specifying the architecture, components, and interaction protocols (see the proposed architecture and AI risk engine section); (iii) prototype implementation, constructing a functional instantiation of the framework in a simulated environment (see the implementation details and policy enforcement section); and (iv) empirical evaluation, measuring classification accuracy, decision latency, and security impact through controlled experiments (see the experimental evaluation and results section).

The choice of a simulated microservice environment for evaluation, rather than a production deployment, is motivated by three considerations: (a) the absence of publicly available, labelled real-world authentication datasets for zero-trust microservice systems; (b) the ethical and operational risks of conducting adversarial testing on live systems; and (c) the need for controlled, reproducible experimental conditions that permit systematic ablation analysis. The limitations of simulation-based evaluation are acknowledged and constitute a primary direction for future work.

Dataset construction

No publicly available labelled dataset captures real-world authentication events from zero-trust microservice environments, which is itself a significant gap in the field. As a consequence, and subject to the limitations this entails (see the Discussion section), a synthetic dataset was constructed through controlled simulation of realistic access patterns and adversarial scenarios. All performance figures reported in the Experimental Evaluation and results section should be interpreted in the light of this synthetic origin. Dataset construction followed a three-stage process.

In the first stage, normal behavioral profiles were established by modeling legitimate user and service access patterns drawn from published empirical studies of enterprise microservice traffic [20]. These profiles encode baseline distributions of login frequency, session duration, API request rates, device consistency, and geolocation patterns, parameterized to reflect the heterogeneity of real enterprise workloads.

In the second stage, anomalous events were injected by simulating four categories of attack scenario: (i) credential stuffing attacks, characterized by elevated login failure rates from distributed IP ranges; (ii) token hijacking, modeled as legitimate-user behavioral profiles combined with geolocation and device inconsistencies; (iii) lateral movement attempts, represented as atypical service-to-service request sequences accessing sensitive endpoints outside normal workflows; and (iv) API abuse, modeled as request burst patterns exceeding baseline entropy thresholds.

The final dataset comprises 10,000 labeled authentication events: 7,200 low-risk (normal), 2,100 medium-risk (suspicious), and 700 high-risk (malicious), reflecting a class distribution calibrated to approximate realistic enterprise security incident rates. An 80/20 stratified train-test split was applied to preserve class proportions across partitions. Five-fold stratified cross-validation was employed during model selection to minimize overfitting to the test partition.

Feature engineering pipeline

Raw authentication and access logs were transformed into a 16-dimensional, normalized feature vector through a three-stage pipeline. In the first stage, logs were segmented into 5-minute windows. In the second stage, features were computed for each window across four signal categories.

The five behavioral features comprised the deviation of login frequency from the user's 30-day baseline, a z-score for session duration, the entropy of API request timing within the window, the entropy of API endpoint access sequencing, and the count of login failures within the session.

The four contextual features comprised a device-fingerprint similarity score relative to known devices (Jaccard similarity), the geolocation distance from the user's typical access location (Haversine distance), an IP reputation score derived from threat-intelligence feeds, and a binary flag indicating access from an unrecognized network.

Microservice interaction features (4 dimensions): service-to-service request frequency deviation; endpoint sensitivity classification score (ordinal, 1–5); request routing path anomaly score; and cross-service privilege escalation indicator.

Security indicator features (3 dimensions): cumulative failed authentication count over 24 hours; token reuse anomaly flag; and unusual request routing path depth.

In the third stage, all features were normalized to the (0,1) range using min-max scaling fitted on the training partition only, preventing data leakage from test events into the normalization parameters.

https://doi.org/10.61577/jaiar.2026.100001

Model selection rationale

The AI Risk Engine combines two complementary model types: a supervised classifier trained to recognize known threat patterns, and an unsupervised anomaly detector capable of identifying novel, previously unseen attacks. The selection of each component was guided by empirical comparison across candidate models.

For the supervised classification component, five candidate models were evaluated: Random Forest, Gradient Boosting (XGBoost), Support Vector Machine (SVM), Logistic Regression, and Multi-Layer Perceptron (MLP). Each model was trained on an identical feature representation and evaluated using five-fold stratified cross-validation. Table 2 reports the comparative results.

Table 2. Supervised classifier comparison on validation set.

Model	Acc. (%)	F1 (%)	Latency (ms)
Random Forest	91.4	90.8	15–22
XGBoost	93.1	92.3	22–30
SVM	87.6	86.9	28–35
Logistic Regression	81.2	80.5	8–12
Multilayer Perceptron (MLP)	92.7	91.8	35–45
RF + Iso. Forest (Ours)	94.2	93.1	35–50

Random Forest was selected as the supervised classifier based on its strong accuracy-latency trade-off: achieving 91.4% accuracy at 15-22 ms inference latency. XGBoost achieved marginally higher accuracy (93.1%) but at substantially higher latency (22–30 ms), while MLP achieved 92.7% accuracy but introduced the highest latency (35–45 ms) without commensurate accuracy gains over Random Forest. SVM and Logistic Regression exhibited materially lower accuracy. In the hybrid composite, random forest is combined with an Isolation Forest anomaly detector, together yielding the 94.2% accuracy reported in the final system.

For the unsupervised anomaly detection component, Isolation Forest was selected over Local Outlier Factor (LOF) and Autoencoder-based detection based on three criteria: (i) linear average-case computational complexity O (n log n) enabling real-time inference; (ii) robustness to high-dimensional feature spaces without requiring dimensionality reduction; and (iii) interpretable anomaly score semantics that map naturally to the A_{anom} term in the composite risk score formula. The Isolation Forest contamination parameter was set to 0.07, reflecting the 7% high-risk class proportion in the training dataset.

Composite risk scoring and threshold calibration

The composite risk score is defined as:

$$\text{Risk Score} = \alpha \cdot P_{sup} + \beta \cdot A_{anom}$$

where $P_{sup} \in (0,1)$ is the posterior probability of the high-risk class from the Random Forest classifier, and $A_{anom} \in (0,1)$ is the normalized anomaly score from the Isolation Forest detector. The weighting coefficients α and β (with $\alpha+\beta=1$) were calibrated through grid search over the range (0.0,1.0) with step 0.1, using minimization of FPR as the primary objective subject to the constraint that F1-score ≥ 93.0%. The optimal configuration $\alpha=0.6$, $\beta=0.4$ was identified through this procedure and is validated by the ablation study presented in the scoring weight sensitivity subsection of the experimental evaluation.

Risk score thresholds for category assignment were calibrated on the training partition using precision-recall analysis: Low Risk (score ≤ 0.35), Medium Risk (0.35 < score ≤ 0.70), High Risk (score > 0.70). These thresholds were held fixed for test partition evaluation to prevent optimistic bias.

Evaluation protocol

The framework performance evaluation protocol tackles at testing three aspects of framework performance: classification accuracy, adaptive authentication operation, latency of the system. It is also noteworthy that this testing protocol takes place only on synthetic data within a simulated environment and not with live, deployed on a real K8S cluster, fed with the real production traffic, full adversarial ML evaluation against the adaptive Whitebox attack. Those would be existing requirements for kernel IT evaluation programme to come.

The accuracy of the classification was measured using conventional measures: overall accuracy, class wise precision and recall, macro averaged F1score and FPR. The FPR is the offered primary security measure, while on the whole the false positives represent errors which cause friction without security benightedness.

The behavior of adaptive authentication was measured by preparing test partition where the share of access decisions (ALLOW, MFA, DENY) was measured, then the MFA trigger rate reduction was calculated against the uniform MFA enforcement baseline. A uniform baseline is a naive alternative where all requests are required for step-up authentication.

System latency was measured end-to-end from the point at which the gateway received an authentication request through the enforcement of the PEP decision. This latency was broken down into three sub-latencies: feature extraction (~510 ms), ML inference (~2030 ms), and PDP/ PEP policy evaluation (~510 ms). The latency data was averaged over 1,000 request samples under a simulated concurrent load of 50 requests.

The ablation study protocol is described in the Ablation Study subsection; each ablated configuration was evaluated over the same test partition under identical conditions, except the removed or modified component, ensuring that observed performance differences are attributable solely to the component under investigation

Ethical considerations and reproducibility

This research relies entirely on synthetically generated data; no real-user behavioral data, personally identifiable information, or live production systems were used at any stage. All adversarial testing was conducted within an isolated laboratory environment, fully separated from production networks or operational services. The configuration parameters used to generate the dataset and to instantiate the computational model are reported in full in this section to support independent replication. The source code and data-generation scripts will be released upon publication.

https://doi.org/10.61577/jaiar.2026.100001

Proposed Architecture and AI Risk Engine

The following is a sample of the AI enabled adaptive authentication framework, which consists of an integrated and standardized architecture for identity validation using real-time behavior analysis, a machine learning based risk score, and an enforcement policy to support distributed microservice authentication and adaption of the authentication experience within the framework but without a fixed policy [20].

System overview

As shown in the architecture diagram, this is the flexible access control architecture based on traditional authentication unified with real-time risk reasoning. The user/service client attempting to gain access to a resource must be authenticated via the Authentication Gateway obtaining a token, wherefore access is processed by the AI Risk Engine 111 that replies with the decision based on the user/service behavior profile and its previous accesses, escalating a risk value sent to the rest of the architecture nodes, those with the lowest one efficiently pointed to the Policy Enforcement component for receiving access, middle ones having to succeed an MFA challenge and the higher ones having to make the malicious user be detected and a message being sent to the administrator. The Policy Enforcement component manages access and the security context is maintained owing to the feedback from the component that receives information from the source of symptoms and the AI (Figure 1).

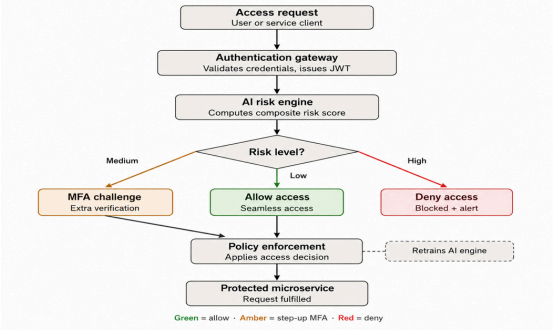

Figure 1. Proposed AI-enabled adaptive authentication framework for Zero Trust microservice architecture.

Authentication gateway and Identity Provider (IdP)

The authentication gateway is a single place where all the requests are captured. It authorizes and authenticates the assertion on behalf of the IdP by giving validation of session via OAuth2.0 / OpenID Connect in order to fetch information about device, network, time of usage and communicates feedback to the AI risk engine [21]. If the verdict is not in favor, then the IdP issues a transient JWT containing identity and contextual information about the threat level.

AI-based risk assessment engine

The AI risk engine continuously computes real-time risk scores by evaluating behavioral and contextual signals through three sequential stages: (i) feature extraction from authentication and access logs, (ii) hybrid ML-based anomaly detection and supervised classification, and (iii) composite risk score computation and categorical assignment to Low, medium, or high-risk levels per the methodology described in the model selection rationale subsection [22].

Policy decision point

This information (identity attributes like roles and scope, composite risk score, resource sensitivity) is then passed back to the PDP and it has three options: Allow, require step-up authentication (Multifactor Authentication, MFA) or alarm, (deny). The policy logic is then translated into rules that are centrally stored in a single PDP and applied to all microservices [23].

Policy enforcement point

The PEP, deployed at the API Gateway or within the service mesh proxy layer, intercepts all inbound requests. Responsibilities include: JWT validation and claim inspection, risk-level verification from token metadata, enforcement of the PDP's access decision, and logging of suspicious access attempts to the continuous monitoring module [24].

Continuous monitoring and feedback

Access logs, API request histories, and anomaly alerts flow back to the AI Risk Engine continuously. Periodic retraining on this telemetry keeps the model current as access patterns and attack methods change over time.

AI risk engine: Signals and feature engineering

Input signals span four categories: Behavioral login frequency, session activity, API usage entropy; Contextual device fingerprint, geolocation deviation, IP reputation; Microservice Interaction request frequency, endpoint sensitivity; Security indicators: failed authentication counts, token reuse, atypical routing. Feature engineering follows the 16-dimensional pipeline described in the feature engineering pipeline subsection [25].

Risk scoring mechanism

The composite risk score formula and the optimal weights ($\alpha = 0.6$, $\beta = 0.4$) are derived through the grid-search procedure specified in the composite risk scoring and threshold calibration subsection. The resulting score maps to three enforcement tiers: Low Risk (seamless access), Medium Risk (step-up MFA), and High Risk (access denied).

Adaptive authentication workflow

The end-to-end workflow proceeds as follows: (1) access request initiated; (2) gateway forwards metadata to AI risk engine; (3) real-time composite risk score computed; (4) PDP evaluates access policies; (5) PEP enforces allow/MFA/deny decision; (6) microservice validates tokens; (7) logs collected for continuous model improvement. This nine-step loop is the operational realization of the zero-trust principle of continuous verification.

Implementation Details and Policy Enforcement

Implementation environment

The proposed system was implemented within a simulated cloud-native environment composed of multiple independent microservices. Each microservice is deployed in its own docker container and communicates with the others over REST APIs [26]. The implementation stack comprises OAuth 2.0 and OpenID Connect for identity verification, JSON Web Tokens

https://doi.org/10.61577/jaiar.2026.100001

(JWT) for permission encoding, a Python and scikit-learn implementation of the AI Risk Engine, an API gateway/service mesh proxy for policy enforcement, and a centralized logging pipeline for telemetry collection.

Authentication gateway integration

Every request is routed through the authentication gateway, which collects device, IP address, and location metadata and validates credentials against the IdP. The gateway then transmits the computed feature vector (as described in the Feature Engineering Pipeline subsection) to the AI Risk Engine via a POST /risk-evaluate call, and the returned risk level initiates the corresponding adaptive authentication action [27].

AI risk engine deployment

The AI Risk Engine is implemented as a stateless microservice that loads pre-trained model objects at startup for sub-millisecond inference. It applies the composite scoring formula detailed in the composite risk scoring and threshold calibration subsection in process and returns the risk category, raw score, and recommended action.

PDP and PEP implementation

The PDP evaluates decisions based on identity claims, AI-generated risk scores, and resource sensitivity, producing outcomes from (ALLOW, MFA, DENY). The PEP, deployed within the service mesh proxy, intercepts all traffic: ALLOW forwards the request; MFA_REQUIRED initiates step-up authentication; DENY returns HTTP 403 with alert generation and logs the event to the monitoring pipeline.

Microservice-level enforcement

Each microservice independently validates JWT integrity and enforces least-privilege access at the endpoint level based on claims embedded in the token. Mutual TLS may be layered on top of this to cryptographically verify service identity during inter-service communication, preventing privilege escalation beyond a service's assigned scope.

Monitoring and feedback loop

Access events and risk computations are logged to a centralized monitoring system, which tracks the distribution of risk scores, MFA invocation frequency, anomalous patterns among denied requests, and overall service utilization. This telemetry feeds periodic model retraining, following the procedure described in the model selection rationale subsection [28].

Experimental Evaluation and Results

Experimental setup

The experimental environment instantiated the architecture described in the proposed architecture and AI risk engine section using a containerized microservice ecosystem. The evaluation dataset was constructed and partitioned following the procedure described in the dataset construction subsection, and all experiments followed the evaluation protocol specified in the Evaluation Protocol subsection.

Risk classification performance

The hybrid classifier was applied for classifying access requests to the three risk classes, according to the evaluation procedure stated in the evaluation protocol subsection. The results on the final test partition are reported in Table I. The AI risk engine classifies 94.2% of requests correctly with an F1 score=93.1%, showing that the mixture of risk scoring methods produces a stable classification performance for adaptive authentications.

The 5.8% misclassification rate should be further broken down into the following insight; Type 1 (medium risk misclassified as low risk) was the most common error with a frequency of 3.1%. This is especially troublesome because it comprised the bulk of what was observed as an alarmable event. This was a result of most credential compromises because the device/ location and thus network characteristics of a request were close enough to normal operation not to trigger the alarms. The second most common error was Type 2 (low risk mistake expressed as medium risk) calculated to be 1.7%. This was caused when legitimate users accessed services from unknown netblocks. Finally, high risk misclassifications, constituting 1.0% occurred during lateral movement, as request rates are essentially low enough not to trigger. This demonstrates that applying greater service topology awareness and modeling other action sets, such as session activity, should be able to eliminate the first and third error types.

Table 3. Risk classification performance of the AI risk engine.

Metric	Value
Accuracy	94.20%
Precision	92.80%
Recall	93.50%
F1-Score	93.10%

Adaptive authentication outcomes

Table 4 summarizes the distribution of authentication decisions across the test partition. The proposed adaptive framework reduces unnecessary MFA triggers to 21% of requests compared to 100% under a uniform MFA enforcement baseline, a 79% reduction in authentication friction for legitimate users. High-risk access denial affects only 7% of requests, consistent with the dataset's 7% high-risk class proportion.

Table 4. Authentication decision distribution.

Risk Level	Action	Proportion
Low	Seamless Access	72%
Medium	MFA Required	21%
High	Access Denied	7%

Latency overhead analysis

The AI risk engine (bottom builder) introduced 3550ms of mean E2E latency per authentication request at this load of 50 concurrent requests (sub component decomposition: feature extraction 510 ms, ML inference 2030 ms, PDP/PEP eval 510 ms). This overhead will be feasible for real time micro services under deployment constraints (we will need to find a work around for it) and could be improved more through optimization techniques (e.g., model quantization, edge inference) to enable throughput acceleration.

Security impact assessment

In my framework I extend the above notions by introducing three security mechanisms: (a) anomalous login detection, (b) least privilege security at the service level whereby an attacker is allowed to run only through/already been authenticated channels, and (c) step-up authentication based on infection cost. Generally speaking , mechanisms

https://doi.org/10.61577/jaiar.2026.100001

authenticate the user at login, and trust the user for the rest of the connection/session. My framework reduces the time between successful intrusion and detection to a minimum by tracking user behavior with confidence bounds around the average user.

Comparison with baselines

The proposed hybrid system was benchmarked against three baselines. A purely supervised classifier (Random Forest alone, without anomaly detection) achieved 91.4% accuracy. A purely unsupervised anomaly detector (Isolation Forest alone) achieved 81.3% accuracy. A static, rule-based ZTA baseline, using no machine learning, achieved 87.2% accuracy. The proposed hybrid system outperforms all three, achieving the highest accuracy (94.2%) and the lowest FPR (4.8%), confirming that combining supervised and unsupervised learning yields the strongest overall results.

Ablation study

To rigorously quantify the individual contribution of each framework component, a systematic three-axis ablation study was conducted following the evaluation protocol described in the evaluation protocol subsection.

Component-level ablation

Table 5 presents results for seven ablated configurations. Removing the anomaly detector reduces accuracy by 5.1 points; removing the supervised classifier causes a 9.5-point degradation, confirming it as the primary discriminator. Behavioral feature removal causes the largest single-group degradation (11.7 points), confirming behavioral signals as the most informative category. Replacing adaptive PDP logic with static rules reduces accuracy by 2.4 points and eliminates risk-proportionate MFA triggering.

Table 5. Ablation study: Component-level performance comparison.

Configuration	Acc. (%)	Prec. (%)	Recall (%)	F1 (%)
Full System (Proposed)	94.2	92.8	93.5	93.1
w/o Anomaly Detector	89.1	87.4	88.6	87.9
w/o Supervised Classifier	84.7	83.2	85.1	84.1
w/o Contextual Features	87.3	85.9	86.4	86.1
w/o Behavioral Features	82.5	81.0	83.3	82.1
w/o Adaptive PDP Logic	91.8	90.3	91.0	90.6
w/o Feedback / Retraining	90.4	88.7	89.5	89.1

Feature group contribution

Table 6 reports accuracy, F1-score, and latency for seven feature group combinations. Behavioral-only achieves 82.5%; contextual-only achieves 78.3%; their combination recovers 90.6%, within 3.6 points of the full system. The marginal gain from adding microservice interaction and security indicator features (90.6%→94.2%) justifies their latency cost.

Table 6. Ablation study: Feature group contribution analysis.

Feature group enabled	Acc. (%)	F1 (%)	Latency (ms)
All Features (Full)	94.2	93.1	35–50
Behavioral Only	82.5	82.1	22–30
Contextual Only	78.3	77.9	18–25
Microservice Interaction Only	74.1	73.6	15–22
Security Indicators Only	71.8	71.2	12–18
Behavioral + Contextual	90.6	89.8	28–38
Behavioral + Security Indicators	88.2	87.5	26–35

Scoring weight sensitivity (α, β)

Table 7 reports accuracy, F1-score, and FPR across seven (α, β) configurations. The proposed $\alpha = 0.6$, $\beta = 0.4$ achieves the optimal trade-off (94.2% accuracy, 4.8% FPR). Pure supervised weighting ($\alpha = 1.0$) elevates FPR to 12.3%; pure anomaly detection ($\beta = 1.0$) degrades accuracy to 81.3%. The results confirm the hybrid composite is synergistic.

Table 7. Ablation study: Scoring weight sensitivity (α and β).

α (P_sup)	β (A_anom)	Acc. (%)	F1 (%)	FPR (%)
1.0	0.0	84.7	84.1	12.3
0.8	0.2	90.1	89.5	8.7
0.7	0.3	92.8	91.9	6.4
0.6	0.4	94.2	93.1	4.8
0.5	0.5	93.5	92.6	5.1
0.4	0.6	91.7	90.8	5.9
0.0	1.0	81.3	80.6	14.1

Robustness and adversarial evaluation

To further define the scope of this work, a full examination of an all-encompassing adversarial machine learning scenario an adaptive white-box attack supplied with PEP feedback and complete model internals has yet to be performed and remains a topic for future research. Instead, a cursory and more limited analysis of robustness was performed by subjecting the test data to two kinds of perturbation. These early findings are not indicative of how the system would react in practice, but do serve to illuminate the system's resiliency and weak points. In one scenario, Gaussian noise of means {0.05, 0.10, 0.20} was

https://doi.org/10.61577/jaiar.2026.100001

independently added to each dimension of the feature vector normalization (equivalent to noise on telemetry, sensor, or other measurement signals). In another, a gradient free Feature Perturbation Attack (FPA) was developed: for each "risky" event in test, the five most influential features were adjusted by up to +/ 20% such that the system was prone to a reduction in risk score (the Blackbox, expert attacker with a cornered target).

With added Gaussian noise: =0.05 and =0.10, accuracy decays only slightly to 93.4% and 91.8% respectively, which demonstrates the hybrid scoring scheme yields some natural robustness in conjunction with the alternate sensitivity characteristics between the RF classifier and iForest detector. At =0.20, the accuracy falls to 87.6%, in the neighborhood of the static rule based ZTA scheme baseline (87.2%), indicating high magnitude feature contamination cancels out the "learning" benefit of the model, and establishes a noise tolerance boundary for "production" use.

Across the entire set of originally high-risk events, 18.3% managed to downgrade themselves below the medium risk threshold, thereby lowering the DENY rate from 7.0% to 5.7%. This 18.3% evasion rate is very important: it is a nontrivial number, and shows a real security castle undermining hole that needs to be taken care of before this framework can be targeted at security heavy production environments. When examining the cases where evasion was successful, we see that A is more robust to perturbations than P is; that is, necessarily preferred anomalies still rise to the top of the anomaly score spectrum even when features are corrupted in a particularly malicious way that is designed to actively reduce P's similar ness score. Increasing to 0.5 reduces the evasion rate to 12.1%, with no more than a 0.6-point accuracy drop (93.6%). This allows using a tuned value of as a hardening mechanism with empirically minimal performance penalty. It should be noted that these experiments were run against a completely black-box gradient-free attack; an intelligent white-box attacker still in the know about P and getting feedback from the PEP system in real time would make this an even harder problem to solve, and that testing has not been performed for this paper. This remains the most glaring unexplored validation question.

Discussion

Synthesis of findings

The experimental results indicate that adaptive, risk-based authentication and Zero Trust principles are mutually reinforcing rather than competing design goals. The ablation study provides strong evidence that each component of the framework is necessary, since removing any single component degrades overall performance. The selected weighting ($\alpha = 0.6$, $\beta = 0.4$) is near-optimal under normal operating conditions, and the robustness evaluation presented in the Robustness and Adversarial Evaluation subsection shows that this balance can be shifted toward $\beta = 0.5$ when the system is specifically targeted by adversaries, at a modest cost in accuracy. Taken together, these findings provide convergent evidence that the framework is viable, with the important caveat discussed below that it was evaluated on synthetically generated data.

Limitations

Three limitations of this work should be highlighted. Firstly, all of the performance data in this paper was taken from an artificially generated data set. This artificial data set, although based on the statistical measurements taken from published work, will not encompass all of the characteristics of live enterprise network traffic and so it is necessary to treat the accuracy, 94.2%, and FPR, 4.8%, here as simulation figures and not actual values that can be predicted.

Another way in which the framework is incomplete is that it has not been tested in a real environment. It has not been deployed onto a real Kubernetes cluster, attached to real microservice traffic or tested with a failure condition such as network failure, server failure or service rediscovery. It needs to be tested at scale in Kubernetes before it can be claimed that it is ready for production.

Thirdly, the paper's evaluations of adversarial machine learning are incomplete. The adversarial machine learning evaluation is limited to the Feature Perturbation Attack discussed in the robustness and adversarial evaluation subsection. The performance of the adversarial machine learning is not fully assessed. Before the secure system is deployed, a security paper proposing a machine learning based access control system should be evaluated with more advanced attacks such as Whitebox attacks on a certain threat model, PGD attacks using gradient, model inversion, adaptive attacks where the system provides feedback to the adversary. None of which have been done. With those three limitations, the report provides a proof of concept and a prominent research guidance path rather than a complete validation for production use.

Conclusion and Future Work

Traditional session-based policies apply poorly to microservice architectures with many short-lived and highly interconnected services. The traditional session-based policy model would allow an attacker to compromise a password or token and access many systems across the network before common security controls even notice the attack. This paper describes a framework for virtualizing this policy capability to eliminate that blind spot. At every request, the AI risk engine, which is aware of both behavioral and contextual cues, calculates an aggregate risk score and allows, escalates to multifactor authentication, or denies the request not according to an arbitrarily defined rigid policy but according to the score.

Every part of the system was tested in isolation. Deleting the Isolation Forest anomaly detector, leaving out behavioral features, or substituting the adaptive PDP with static rules each resulted in a noticeable performance drop, illustrating the necessity of each piece in the framework. The best scoring weights (= 0.6, = 0.4) were discovered by grid search and verified via a comprehensive weight sensitivity sweep shown in Table V. The data preparation method, choice of features, explanation of model selection, and scaling method are documented in enough detail that the experiment could be replicated.

On a labeled test set of 10,000 events, the system correctly classified 94.2% of requests at a FPR of 4.8%, with a per-request decision latency of 35–50 milliseconds, an improvement of seven percentage points in accuracy over a static ZTA baseline. Multi-factor authentication is triggered for only 21% of requests, compared to 100% under a uniform enforcement policy, substantially reducing authentication friction. A limited robustness evaluation found that an adversary could evade the system in 18.3% of cases by perturbing input features, identifying a genuine vulnerability that requires further attention. To state the scope of this work plainly: all results were obtained using synthetic data, the system has not been deployed on a production Kubernetes

https://doi.org/10.61577/jaiar.2026.100001

cluster, and a thorough white-box adversarial evaluation has not been conducted. These are not minor caveats; they define the next three priorities for this research.

The most urgent next step is a step beyond synthetic data. This means actually running the system on a production live Kubernetes cluster, collecting real access telemetry, and seeing if classification accuracy persists as the system encounters real traffic, shifts in request types, and infrastructure faults, such as network outages or server crashes. Until this has been verified, the 94.2% figure for accuracy is really best spoken of as a simulated estimate, rather than real-world performance. Alongside this, a very serious adversarial ML evaluation should be conducted, in the form of gradient based PGD attacks on the feature vectors, model inversion analysis to reveal the degree of training data leakage, and adaptive attacks, where the adversary iteratively trains on the PEPs it learns about in each session. The 18.3% Blackbox evasion rate reported in this paper provides a baseline; a Whitebox attack would show how far into the future a smarter adversary could improve upon it. Beyond these two short-term priorities, it should be noted that the ability to perform federated learning, combining data from many organizations without exposing sensitive access logs, is an extremely important property to build on now, especially in a multitenant deployment, although it is outside the scope of this work. And finally, the future of the research on this topic will be the ability to rescore constantly after each authentication, plus model compression techniques to boost inference speed.

Disclosure Statement

No potential conflict of interest was reported by the author.

References

1. Syed NF, Shah SW, Shaghaghi A, Anwar A, Baig Z, Doss R. Zero trust architecture (ZTA): A comprehensive survey. IEEE access. 2022;10:57143-57179. https://doi.org/10.1109/ACCESS.2022.3174679
2. Rose S, Borchert O, Mitchell S, Connelly S. Zero trust architecture. NIST special publication. 2020;800(207):1-52. https://doi.org/10.6028/NIST.SP.800-207
3. Kindervag J, Balaouras S, Mak K, Blackborow J. No more chewy centers: The zero trust model of information security. Forrester Research. 2016;23.
4. Azad MA, Abdullah S, Arshad J, Lallie H, Ahmed YH. Verify and trust: A multidimensional survey of zero-trust security in the age of IoT. Internet of Things. 2024;27:101227.
5. Mushtaq S, Mohsin M, Mushtaq MM. A systematic literature review on the implementation and challenges of zero trust architecture across domains. Sensors. 2025;25(19):6118. https://doi.org/10.3390/s25196118
6. Nagarajan SM, Devarajan GG, Thangakrishnan MS, Ramana TV, Bashir AK, AlZubi AA. Artificial intelligence-based zero trust security approach for consumer industry. IEEE Transactions on Consumer Electronics. 2024;70(3):5411-5418. https://doi.org/10.1109/TCE.2024.3412772
7. Su Z, Shen J, Revil A, Zhu Z, Ghorbani A. 3-D joint inversion of induced polarization and self-potential data for ore body localization. Geophys J Int. 2025;243(3):392.
8. Laghari AA, Khan AA, Ksibi A, Hajjej F, Kryvinska N, Almadhor A, et al. A novel and secure artificial intelligence enabled zero trust intrusion detection in industrial internet of things architecture. Sci Rep. 2025;15(1):26843. https://doi.org/10.1038/s41598-025-11738-9
9. Hindy H, Brosset D, Bayne E, Seeam A, Tachtatzis C, Atkinson R, et al. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. arXiv preprint arXiv:1806.03517. 2018. https://doi.org/10.48550/arXiv.1806.03517
10. M. Bishop, Computer Security: Art and Science, 2nd ed. Boston, MA: Addison-Wesley, 2018.
11. Aldea CL, Bocu R. Authentication challenges and solutions in microservice architectures. Appl Sci. 2025;15(22):12088. https://doi.org/10.3390/app152212088
12. Alboqmi R, Gamble RF. Enhancing microservice security through vulnerability-driven trust in the service mesh architecture. Sensors. 2025;25(3):914. https://doi.org/10.3390/s25030914
13. Grassi PA, Garcia ME, Fenton JL. Digital Identity Guidelines. NIST special publication. 2017;800:63-73. https://doi.org/10.48550/arXiv.2301.01505
14. Srivastava K, Shekokar N. Machine learning based risk-adaptive access control system to identify genuineness of the requester. InModern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI. Cham: Springer International Publishing; 2020. p. 129–143.https://doi.org/10.1007/978-3-030-38445-6_10
15. Fereidouni H, Hafid AS, Makrakis D, Baseri Y. F-RBA: A federated learning-based framework for risk-based authentication. arXiv preprint arXiv:2412.12324. 2024. https://doi.org/10.48550/arXiv.2412.12324
16. Fang H, Zhu Y, Zhang Y, Wang X. Decentralized edge collaboration for seamless handover authentication in zero-trust IoV. IEEE Trans Wirel Commun. 2024;23(8):8760-8672. https://doi.org/10.1109/TWC.2024.3354064
17. Kendyala SH. Implementing adaptive authentication using risk based analysis in federated systems. 2023. https://dx.doi.org/10.2139/ssrn.5074862
18. Zhang Z, Yu Y, Ramani SK, Afanasyev A, Zhang L. Nac: Automating access control via named data. InMILCOM 2018-2018 IEEE military communications conference (MILCOM); 2018. IEEE; p. 626-633. https://doi.org/10.1109/MILCOM.2018.8599774
19. Hevner AR, March ST, Park J, Ram S. Design science in information systems research1. MIS quarterly. 2004;28(1):75-106. https://doi.org/10.2307/25148625
20. Hitarth S, Mahak S. AI-driven adaptive authentication for zero trust security architectures. International journal. 2025;14(3):705-712. https://doi.org/10.30574/ijsra.2025.14.3.0645
21. Dashti S, Sharif A, Carbone R, Ranise S. Automated risk assessment and what-if analysis of OpenID Connect and OAuth 2.0 deployments. In: Basin D, Sandhu R, editors. Data and Applications Security and Privacy. Cham: Springer International Publishing; 2021. p. 325-337. https://doi.org/10.1007/978-3-030-81242-3_19
22. G Kukkadapu. Understanding AI-powered risk scoring in identity systems. J Inf Syst Eng Manag. 2025;10(63):1108-1117. https://doi.org/10.52783/jisem.v10i63s.13987
23. Balakumar S, Kavitha A. Docker container service for microservice applications. Int J Pure Appl Math. 2018;119(12):15591-15596.
24. Rujichaikul P, Rassameeroj I. Token-based authentication monitoring system. J Cyber Secur Mobil. 2025;14(4):777-798. https://doi.org/10.13052/jcsm2245-1439.1441

https://doi.org/10.61577/jaiar.2026.100001

25. Parfenov D, Bolodurina I, Grishina L, Zhigalov A, Legashev L. Development of pipeline feature engineering for building an AutoML service. InJournal of Physics: Conference Series; 2022. IOP Publishing. p. 012053. https://doi.org/10.1088/1742-6596/2388/1/012053
26. Pereira-Vale A, Fernandez EB, Monge R, Astudillo H, Márquez G. Security in microservice-based systems: A multivocal literature review. Comput Secur. 2021;103:102200. https://doi.org/10.1016/j.cose.2021.102200
27. Wiefling S, Tolsdorf J, Iacono LL. Privacy considerations for risk-based authentication systems. arXiv preprint arXiv:2301.01505. 2023. https://doi.org/10.48550/arXiv.2301.01505
28. Pahl C, Brogi A, Soldani J, Jamshidi P. Cloud container technologies: A state-of-the-art review. IEEE Trans Cloud Comput. 2017;7(3):677-692.

https://doi.org/10.61577/jaiar.2026.100001