An Enhanced Dynamic Zero-Trust Cloud Architecture Framework for Multi-Cloud and Edge Computing

Onwuachu Uzochukwu Christian[1] and Ndueso, Etukudo Ekefre[2]

[1]Imo State University, Owerri, Imo State. Nigeria

[2]University of Port Harcourt, Port Harcourt, Rivers State. Nigeria.

DOI: https://doi.org/10.51584/IJRIAS.2026.11100092

Received: 22 October 2026; Accepted: 27 November 2026; Published: 13 November 2026

ABSTRACT

The increasing adoption of multi-cloud environments and edge computing has introduced significant security challenges, including unauthorized access, identity compromise, inconsistent security policies, and the difficulty of monitoring distributed resources. Traditional perimeter-based security models are insufficient for modern distributed environments because they often assume that entities within a trusted network can be granted access with limited additional verification. This research proposes an Enhanced Dynamic Zero-Trust Cloud Architecture Framework for Multi-Cloud and Edge Computing. The proposed framework integrates continuous identity verification, device posture assessment, context-aware access control, risk-based authentication, centralized policy management, and distributed monitoring. It evaluates contextual information such as user identity, device health, access location, resource sensitivity, and observed behavior before granting or maintaining access to cloud and edge resources. The framework is designed to support heterogeneous cloud platforms and resource-constrained edge devices through policy enforcement points, secure communication channels, and coordinated security decisions. A prototype can be implemented using Python, Flask, and simulated cloud and edge nodes. Its effectiveness can be evaluated against a baseline access-control model using access-decision accuracy, unauthorized-access blocking rate, authentication latency, policy enforcement time, and resource overhead. The study aims to provide a scalable and adaptable security framework that improves access governance across distributed computing environments. Experimental results will be reported only after implementation and testing.

Keywords: Zero Trust, multi-cloud security, edge computing, dynamic access control, continuous authentication, cloud security, risk-based authorization.

INTRODUCTION

The rapid development of cloud computing, edge computing, and distributed information systems has transformed how organizations store, process, and manage data. Businesses, government institutions, healthcare organizations, educational institutions, and financial service providers increasingly depend on cloud-based infrastructure to support their daily operations. Cloud computing provides scalable storage, flexible computing resources, and on-demand access to applications, while edge computing enables data processing closer to the devices and users generating the data. The integration of these technologies has created complex computing environments in which applications and data are distributed across multiple cloud service providers, local data centres, and edge devices.

Despite the advantages of these technologies, their adoption has introduced significant security challenges. Multi-cloud environments require organizations to manage identities, access permissions, network configurations, and security policies across different cloud platforms. At the same time, edge computing expands the attack surface because edge devices may operate in remote locations, have limited computing resources, and connect through networks with varying levels of security. These conditions increase the possibility of unauthorized access, compromised credentials, data leakage, malicious activities, and inconsistent enforcement of security policies.

Traditional network security architectures often depend on perimeter-based protection, in which users and devices inside a designated network are considered more trustworthy than those outside it. Although firewalls, virtual private networks, and conventional access-control mechanisms remain useful, they may not provide sufficient protection when resources are distributed across multiple cloud providers and edge locations. An attacker who obtains valid credentials or compromises an authorized device may exploit excessive permissions to access sensitive resources. Furthermore, once a user or device has been authenticated, some conventional systems do not continuously reassess its security status or the risks associated with subsequent requests.

To address these limitations, the Zero-Trust security model has emerged as an important approach to modern cybersecurity. Zero Trust is based on the principle of never trust, always verify, meaning that access should not be granted solely because a user, device, or application is connected to an internal network. Instead, access decisions should be based on verified identity, authorization policies, device security posture, contextual information, and the sensitivity of the requested resource. The National Institute of Standards and Technology (NIST), through Special Publication 800-207, provides a formal architectural foundation for implementing Zero Trust in enterprise environments (Rose et al., 2020).

An effective Zero-Trust architecture requires more than an initial login or identity check. It should support continuous monitoring, least-privilege access, secure communication, and timely reassessment of access privileges when relevant security conditions change. For example, if an authorized employee attempts to access a sensitive cloud database from an unfamiliar device or exhibits suspicious access behaviour, the system should be able to require additional verification, restrict the requested operation, or deny access. Similarly, an edge device that becomes compromised should not automatically retain unrestricted access to other devices, applications, or cloud services.

However, implementing Zero Trust across multi-cloud and edge computing environments presents several challenges. Different cloud providers may use different identity-management systems, access-control mechanisms, logging formats, and security configurations. Edge devices may also have limited processing power, memory, and network connectivity, making it difficult to apply complex security procedures consistently. In addition, the continuous collection and analysis of security information can introduce computational overhead and communication delays, particularly in latency-sensitive edge applications. Consequently, there is a need for a flexible security framework that can coordinate access decisions across distributed environments while considering both security requirements and operational performance.

This research therefore focuses on an enhanced dynamic Zero-Trust Cloud Architecture Framework for Multi-Cloud and Edge Computing. The proposed framework is intended to integrate identity verification, contextual risk assessment, dynamic access control, policy enforcement, and continuous security monitoring. It will evaluate relevant information associated with users, devices, applications, and access requests to determine whether access should be permitted, restricted, subjected to additional verification, or denied. The framework will also incorporate a coordinated policy-management mechanism to improve consistency across multiple cloud platforms and edge computing nodes.

The proposed framework will be developed and evaluated using a prototype implementation, with Python and Flask serving as the primary application technologies. A simulated multi-cloud and edge environment can be used to model legitimate access requests, unauthorized access attempts, compromised devices, changing risk conditions, and communication failures. The system will be evaluated using measurable indicators such as unauthorized-access blocking rate, legitimate-access acceptance rate, false acceptance rate, authentication latency, policy decision latency, and resource overhead. These measurements will provide an evidence-based assessment of the framework's security effectiveness and operational suitability.

Ultimately, the research seeks to contribute a structured and adaptable approach to securing distributed computing environments. By combining dynamic risk-based access control with Zero-Trust principles, the framework aims to reduce reliance on implicit trust, improve policy enforcement, and support more secure communication between cloud services and edge devices. Its practical value will be determined through implementation, controlled testing, and comparison with a clearly defined baseline security model.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

LITERATURE REVIEW

The reviewed literature reveals several complementary approaches to Zero-Trust security. The following table summarizes their contributions and relevance to the proposed research.

Authors	Approach	Main contribution or reported evidence	Relevance to the proposed study
Syed et al. (2022)	Comprehensive survey of Zero Trust	Examined authentication, access control, encryption, micro-segmentation, and implementation challenges.	Provides theoretical and technical foundations.
NIST (2025)	Practical Zero-Trust implementation guide	Documented 19 example implementations developed with 24 collaborating organizations.	Supports architecture design and implementation planning.
Tan et al. (2026)	Trusted-label cloud–edge–endpoint access control	Reported low resource overhead and attack-blocking results in its experimental environment.	Supports distributed, lightweight enforcement.
2026 dynamic access-control study	Behaviour modelling, trust scoring, and compliance auditing	Reported access-decision accuracy of 92.8% and a response time of 146.3 ms at 1,000 concurrent connections.	Supports risk-based authorization and measurable evaluation.

MATERIALS AND METHODS

The proposed system introduces a dynamic Zero-Trust security mechanism that evaluates each access request according to identity, device condition, contextual information, behavioural indicators, and resource sensitivity. The system follows a continuous verification principle. An initial successful login does not automatically authorize every subsequent operation. Access permissions are limited to the requested resource and can be reassessed when relevant conditions change. The main operations are: Verify the identity and authentication status of the requester, Check device security information and credential validity, Collect contextual attributes associated with the request, Calculate a normalized risk score, Evaluate the request against mandatory security policies, Permit, deny, restrict, or request additional verification, Enforce the decision at the relevant cloud or edge gateway, Record the decision and supporting information in an audit log and Reassess access when new risk information becomes available.

System architecture

The system architecture of the Enhanced Dynamic Zero-Trust Cloud Architecture Framework for Multi-Cloud and Edge Computing illustrates the structural organization and interaction of the components responsible for securing users, applications, cloud resources, and edge devices. The architecture is designed around the Zero-Trust principle of never trust, always verify, whereby every access request must satisfy the applicable identity, security, and authorization requirements before access is granted. The proposed architecture consists of six major components: the user and device layer, identity verification layer, context and risk assessment layer,

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

policy decision layer, policy enforcement layer, and resources layer. A monitoring and audit layer operates alongside these components to provide visibility, detect suspicious activities, and support continuous security reassessment.

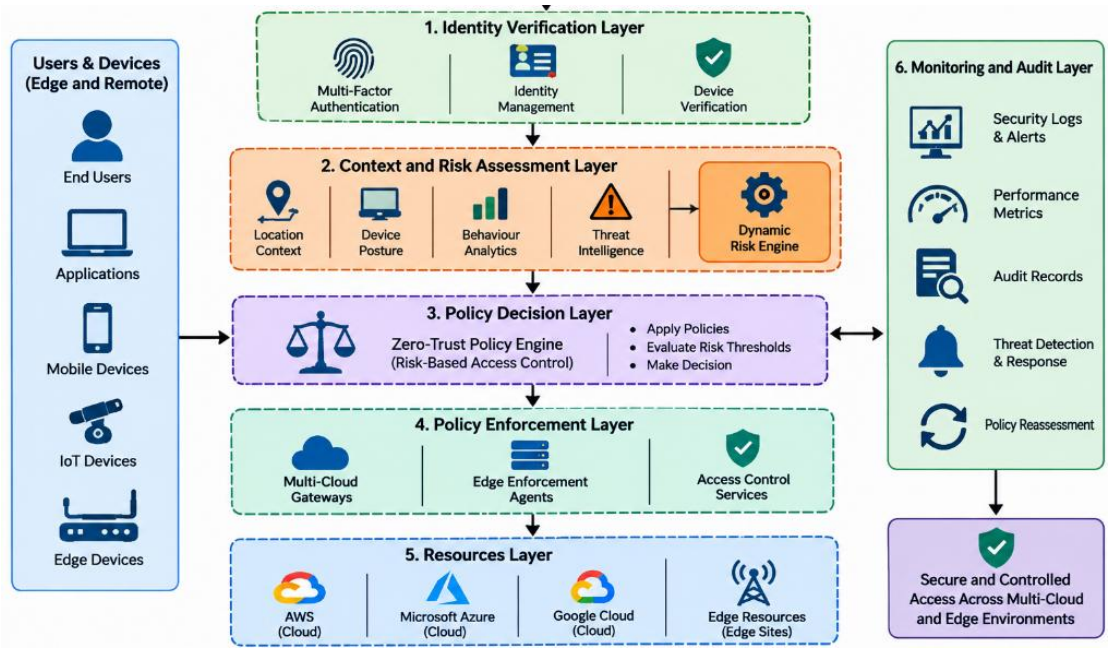

Figure 1: the system architecture of the proposed system.

User and Device Layer: The user and device layer represents the entry point into the system. It includes end users, applications, mobile devices, Internet of Things (IoT) devices, and edge devices that request access to cloud-based or edge-hosted resources. When a user or device attempts to access a resource, the system receives the request and forwards it for identity verification and security assessment. The system does not automatically trust a request based on its network location or previous access history. This layer helps establish the origin of each access request and provides the identity and device information needed by the subsequent security components.

Identity Verification Layer: The identity verification layer confirms the identity of users, devices, applications, and workloads before access is authorized. It may incorporate multi-factor authentication, identity management, credential validation, and device verification. Multi-factor authentication provides additional protection by requiring more than one form of evidence when appropriate. Device verification determines whether a requesting device is registered and whether its security status satisfies the applicable access policy. If the identity is invalid, the credentials have been revoked, or a mandatory verification requirement fails, the system denies the request. Successful identity verification does not automatically grant unrestricted access; the request must proceed through the remaining security checks.

Context and Risk Assessment Layer: The context and risk assessment layer evaluates the security circumstances surrounding each access request. It collects contextual information and sends relevant attributes to the dynamic risk engine. The major information sources include: Location context: Identifies unusual or policy-prohibited access locations where reliable information is available. Device posture: Assesses whether the device satisfies defined security requirements. Behaviour analytics: Examines access patterns and identifies deviations from established or expected behaviour. Threat intelligence: Incorporates relevant indicators of known threats. Resource sensitivity: Determines the level of protection required for the requested resource or operation. The dynamic risk engine combines these factors to calculate a risk score. Higher scores indicate greater assessed risk under the configured model. For example, a valid user requesting a sensitive resource from a device that fails security checks may receive a higher risk score than a user accessing an ordinary resource from a compliant device. The risk score supports the decision process but does not override mandatory security requirements.

Policy Decision Layer: The policy decision layer determines whether an access request should be permitted, restricted, subjected to additional verification, or denied. It combines the calculated risk score with access policies, user permissions, device status, and resource sensitivity. The decision process can be represented as:

$$D=f(R,I,P,S)$$

Where: D represents the access decision.

R represents the calculated risk score.

I represents identity and authentication status.

P represents the applicable access policy.

S represents resource sensitivity.

The system evaluates these inputs against predefined rules. Low-risk requests may be permitted when all mandatory checks pass, medium-risk requests may require additional verification or restricted permissions, and high-risk requests may be denied. This layer is important because it makes authorization responsive to changing security conditions rather than relying exclusively on static permissions.

Policy Enforcement Layer: The policy enforcement layer applies the decisions generated by the policy decision component. It includes multi-cloud gateways, edge enforcement agents, and access-control services positioned to protect the relevant resources. When access is permitted, the enforcement component allows only the authorized operations. When access is denied, the component blocks the request. Where additional verification or restrictions are required, it enforces those conditions before allowing the relevant operation. For example, a user may be allowed to view a cloud resource but prevented from modifying or downloading its contents. Similarly, an edge device may be permitted to communicate with a designated service while being denied access to unrelated devices. This separation between decision-making and enforcement supports consistent authorization across different cloud platforms and edge locations.

Resources Layer: The resources layer contains the services and infrastructure protected by the framework. It includes resources hosted on Amazon Web Services (AWS), Microsoft Azure, Google Cloud, and edge computing sites. Each resource can be assigned an appropriate sensitivity classification and access policy.

The framework evaluates requests according to the resource being accessed and the privileges required for the requested operation. The multi-cloud design aims to provide a coordinated security approach without requiring all protected resources to reside on a single cloud platform. Edge resources can also enforce locally applicable security rules, subject to the framework's defined policy synchronization and connectivity procedures.

Monitoring and Audit Layer: The monitoring and audit layer observes system activity and records information about access requests, risk assessments, authorization decisions, and security events. Its major functions include: Security logging: Recording access attempts, decisions, timestamps, and relevant events. Performance monitoring: Measuring policy-decision latency, resource utilization, and other operational indicators.

Threat detection: Identifying suspicious access patterns and potential security violations. Policy reassessment: Triggering a new evaluation when device status, user behaviour, or other relevant risk conditions change. Audit reporting: Providing records that support security reviews and incident investigations

The activity diagram

The activity diagram of the Enhanced Dynamic Zero-Trust Cloud Architecture Framework for Multi-Cloud and Edge Computing illustrates the sequence of activities performed when a user, application, or device requests access to a cloud or edge resource. It shows how the system verifies identity, collects contextual

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

information, calculates a risk score, makes an access decision, enforces security policies, and continuously monitors activities to detect changes in security conditions.

The primary purpose of the activity diagram is to demonstrate how the proposed framework processes access requests dynamically while ensuring that no user or device is trusted automatically

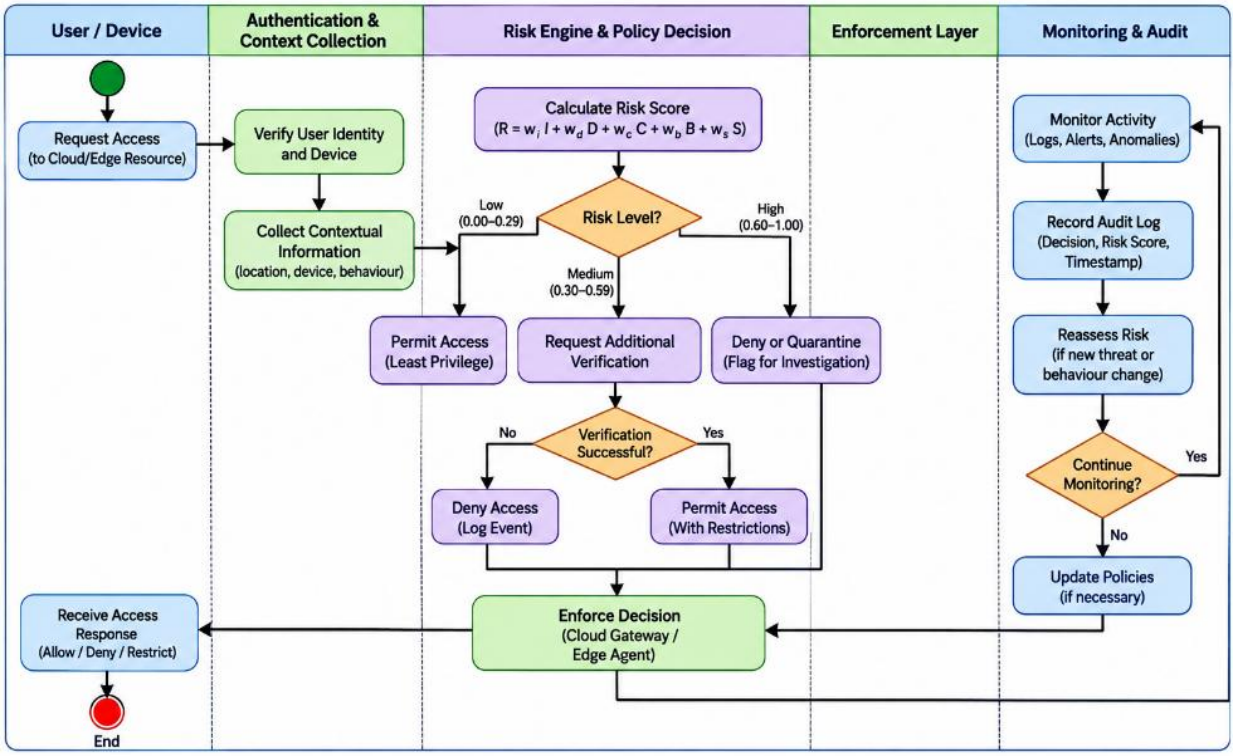

Figure 2: the activity diagram of the proposed system

The activity diagram demonstrates that the proposed framework uses a structured sequence of identity verification, contextual risk assessment, policy evaluation, access enforcement, and continuous monitoring.

Its dynamic decision process is intended to reduce unauthorized access while allowing legitimate users to access approved cloud and edge resources. The diagram also illustrates how additional verification, audit logging, and policy reassessment contribute to the security of distributed computing environments.

Use Case Diagram of the Proposed System

The use case diagram illustrates the interactions between the system's external actors and the functions provided by the Enhanced Dynamic Zero-Trust Cloud Architecture Framework for Multi-Cloud and Edge Computing.

The main actors are the End User, Administrator, Cloud Providers, and Edge Devices. The diagram shows how users authenticate and request resources, how the system verifies identity and device posture, calculates risk scores, makes access decisions, and enforces permissions. It also shows the administrator's responsibilities for managing users, configuring security policies, monitoring activities, and reviewing audit logs.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

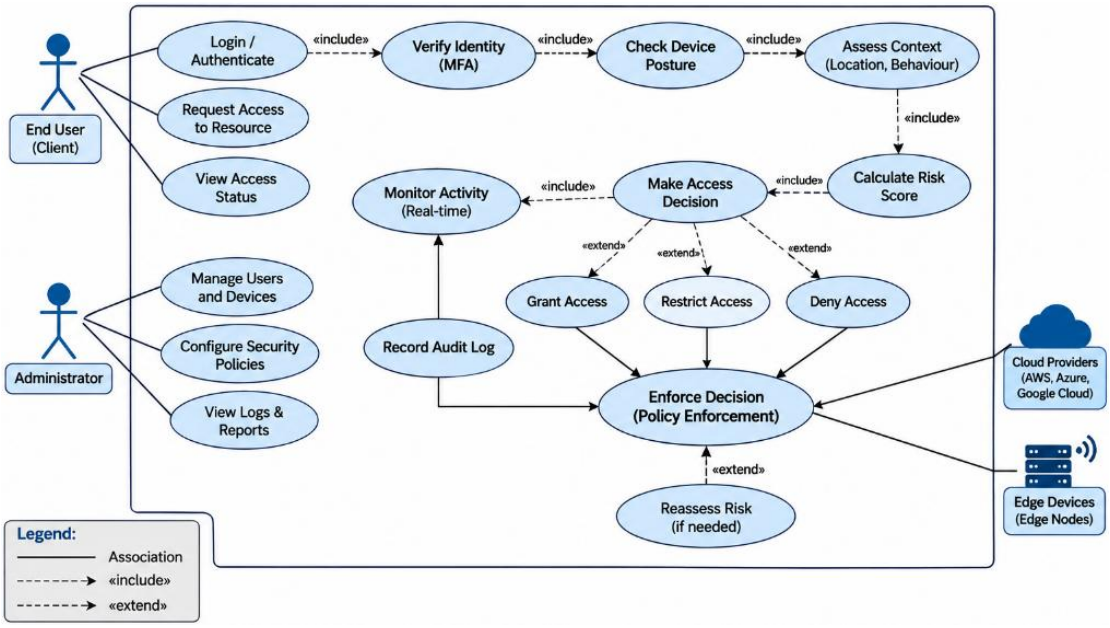

Figure 3: the Use Case diagram of the proposed system.

The overall process begins when a user logs into the application and submits a request to access a cloud or edge resource. The framework evaluates the request using the configured risk-assessment mechanism and mandatory security checks. The policy decision component then determines the appropriate response. Depending on the result, the request may be permitted, challenged, restricted, or denied. The system records the relevant event in the audit log, while the administrator can review the records and update security policies when necessary.

The sequence diagram of the proposed system

The sequence operation and interaction proceeds as follows:

- The user submits login credentials.
- The authentication service returns the authentication result.
- The user submits a request to access a resource.
- Relevant request and context information is sent for risk evaluation.
- The risk assessment engine calculates the risk score.
- The policy decision engine evaluates the request against security policies.
- An access decision is returned.
- The enforcement gateway applies the decision.
- The cloud or edge resource returns the appropriate response when the request is allowed to proceed.
- The user receives an access-granted or access-denied response.
- The system records the access event.
- The audit log database confirms that the event has been stored.

It demonstrates how the proposed Zero Trust framework processes an access request from authentication through risk evaluation, policy enforcement, and audit logging. It emphasizes that access is determined by the request's security conditions rather than being automatically trusted.

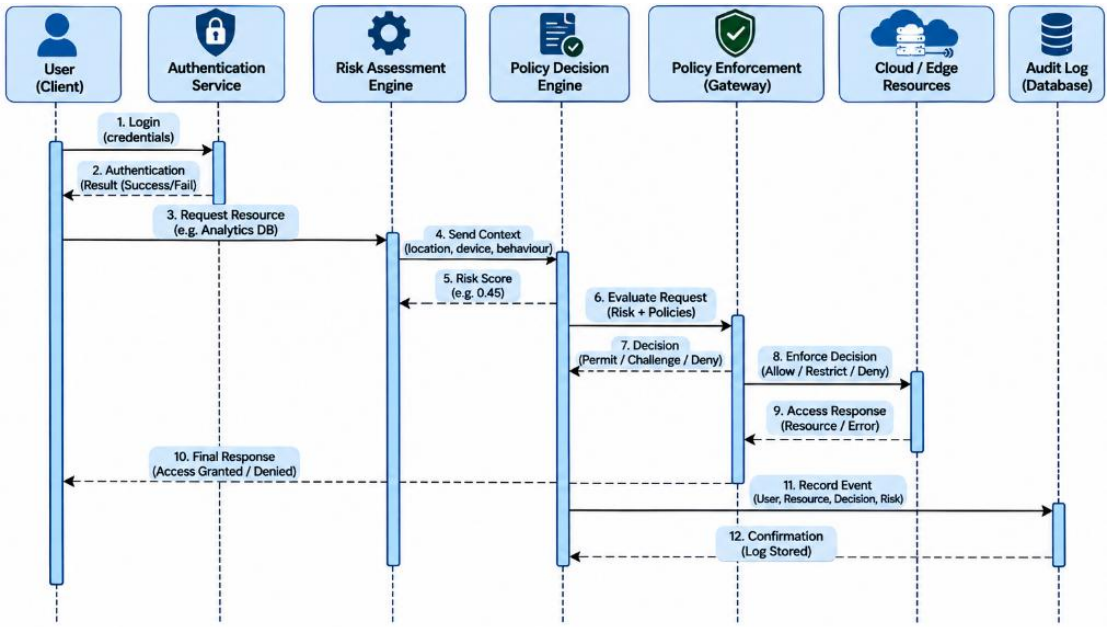

figure 4: the sequence diagram of the proposed system

The sequence diagram illustrates the chronological exchange of messages between the user, authentication service, risk assessment engine, policy decision engine, policy enforcement gateway, cloud/edge resources, and audit log database.

EXPERIMENT AND RESULT

The proposed framework uses a risk-scoring mechanism to evaluate access requests based on five factors: user identity, device compliance, access context, user behaviour, and resource sensitivity. The risk score is calculated using the following equation:

$$R = 0.20I + 0.25D + 0.15C + 0.25B + 0.15S$$

Where: I represents identity-related risk.

D represents device-related risk.

C represents contextual risk.

B represents behavioural risk.

S represents resource-sensitivity risk. Each factor is normalized to a value between 0 and 1, with higher values indicating greater risk. The weights sum to 1, so the resulting score remains within the range of 0 to 1 when all inputs are normalized.

Table 1: decision thresholds

Risk score	Access decision	Interpretation
Below 0.30	Permit	Low-risk request, provided mandatory checks pass

0.30 to below 0.60	Challenge or restrict	Additional verification or restricted access is required
0.60 and above	Deny	High-risk request is rejected

These thresholds enable the framework to respond differently to requests according to their assessed risk.

The access-control component combines the calculated risk score with mandatory security checks. For example, an inactive user or a device that fails the required compliance check is denied access regardless of the calculated risk score. Low-risk requests may be permitted, while medium-risk requests may require additional verification. medium-risk request can result in restricted access after the simulated additional-verification step succeeds. High-risk requests are denied. This approach supports the principle of least privilege because access decisions are based on the current request and the security requirements of the resource. Nevertheless, a successful verification step should not automatically guarantee unrestricted access; the policy must also define which operations and resources remain available.

The prototype represents resources from different cloud providers and edge environments. Each resource has a sensitivity value, which contributes to the access-risk calculation. The expected behaviour is that requests involving sensitive resources receive stricter scrutiny than requests involving less sensitive resources when other conditions are comparable. This is important because a compromised account accessing an ordinary monitoring service may present a different level of risk from the same account attempting to access a sensitive identity service or customer database. The prototype demonstrates the decision logic for different resource types.

The audit-logging component records access-related events, including the requesting user, target resource, assessed risk, resulting decision, and relevant timestamps. Audit records can help administrators investigate suspicious access attempts, identify repeated denials, and review how access policies are being applied. They also provide a basis for analysing the consistency of decisions across different resources. The usefulness of the audit mechanism depends on the completeness and integrity of the records. In a production deployment, audit logs should have appropriate access controls, retention policies, secure storage, and protection against unauthorized modification or deletion.

To determine whether the proposed framework performs effectively, the implementation should be tested using a documented set of access requests. The following metrics are recommended.

Access-decision accuracy: Measures the proportion of access decisions that agree with independently established expected decisions.

$$Accuracy = \frac{Correct\ Decisions}{Total\ Decisions} \times 100$$

Expected decisions must be defined before testing; otherwise, the metric may simply reproduce the framework's own policy rules rather than measure security effectiveness.

False acceptance rate: Measures the proportion of unauthorized or malicious requests incorrectly granted access. $$FAR = \frac{Unauthorized\ Requests\ Granted}{All\ Unauthorized\ Requests} \times 100$$

False rejection rate: Measures the proportion of legitimate requests incorrectly denied.

$$FRR = \frac{Legitimate\ Requests\ Denied}{All\ Legitimate\ Requests} \times 100$$

Average decision latency: Measures the average time required to evaluate and return an access decision. $$T_{avg} = \frac{\sum_{i=1}^{n} T_i}{n}$$

Where Ti is the measured decision time for request i, and n is the number of tested requests. The experiment should also record CPU and memory usage if system overhead is an evaluation objective.

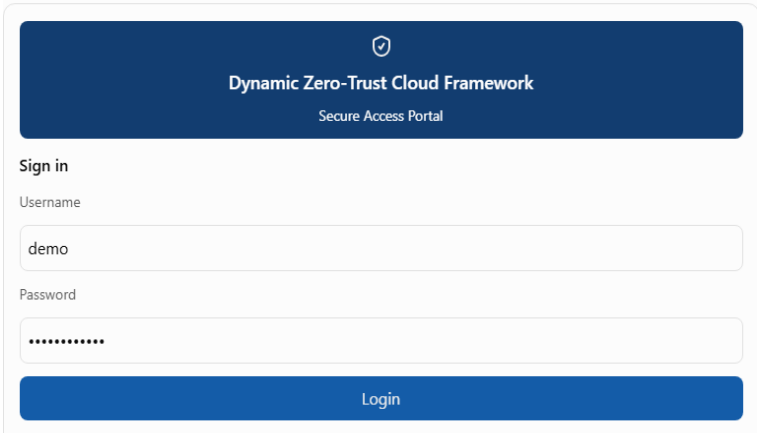

Figure 5: Login Page Output

The login page allows registered users and administrators to authenticate before accessing the Zero-Trust dashboard. The system validates the submitted username and password. If the credentials are correct and the account is active, the user is redirected to the dashboard. Invalid credentials result in an error message.

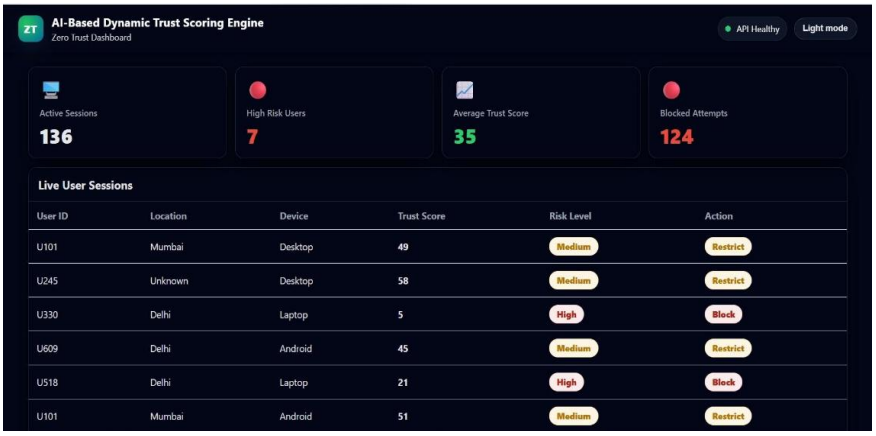

Figure 6: Security Dashboard Output

The dashboard provides an overview of the system, including available resources, access requests, security decisions, and monitoring activities. The dashboard provides an overview of simulated cloud and edge resources and summarizes recorded access decisions. The dashboard displays the number of recorded decisions, denied requests, protected resources, and recent access activity.



Figure 7: Access Request Output

The access-request interface allows a user to select a cloud or edge resource and submit security attributes for evaluation. The system calculates the risk score using the configured weights. It then applies mandatory checks and the risk thresholds to determine the decision.

Test scenario	Risk score	Expected decision	Reason
Compliant device, low risk	0.150	PERMIT	Mandatory checks pass
Compliant device, medium risk	0.450	CHALLENGE	Additional verification required
Compliant device, high risk	0.750	DENY	Risk exceeds threshold
Non-compliant device	0.150	DENY	Mandatory device check fails

Figure 8: Risk Assessment and Access Decision Output.

These are illustrative test cases. The scores assume the specified inputs and weights; they are not measured success rates or performance results.

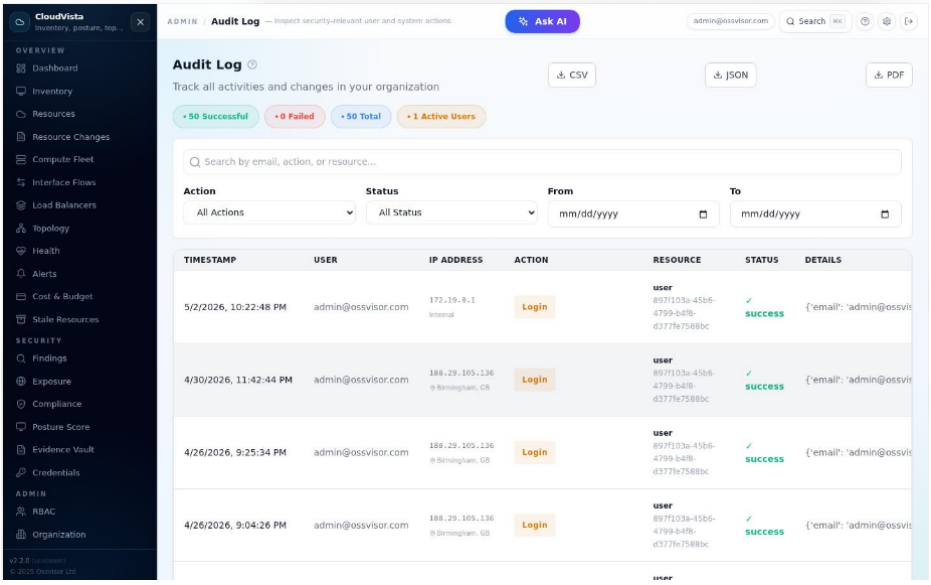

Figure 9: Audit Log Output

The audit log records the activities performed within the system.

Typical information includes: Username., Requested resource., Risk score. Access decision and Date and time of request. The administrator can review these records to identify suspicious access attempts and monitor the security activities of the application. The audit log records the requester, resource, environment, calculated risk score, decision, and reason.

RESULT DISCUSSION

The system is designed to demonstrate how dynamic risk assessment can influence access decisions across simulated cloud and edge resources. The risk engine evaluates contextual factors, while the policy decision module applies mandatory checks and configured thresholds.

The dashboard presents the available resources and recent access decisions. The audit page provides an administrative view of recorded events, while the access request form allows controlled testing of different risk conditions. The implementation demonstrates the core workflow of the proposed architecture. However, the current version is a prototype: cloud providers are simulated, device posture is supplied through test inputs,

and continuous monitoring is not connected to live endpoint telemetry. Its security effectiveness and performance must therefore be established through further testing

CONCLUSION

This paper focused on designing and implementing a security framework that improves access control across cloud and edge computing environments. The study was motivated by the limitations of traditional security approaches that often rely on static access policies and implicit trust, which may expose distributed computing environments to unauthorized access, compromised devices, and other security threats. The proposed framework adopts the principles of Zero Trust Architecture by requiring access requests to undergo security evaluation before access is granted. It incorporates identity verification, device compliance, contextual information, behavioural risk, and resource sensitivity into a dynamic risk-assessment mechanism. Based on the calculated risk score and mandatory security checks, the system determines whether a request should be permitted, challenged, restricted, or denied. The paper also implemented a prototype using Python Flask and an SQLite database. The prototype provides user authentication, resource selection, risk-based access decisions, and audit logging. These features demonstrate how dynamic access-control policies can be organized within a web-based application and provide a foundation for further experimentation.

Furthermore, the framework supports the logical representation of resources across multiple cloud and edge environments. By considering resource sensitivity during access evaluation, it provides a basis for applying stricter security controls to critical resources while maintaining appropriate access for lower-risk requests. Its audit-logging functionality also supports accountability by recording access-related events for subsequent review. Despite these contributions, the current implementation remains a prototype. It does not establish full integration with live cloud platforms, production-grade multi-factor authentication, or real-time edge-device monitoring. In addition, the effectiveness of the risk thresholds and the performance of the framework must be confirmed through controlled experiments using representative access-request datasets. Consequently, the project demonstrates the design and implementation of the proposed approach but does not yet establish measured superiority over existing security systems.

In conclusion, the project provides a foundation for improving access management in multi-cloud and edge computing through dynamic risk assessment, policy-based authorization, and security auditing. With further testing, integration, and optimization, the framework could contribute to a more adaptive and accountable security architecture for distributed computing environments.

REFERENCES

1. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
2. National Institute of Standards and Technology. (2023). Implementing a zero trust architecture (NIST Special Publication 1800-35). NIST.
3. National Institute of Standards and Technology. (2025). Implementing a zero trust architecture (NIST Special Publication 1800-35). https://doi.org/10.6028/NIST.SP.1800-35
4. Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z. A., & Doss, R. (2022). Zero Trust Architecture (ZTA): A comprehensive survey. IEEE Access, 10, 57143–57179. https://doi.org/10.1109/ACCESS.2022.3174679
5. Tan, R., Zhang, Y., Liu, W., Liu, B., Cao, Y., Liu, X., et al. (2026). TLBAC: A zero-trust based cloud-edge-endpoint access control system using trusted labels. Cybersecurity, 9, Article 180. https://link.springer.com/article/10.1186/s42400-026-00616-0
6. Dynamic access control and privacy compliance strategies in a zero-trust data security architecture. (2026). In Proceedings of the 7th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT). https://doi.org/10.1109/AINIT70033.2026.11557827