IJEMD-CSAI, Special Issue (2026) https://doi.org/10.54938/ijemdcsai.2026.04.2.666

IJEMD

International Journal of Emerging Multidisciplinaries: Computer Science and Artificial Intelligence

Research Paper

Journal Homepage: www.ojs.ijemd.com
ISSN (print): 2791-0164 ISSN (online): 2957-5036

Towards Behaviour-Aware Adaptive Zero Trust Architecture

Fatiha Azzwa Binti Ab Hadi [1], Muhammad Musyrif Mifzal Bin Jefry [1], Kazi Samir Miah [1], Ahmad I. M. Allahham [1], Brendon Chong Zi Yeung [1], Adwin Chee Hansen [1], Vasha Jahangir [1], Siva Raja Sindiramutty [1]

1. School of Computer Science, Taylor's University, Subang Jaya, 47500, Selangor Malaysia

1.0 Abstract

The proposed study will present an Adaptive Behaviour-Centric Zero Trust Architecture (AB-ZTA), which targets critical weaknesses of current Zero Trust systems. Although the existing industry designs offer robust identity and access-level controls, they are still prone to heavy reliance on fixed policies, environment-specific ecosystems, as well as narrow behavioural context. These silos decrease visibility in mixed environments, responsiveness to new threats, and can generally lead to user-experience friction by forcing unnecessary repeated authentication. AB-ZTA model has a dynamic and intelligence-based approach that is constantly validating users, devices, and session based on real-time behavioural analytics and adaptive trust scoring. Risk-adaptive access control of a highly granular nature is facilitated by core ingredients such as distributed Policy Decision Point, dynamic Policy Enforcement Point, machine-learning-based anomaly detection, and context-driven data shielding. Cryptographic standards (TLS 1.3, AES-256, FIDO2, JWT/JWE) are also implemented in the architecture to ensure the privacy of communication, sessions, and data at rest. The comparative analysis demonstrates that AB-ZTA improves threat detection, minimizes the lateral mobility with the help of micro-segmentation, and minimizes data disclosure with device-bound session tokens and dynamic masking. Even though there is some performance overhead on the part of continuous behavioural verification, these impacts are manageable, and they are compensated by the level of security enhancements. Altogether, AB-ZTA is a scalable and vendor-neutral and user-conscious Zero Trust architecture that can be applied to contemporary enterprise networks.

Keywords: Zero Trust Architecture (ZTA), Behaviour-Aware Security, Adaptive Trust Scoring, Continuous Authentication and Authorization, Enterprise Network Security

Email Addresses : 0370969@sd.taylors.edu.my, 0370999@sd.taylors.edu.my, 0370235@sd.taylors.edu.my, 0371531@sd.taylors.edu.my, 0371789@sd.taylors.edu.my, 0362624@sd.taylors.edu.my, 0371173@sd.taylors.edu.my, siva.sindiramutty@taylors.edu.my

2.0 Introduction & Problem Identification

2.1 Overview of Enterprise Network Security

Enterprise network is an infrastructure that comprises software and hardware appliances to connect an organization's computers, servers, and other devices [3]; [17]. Differs from a general internet network, enterprise runs a private network where it is not available for everyone to connect. Instead, it could only be accessed to those authorized in the specific organization. It integrates IT systems used by a business and provides a specific channel to perform related actions and technical tasks. It uses both Local Area Network (LAN) to connect the system within close proximity and Wide Area Network (WAN) that could span across different geographic locations. For a large business, it is ideal to implement this network to support easier communication and data exchange within the organization itself. Nowadays, enterprise networks have been widely adopted especially for large businesses which calls for a continuous evolution to meet growing demands [8]; [27].

While the enterprise network is rapidly expanding, cybersecurity threats are also emerging and pose significant challenges within the domain. Since COVID-19, employees have adopted remote work where they can work from anywhere without having to come to the office premises. This has prompted most organizations to migrate their workload to cloud services where it is easier to access [2]; [26]. Additionally, a new IT policy called 'Bring Your Own Device' has also been introduced where the employees are allowed to use their personal devices. Such migration shifts a higher risk of account compromise and exposure of sensitive data. Next, the enterprise network is also vulnerable to insider threats where any employees using the network might potentially misuse their access to the system. This would result in data theft and causing reputational or financial damage to the organization. Moreover, when most enterprises are expanding their business, the increasing network complexities have become a growing concern [16]; [7]; [5]. With more devices and technologies, the network is most likely to have inconsistent security policies which could be exploited by attackers.

As conventional security relies heavily on static credentials, it lacks continuous verification and behavioural analysis which makes it easier for attackers to move laterally once an account has been compromised [10]; [19]. Following this, the sensitive data is prone to get leaked since there's no protection for the data itself and it rather focuses on securing the perimeter only. This risk is not only limited to external attackers; indeed, an insider or employee might use this vulnerability to pose a threat to the organization itself. Furthermore, traditional security fails to adapt to the new dynamic cloud environment [1]; [22]. While it is designed for a defined perimeter with an on-premises network, it struggles in this new environment where workloads are constantly shifting and distributed across multiple platforms. In addition, the traditional approach uses manual configuration and siloed tools for firewalls, creating a gap and inconsistent security policies. This is by cause of increasing human errors during configuration and tools that often don't integrate well. On the whole, the traditional security measures are no longer reliable and inadequate to address modern cybersecurity threats.

2.2 Concept of Zero Trust Architecture (ZTA)

Zero Trust Architecture (ZTA) was introduced by John Kindervag and standardized by the National Institute of Standards and Technology (NIST) to strengthen cybersecurity especially in enterprise

networks. Its core principle, “never trust, always verify”, is to assume that all entities are not trusted by default whether they are from inside or outside the organization’s network and identity must be authenticated before granting an access [18]; [24]. Therefore, all users or entities must be continuously verified and authorized at every stage of the interaction before they could be granted access to the resources. Moreover, this architecture also follows a basic principle of “minimum privilege” which only gives users and devices a minimum access needed to do their specific task (Rose, Borchert, Mitchell, & Connelly, 2020; [21]. Using these basic principles, ZTA helps to give more protection and prevent breaches to networks which are very essential for today’s enterprise security. Nowadays, most organizations have opted to shift to this new paradigm while some of them are still in the hybrid stages rather than fully using Zero Trust.

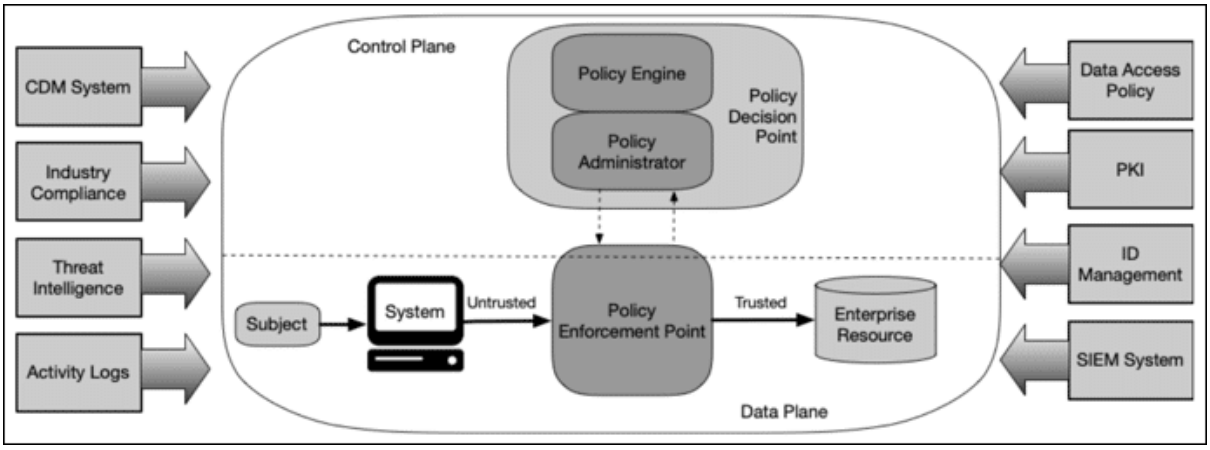


Figure 1

Figure 1 illustrates how ZTA works at a basic level, In ZTA, all access requests begin with the subject attempting to access protected resources from outside or inside the network where it is called an untrusted zone. The request would then go through an evaluation process where the Zero Trust principle is applied before granting further access to the implicit trust zone. ZTA is made up of two key components: Policy Decision Point (PDP) and Policy Enforcement Point (PEP). PDP acts as the brain of ZTA and is composed of Policy Engine (PE) and Policy Administrator (PA). First, PE would analyse the user’s metadata such as user identity, device compliance, and geolocation using the Trust Algorithm (TA). The algorithm also takes into account threat intel feeds where it provides external data such as malicious IP address, malware signatures, and others. Meanwhile, PA makes a decision either to grant or restrict access based on evaluation. This decision would then be executed in real time by PEP. In addition, PEP is positioned closest to the resources to create a small implicit trust zone as it is generally considered a good practice in Zero Trust design.

Against today’s cloud environments and evolving cyber threats, ZTA has become ideal in most cybersecurity especially for enterprise networks. As most organizations have already shifted to cloud service and remote work models, the traditional security is no longer reliable. ZTA defeats perimeter-based defences by enforcing continuous verification, least privilege access, and contextual authentication for every user and device. This can minimize the risk of insider threats, credential theft and lateral movement by attackers once they’re inside the network. Furthermore, ZTA supports faster anomalies detection and responds to potential breaches because it is easier to control across hybrid

infrastructures. All in all, adopting ZTA has become crucial for an organization as it builds a proactive defence and protects data more effectively.

2.3 Real World Attack Example

The importance of adopting ZTA could be further demonstrated by looking back at one of the most significant cyberattacks in recent years, the SolarWinds Supply Chain Attack. First of all, SolarWinds is an American company that builds software to help businesses to manage their networks and IT infrastructure. On 13 December 2020, it was reported that multiple government and high profile agencies who were using their services were breached. The attack was done through a regular update which has been tainted with malicious code [4]; [25]. It was later found out that the hackers were already accessing and intruding SolarWinds' system around September 2019. They created a backdoor which enabled them to access the customer's system and install further malware. Though, experts said that the preparation for the attack went way back longer [20]; [15].

The SolarWinds attack was successful as the attackers exploited implicit trust within the software supply chain. Seeing as the attack originated from a trusted and whitelisted application, the attackers gained privileged access without triggering the traditional perimeter defences. This highlights the importance of ZTA where it eliminates the assumption of inherent trust. Thus, every entity must be verified continuously even if they came from inside the network. Next, the attack was due to lack of network segmentation. As the internal traffic was not consistently monitored, the threat actors were able to move laterally across systems and access any sensitive data. In contrast, ZTA would limit each component's access and the compromised instances would have been isolated to minimal network zones.

3.0 Literature Review & Gap Analysis

3.1 Overview of Prior Work

This section presents an overview of major Zero Trust frameworks developed by several leading vendors. Each framework is briefly described in terms of its origin, design principles, and key components. Additional reference documents for further reading are provided in Appendix A.

- **Google BeyondCorp**

Google BeyondCorp has become the blueprint for modern Zero Trust models. It was developed by Google Inc. and initiated by Google's Enterprise Security Team as a response to Operation Aurora in 2009 [6]. Its goal is to shift access controls from the network perimeter to individual devices and users by the underlying concept of "Access without a VPN" where users can securely reach corporate resources from any device, anywhere, if both the device and user are verified and meet the policy. The structure centres on device-based and user-based trust. The key components include Device Inventory Service, Access Control Engine, Gateways, Trust Inference System, and User/Device Certificates. While it was initially applied internally at Google's enterprise network for all employees and systems, now, it is widely adopted as the foundation for ZTA in industry.

- **Microsoft Zero Trust**

Microsoft Zero Trust is a security strategy based on Microsoft's experience "helping customers secure their organizations, and by implementing our own Zero Trust model for ourselves". Microsoft positions it as an end-to-end security strategy that spans identity, devices, applications, infrastructure, and data. The goal is to protect the digital estate wherever they are located and to reduce risk of breach or minimize blast radius, if breach happens. The model focuses on verifying identity (R et al., 2021), device health/posture, application/workload context, data classification, location, and risk signals for every access request. Its structure applies across six functional areas including identity, devices, application, infrastructure, data, and network. It is mostly applied in environments using Microsoft's cloud services but also the strategy applies to on-premises/hybrid systems and non-Microsoft platforms. It is intended for all sizes of organizations, though adoption requires phased maturity and is often guided by C-suite buy-in and integration across business and IT functions.

- **Cisco Zero Trust Architecture**

Cisco Zero Trust was developed by Cisco and builds on their own experience and solutions that aligns with the broader industry frameworks. The structure is organised around three core pillars which are User & Device Security, Network & Cloud Security, and Application & Data Security. It also uses a "Reference Architecture" and "Reference Design" that map capabilities to those pillars, show business flows, and list supporting technologies. Exclusively in Cisco implementation, the framework is defined using SAFE methodology which is a security-centric methodology and model for an effective Security Architecture. Some of the common capabilities in Cisco Zero Trust include anomaly detection, device posture management, flow analytics, and Security Orchestration Automation and Response (SOAR). This model has already been applied in modern networks such as enterprises with hybrid, multi-cloud, distributed infrastructure, etc.

- **VMWare Zero Trust**

VMware Zero Trust emphasizes that every component should be subject to verification and policy enforcement. The design principles include constant testing of behaviour, micro-segmentation of workloads, and separation of control plane from workload plane so that control infrastructure remains highly trusted [13]. This framework structures the Zero Trust into endpoint, data centre, control plane, user access, workload access, and network flows. It consists of several different components. For endpoints, it uses VMware Workspace ONE Unified Endpoint Management and Carbon Black Cloud sensor for threat telemetry and device posture. For data centres, it follows the SDN model. VMware NSX is used to easily segment the data centre while VMware vSphere is to provide the cloud infrastructure.

- **Palo Alto Zero Trust Edge**

Palo Alto Zero Trust Edge is designed to focus on protecting three major domains which are users, applications, and infrastructure. The structure is organised around a five-step methodology for implementation and applying Zero Trust across all validation points including identity, device/workload, access, and transaction. Its key principles follow closely with the other frameworks which is to never trust by default and always verify. The Zero Trust platform is integrated with Strata Network Security Platform and Cortex Security Operations Platform. Both platforms work well together as Strata enforces the policies while Cortex monitors, detects, and responds to the access (Pal et al., 2023). Meanwhile, the infrastructure consists of two types depending on the deployment method. Secure Access Service Edge (SASE) is used for cloud-delivered security and Next-Generation Firewalls (NGFWs) for on-premises or data centre enforcement.

- **Okta Identity-Driven ZTA**

Okta Zero Trust Access Management is to help transition organisations away from the tradition perimeter model and into a Zero Trust model. Okta focuses on making identity and access management (IAM) as the core element of Zero Trust. This framework helps incremental implementation by providing a maturity model with stages of IAM/Zero Trust adoption from stage 0 to stage 3. Stage 0 where most of the network started with fragmented identities. Then, everything is consolidated under one IAM system via Universal Directory, SSO, and Okta Access Gateway. Stage 2 means gathering user's, application, and device context and applying access policies. Risk engine for adaptive response is also part of the "Adaptive Workforce" in Stage 3. Okta encourages starting with IAM consolidation as they mostly emphasis that identity must lead the Zero Trust strategy.

3.2 Comparative Evaluation

Framework	Scope & Architectural Focus	Enforcement Type / Trust Decision Model	Integration Level	Strengths	Limitations	Operational Considerations
Google BeyondCorp	• enterprise access from any network without VPN • focuses on user and device authentication, context-aware access and continuous evaluation	• trust is never implicit • decisions based on device posture, user identity, and context • enforcement at access proxies and applications	• best integrated in cloud-first or hybrid environments • requires device management and identity systems	• proven at Google scale • seamless remote access • reduces lateral movement and VPN reliance	• difficult for legacy applications that assume trusted networks • high implementation effort for non-Google ecosystems	• requires monitoring and telemetry • policy management can be complex • careful UX design needed to avoid access friction
Microsoft Zero Trust	• end-to-end coverage across identity, devices, applications, infrastructure, and network • designed for hybrid work and multi-cloud integration	• conditional access policies • continuous evaluation of user, device, and resource context • principle of least privilege enforced dynamically	• high integration with Microsoft ecosystem • moderate integration with heterogeneous environments possible	• broad tooling and guidance • supports hybrid cloud • structured maturity model for phased adoption	• complexity increases with legacy or multi-vendor environments • dependence on Microsoft technologies for full benefits	• requires robust identity and device management • careful policy design to balance security and user productivity • monitoring and audit mechanisms essential
Cisco ZTA	• comprehensive enterprise framework covering network,	• trust decisions use continuous monitoring, device posture, and policy	• high for Cisco-centric infrastructure	• strong network visibility • mature policy enforcement	• may require network redesign	• continuous monitoring and analytics required segmentation must

	cloud, devices, users, applications, and data • emphasizes visibility and analytics	enforcement points across network	• less seamless for cloud-first or SaaS-only deployments	• scalable for enterprise networks	• less agile for SaaS / cloud-native scenarios	be carefully designed
VMware Zero Trust	• focus on infrastructure, data centre, and multi-cloud workloads • leverages virtualization, NSX, and micro-segmentation	• trust decisions combine identity, device posture, and workload context • enforcement via distributed firewalls and micro-segmentation policies	• high integration for VMware-based virtualized environments • limited effectiveness outside VMware ecosystem	• fine-grained workload isolation • strong internal segmentation • unifies security ad network policies within virtualized infrastructure	• less emphasis on identity/application layer • deployment complexity in large workloads	• micro-segmentation requires policy tuning • monitoring and compliance management are critical • performance optimization to avoid bottlenecks
Palo Alto ZTE	• edge-focused framework supporting distributed users, cloud / SaaS access, and remote sites • integrates networking and security at the edge	• trust decisions rely on continuous evaluation of user, device, and application • enforcement through inline proxies, ZTNA, and secure web gateways	• high for organizations with distributed or remote workforce • best in SD-WAN and SADE environments	• reduces VPN dependency • optimized for remote/cloud access • combines network and security at edge to reduce latency	• overkill for traditional centralized networks, investment in edge devices and cloud stack required	• continuous monitoring of edge traffic needed • configuration of edge devices must ensure performance and policy compliance

Okta ZTA	• identity-centric framework securing user access to cloud / SaaS applications • focuses on device context and adaptive access control	• continuous trust assessment based on identity, device posture, user behaviour, and risk signals • enforcement at identity gateways and application access	• high for SaaS-first organizations • moderated for hybrid or on-premises heavy deployments	• strong identity governance • rapid deployment • flexible integration with cloud applications • excellent UX for end-users	• limited coverage for network segmentation and workload-level controls • supplementary solutions needed for full zero-trust coverage	• requires robust identity lifecycle management • behaviour and device posture essential • dependency on continuous context collection

Figure 2

3.3 Research Gap

Despite multiple Zero Trust frameworks offer mature enterprise solutions, the current research still shows several unresolved limitations. First of all, each framework demonstrates strength within its own ecosystem, however, none of them provide a vendor natural model that can integrate seamlessly across any environments. This has made vendor dependency a major challenge and would create visibility gaps when users, devices, and workloads operate across mixed environments.

Secondly, it is obvious that the architectural focused is imbalance. Note well that identity-centric solutions such as Okta and BeyondCorp excel in user authentication and contextual access control but provide limited visibility into networks and workload-level interactions. Contrarily, Cisco and VMware might have better infrastructure control yet neglect adaptive identity-driven policies. This imbalance results fragmented security postures.

Moreover, most implementations rely on static conditional rules that fail to account for evolving contextual risk in time. The dynamic and risk-adaptive trust mechanisms are still underdeveloped. This problem not only reduces responsiveness to emerging threats, but also makes it inefficiency for operational especially when scaling across distributed users and applications.

Last but not least, the existing frameworks focus on tightening security but do not balance it effectively with user experience and performance optimization. To illustrate, over-enforcement such as repeatedly forcing MFA can caused fatigue and get user frustrated. In other word, they lack mechanisms that can intelligently adapt trust levels. Therefore, this research addresses all the gap by developing a new and enhanced framework suitable for enterprise networks.

4.0 Proposed Secure System

4.1 Overview of the Proposed ZTA Design

Adaptive Behaviour Centric Zero Trust Architecture (AB-ZTA) is suggested based on the traditional principles of the Zero Trust introduced by NIST SP 800-207 but are proposed to be a more dynamic and intelligent security framework. Unlike current Zero Trust systems which apply access controls by implementing static policies, AB-ZTA proposes adaptive and context-sensitive decision-making whereby the degree of access trust constantly changes according to the behavioural analytics, the environmental conditions, and real-time risk assessment.

The design is based on five fundamental principles that are continuous verification behaviour, adaptive trust score, dynamic enforcement of policy, distributed decision-making and context aware protection of data. These principles are interconnected to make sure that security enforcement becomes self-correcting and proactive as opposed to reactive or ineffective.

i. Constant Behavioural Checking

Traditional Zero Trust authenticates identity only at login, but AB-ZTA enhances this with continuous behavioural verification. It monitors user activity such as login patterns, device type, location, and interaction frequency. When behaviour deviates from the user's baseline, the system triggers additional verification such as re-authentication or step-up MFA. This ensures trust is evaluated throughout the session, not granted once at login.

ii. Adaptive Trust Scoring

AB-ZTA uses a dynamic trust-scoring mechanism that assigns real-time scores to users, devices, or sessions based on identity assurance, device health, environment, and behavioural consistency. Scores adjust automatically in response to anomalies or verified safe activity. Access privileges map to trust levels, limiting high-risk users. This adaptability strengthens protection against credential theft, insider threats, and compromised devices.

iii. Dynamic Policy Enforcement Point (PEP)

Unlike traditional fixed PEPs, AB-ZTA employs dynamic PEPs where enforcement conditions change according to trust score and behavioural analytics. This responsiveness creates fine-grained, real-time access control and reduces the attack surface. Policies include:

- Complete Access: Characteristic connection to high trusted sessions.
- Condition: Invokes other verifications or MFA.
- Limited Access: User has limited or no access to do anything.
- Denied Access: Blocks the access and sends an alert.

iv. Distributed Policy Decision Point (PDP)

To overcome centralized bottlenecks, AB-ZTA deploys distributed PDPs near endpoints and data sources. These nodes make local decisions using contextual factors like device metadata, location, and environmental signals without relying on a central server. This decentralization improves response

speed, scalability, and fault tolerance, and supports enforcement across hybrid or multi-cloud environments.

v. Context-Aware Data Shield

AB-ZTA incorporates dynamic, context-aware data protection based on sensitivity, device condition, and access context. For example, sensitive files accessed on unmanaged devices may trigger masking, read-only permissions, or enforced AES-256 encryption. By adjusting protection levels to contextual risk and trust score, the model significantly reduces data exposure and limits the impact of breaches.

4.2 Key Components

No	Component Name	Description and Function
i.	User Identify & Device Profiler	Collect user identify, device metadata, and environmental attributes (such as OS version, geolocation, IP address, and network type). It forms the foundation for identity verification and risk assessment. The profiler ensures that every access request originates from a verifies and compliant device, reducing risks of spoofing or unrecognized endpoints.
ii.	Behavioural Analysis Engine	Continuously monitors user behaviour such as login times, session duration, frequency of access, and transaction anomalies. It leverages machine learning algorithms to detect deviations from normal activity patterns. When anomalies are detected, the system triggers re-authentication or temporary access restrictions.
iii.	Adaptive Trust Scoring Module	Calculates a dynamic trust score for each user or device session based on contextual parameters including identity assurance, behavioural consistency, device posture, and network environment. The trust score fluctuates in real time, increasing when user behaviour aligns with normal baselines and decreasing during abnormal activity, influencing the level of access granted.
iv.	Policy Decision Point (PDP)	Serves as the core decision-making component that determines whether to allow, restrict, or deny access. Using Attribute-Based Access Control (ABAC) and XACML policies, the PDP analyses inputs from the Adaptive Trust Scoring Module and Behavioural Analysis Engine. It ensures all access decisions are made contextually and in compliance with enterprise security policies.

v.	Policy Enforcement Point (PEP)	Enforces the decisions made by the PDP. Depending on the trust score and contextual risk, the PEP may allow full access, limited access, or deny access entirely. It acts as the security gatekeeper positioned closest to protected resources to minimize the implicit trust zone.
vi.	Continuous Monitoring System	Operates as a feedback and auditing layer that constantly evaluates system logs, network traffic, and authentication events. It detects real-time anomalies and recalibrates trust scores dynamically. The system ensures adaptive response actions such as session termination, privilege downgrades, or re-authentication requests are executed automatically.
vii.	Context-Aware Data Shield	Protects data based on context, sensitivity, and location. It applies appropriate protection mechanisms such as AES-256 encryption, data masking, or access throttling depending on trust score, device posture, and network type. This ensures that data exposure is proportionate to the evaluated and context of each interaction.

Figure 3

4.3 Workflow and Data Flow Explanation

4.3.1 Load Verification

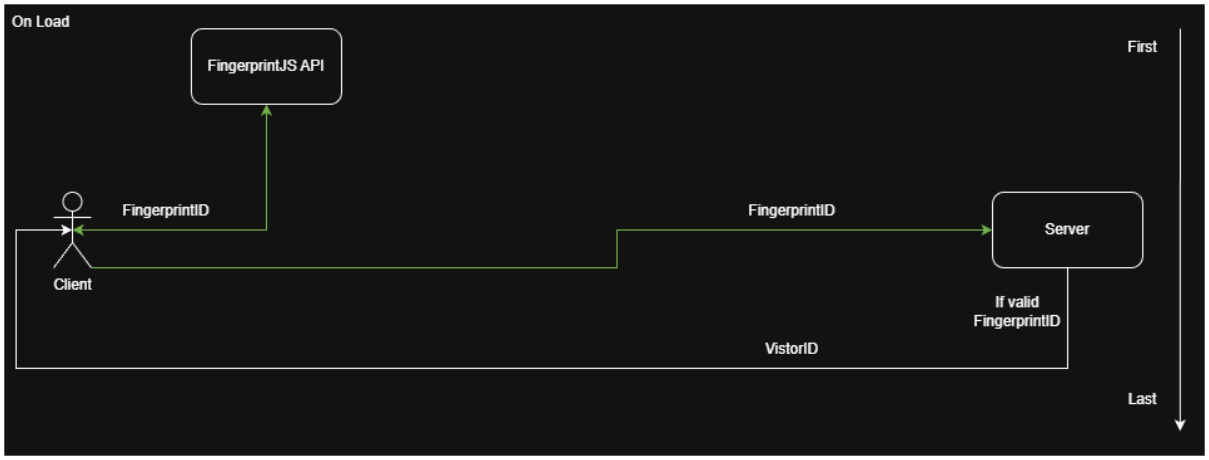


Figure 4

All devices visiting the site must be fingerprinted and registered within the server before any access is granted. The fingerprintJS API allows for easy fingerprinting of devices while also filtering API scrappers. FingerprintJS uses a browser head fingerprinting algorithm where all attributes of a browser head including but not limited to, screen size, user agent, network, to produce a unique fingerprint. The fingerprint is then registered within the server before a visitor id is returned to the client, where all future requests will be referenced with.

4.3.2 User Authentication

User Authentication should differ based on the status of the device. If the device is registered, the user should have the keys. However, in the case where a new device is used, the server should provide a new RSA key for the device.

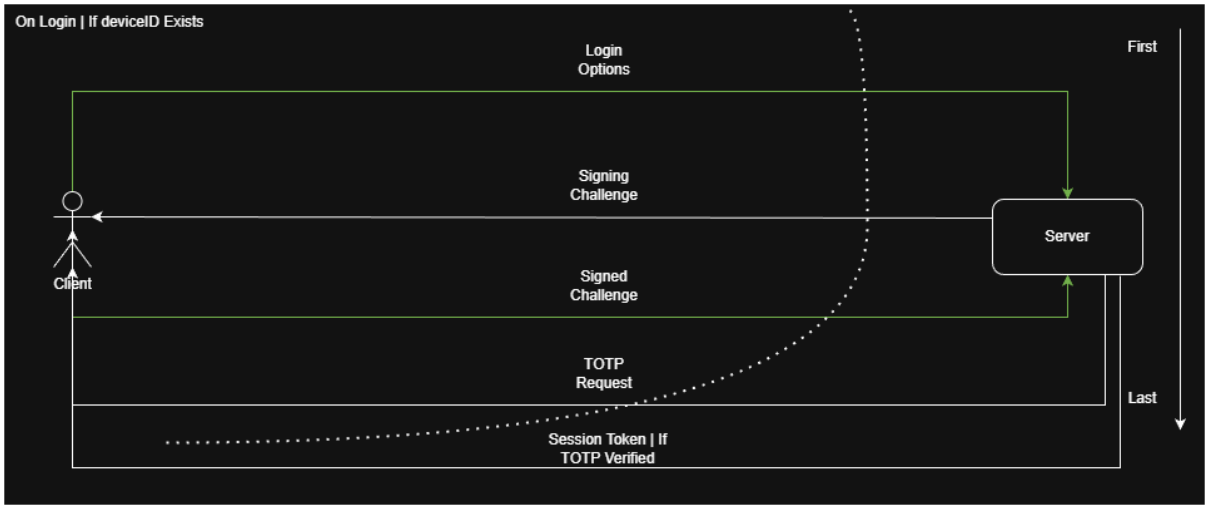


Figure 5

If the device is registered, a login attempt should be initiated by sending Login Options to the server. The server will then return a signing challenge which should be signed by the user using a private key stored during registration of the user account.

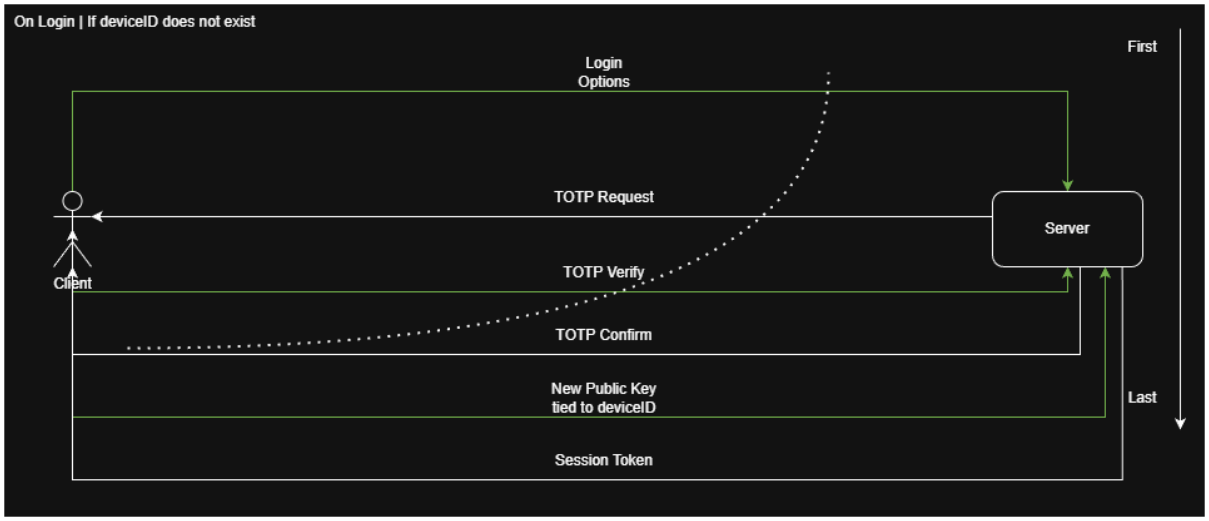


Figure 6

In the case that the device is not registered, the login attempt will be initiated using login options, before a TOTP request is initiated to verify the identity of the user and the authenticity of the request. If and only if the TOTP is verified, the client will generate a new set of RSA keypairs before sending it to the server. After authentication, the server will then return with a session token which is tied to both the public key and device id of the client.

TOTP in either case is mandatory to ensure usability while enhancing security of the system, ensuring that user accounts cannot be compromised without the compromission of their mobile devices as well.

4.3.3 User File Access

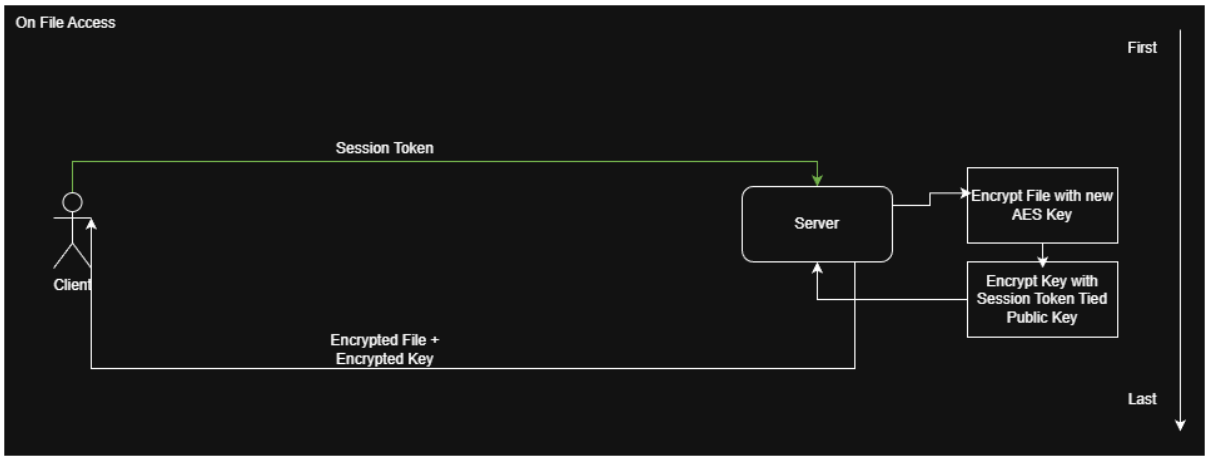


Figure 7

File access is an important feature of the demonstration and thus will be included here. All files are encrypted using the AES algorithm, taking advantage of the speed of the algorithm. The key will then be encrypted using the public key of the user that is stored alongside the session token. Encryption keys should be rotated regularly to enhance security.

4.3.4 Session Token Generation

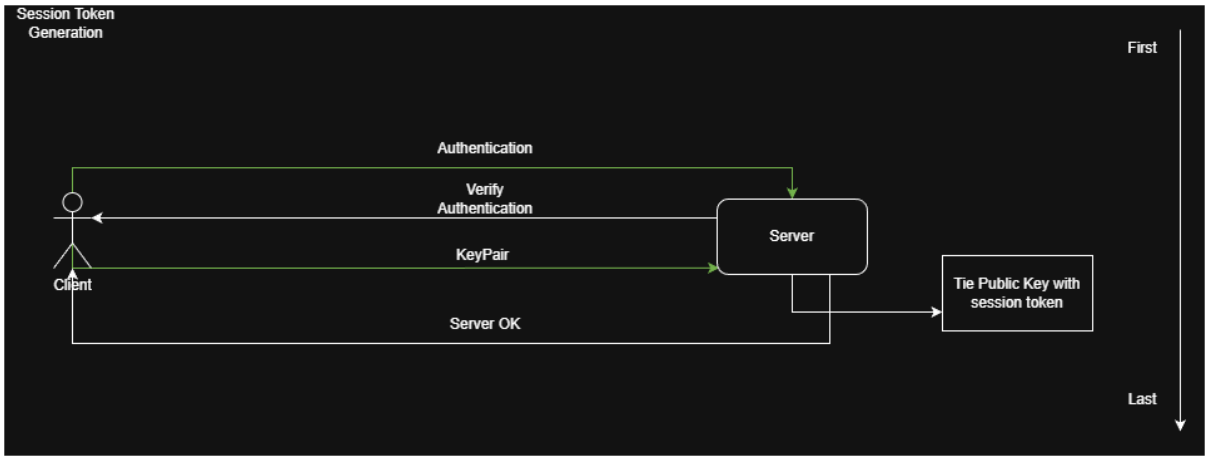


Figure 8

Session tokens are the main identifiers and access token for the service. After each authentication, the public key of the user will be sent to the server. The server will generate a true random session token with addition of the user's device id and fingerprint id to disable any possibilities of using a stolen token. The token is then stored with the public key before sending it back to the client.

4.4 Security Features

i. Continuous verification of identity & device health: Continuously reassess user identity, device posture, and session context throughout a session instead of relying on a single authentication event. These factors contribute to a live trust score utilized by Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs) to dynamically adjust access permissions.

How it mitigates attacks.

- Insider Threats: Ongoing monitoring detects anomalies in normal behaviour and necessitates re-authentication in response to abnormal activity, thereby limiting potential damage from malicious insiders.
- Credential Theft: If credentials are compromised, the attacker's device posture whether it is an unknown device or lacks proper attestation along with any unusual behaviour, will trigger re-authentication or restrict access.
- Device Compromise: Signals related to device health (such as EDR alerts or unpatched vulnerabilities) significantly lower trust levels and can block access to sensitive information.
- Session Hijacking: Continuous verification links a session to device attestation and implements periodic challenges, making it difficult for an attacker who has captured a token to reuse it without access to the device or proper attestation.

ii. Least privilege enforcement: Implement Attribute-Based Access Control (ABAC) and Role-Based Access Control (RBAC) alongside dynamic conditional access policies that consider factors such as role and resource sensitivity. Privileges should be granted on a just-in-time (JIT) and time-bound basis, while privileged operations must undergo elevated checks (PIM/JIT). The policy engine issues minimal RBAC/ABAC permissions for each session.

How it mitigates attacks.

- Insider Threats: By minimising permissions, we reduce the blast radius; users and compromised accounts only receive the access necessary for their roles, while privileged access may require additional verification.
- Credential Theft: Even if credentials are stolen, conditional controls restrict the attacker's actions (for instance, administrative workflows may be blocked unless certain criteria are met).
- Device Compromise: Access can be limited (to read-only or masked data) if the device's security posture is deemed inadequate.
- Session Hijacking: The theft of tokens or cookies results in restricted capabilities due to minimal and dynamic access rights; high-risk operations require fresh re-authentication to proceed.

iii. Micro-segmentation across distributed PDPs (policy decision points): Segment the network and workloads into fine-grained divisions, known as workload-level security zones. Position distributed Policy Decision Points (PDPs) near the data and workloads to enable local policy decisions, resulting in lower latency and more resilient outcomes. Implement distributed enforcement

mechanisms, such as Policy Enforcement Points (PEPs) and micro-firewalls or distributed firewall capabilities (like hypervisor or SDN controls), to uphold inter-segment rules effectively.

How it mitigates attacks.

- Insider Threats: Micro-segmentation inhibits lateral movement, ensuring that a compromised account or insider cannot easily access other workload zones.
- Credential Theft: In the event that an attacker compromises a single service, micro-segmentation restricts access to a small implicit trust zone, containing the breach effectively.
- Device Compromise: When endpoints are compromised, they can be rapidly isolated to their designated segment; Policy Decision Point (PDP) and Policy Enforcement Point (PEP) decisions can block network flows to sensitive areas.
- Session Hijacking: Even if a session token is replayed within one segment, cross-segment communications necessitate separate policy checks and potentially stronger attestation, thereby mitigating the impact.

iv. Strong encryption for data and communication channels: All sensitive communications employ TLS (Transport Layer Security) 1.3, utilizing mutual TLS for machine-to-machine interactions and where client certificates are deemed appropriate. Data at rest is secured through AES-256 encryption, supported by robust key management practices and regular key rotations. Tokens are both signed and encrypted using JWT/JWE standards, while session identifiers are linked to device attestation and fingerprinting to mitigate the risk of token replay.

How it mitigates attacks.

- Insider Threats: Implementing encryption limits the data that a malicious insider can access if they do not possess the encryption keys. Additionally, application-level encryption and data masking help to minimize exposure.
- Credential Theft: Encrypting token transport and utilizing mutual TLS (mTLS) significantly decrease the likelihood that an intercepted token can be exploited from a different host. Furthermore, binding tokens to specific device factors helps prevent their reuse.
- Device Compromise: In the event that data is exfiltrated from a compromised device, robust at-rest encryption ensures that the information remains unreadable without the appropriate keys.
- Session Hijacking: Employing TLS alongside binding session tokens to device fingerprints, mTLS, and utilizing short-lived tokens reduces both the time window for potential attacks and the feasibility of token replay.

v. Threat detection via behavioural anomaly analysis: A behavioural analysis engine, utilizing machine learning techniques such as Isolation Forests or ensemble models, monitors login patterns, resource access sequences, network flows, and endpoint detection and remediation (EDR) telemetry. It generates anomaly scores that contribute to trust scoring and may initiate either automated or manual investigations. By correlating telemetry from identity, network, and endpoint sources, the system aims to reduce false positives.

How it mitigates attacks.

- Insider Threats: Identifies unusual activities, such as exfiltration patterns and privilege misuse, at an early stage, triggering alerts or automatically adjusting user privileges.
- Credential Theft: Abrupt changes in geolocation, device fingerprints, or access times are monitored—stolen credentials accessed from unfamiliar contexts are promptly flagged.
- Device Compromise: Endpoint Detection and Response (EDR) signals, along with abnormal process and network behaviours, are analysed alongside identity signals to determine if a device has been compromised.
- Session Hijacking: Anomalous session behaviours, such as simultaneous logins from different geolocations, indicate potential session takeover attempts.

vi. Dynamic response (downgrading trust / blocking sessions): Automated playbooks within the PDP/PEP and SOAR integration implement dynamic responses based on trust score thresholds. These actions may include reducing session privileges, requiring step-up authentication (such as FIDO2 or biometric re-authentication), revoking refresh tokens, quarantining devices, terminating sessions, or initiating an incident workflow. Responses are categorized by severity—from limited (read-only) to complete access block—to ensure a balance between security and availability.

How it mitigates attacks.

- Insider threats: Implementing automatic throttling or privilege downgrading mitigates ongoing damage while facilitating an investigation.
- Credential theft: The immediate revocation of sessions and tokens, along with enforced re-authentication, closes off access for attackers utilizing stolen credentials.
- Device compromise: Quarantining or isolating affected devices coupled with mandatory credential rotation helps contain the breach.
- Session hijacking: Swift revocation of tokens and termination of sessions prevents further exploitation following detection.

Below shows the mapping between the mechanisms and the primary mitigation.

Mechanism	Insider threats	Credential theft	Device compromise	Session hijacking
Continuous verification	Detect anomalous user behaviour; re-auth	Blocks unknown devices; re-auth	Device posture lowers trust	Device binding prevents reuse
Least privilege (conditional access)	Limits blast radius	Limits allowed actions	Restricts sensitive ops	Minimal session privileges

Micro-segmentation	Prevents lateral movement	Containment per segment	Isolate compromised host	Cross-segment checks needed
Strong encryption (TLS, AES)	Protects data read access	Protects transport/tokens	Protects stored secrets	Prevents token sniffing
Behavioural analytics	Early detection of misuse	Flags out-of-norm usage	Detects malicious processes	Flags session anomalies
Dynamic response	Auto-throttle / revoke	Revoke tokens / require step-up	Quarantine device	Immediate session termination

Figure 9

5.0 Experimental Setup / Comparative Analysis

5.1 Experimental Objectives

The objective of this section is to establish a structured evaluation of the AB-ZTA system. The experiment focuses on analysing how the proposed system conceptually performs when compared against established Zero Trust models. The evaluation aims to determine whether the proposed approach provides architectural, operational, and security improvements based on documented behaviours and known limitations of existing frameworks. Specifically, the experimental objectives are:

i. To identify the security capabilities addressed by the proposed ZTA. This includes examining how the system mitigates common enterprise threats such as malware injection, unauthorized access, lateral movements, and device-based compromise.

ii. To analyse how key components of the proposed system align with or differ from leading ZTA frameworks. The study evaluates core features such as continuous verification and dynamic policy adaptation compared with the others.

iii. To determine which performance areas the proposed system is expected to improve. This objective examines expected enhancements in trust decision automation, policy enforcement efficiency, and visibility based on theoretical expectations and literature findings.

iv. To define how the system would be benchmarked in a real testing environment. This objective outlines which attack scenarios, operational conditions, and metrics would be used to measure performance if a simulation were conducted.

5.2 Benchmark Solution

To provide an evident comparative foundation for evaluating the AB-ZTA system, two established and widely recognized Zero Trust frameworks are selected as the benchmark solutions. There are Google BeyondCorp and Microsoft Zero Trust. These frameworks represent mature and approved by industry as the basis of most modern ZTA implementations. Thus, choosing both of the frameworks is highly suitable as a reference for conceptual and architectural comparison.

Google BeyondCorp	BeyondCorp is one of the earliest and most influential Zero Trust Frameworks. It is chosen as a benchmark because: • it is highly mature and extensively documented in both industry whitepapers and academic research. • it has proven large-scale deployment and securing Google's global workforce and infrastructure. • it highlights identity and device-centric access controls by giving insightful comparisons with AB-ZTA trust engine and policy enforcement model.

Microsoft Zero Trust Model	Microsoft's Zero Trust is selected as the second benchmark due to its strong adoption across enterprise environments and integration with widely identity and cloud platforms. It is included because: • it is one of the most widely deployed ZTA models used by organizations across critical sectors including finance and healthcare. • Microsoft provides comprehensive tooling including Azure AD, Intune, Defender, and Conditional Access Policies making it a well-rounded reference. • it represents an industry standard interpretation of ZTA with strong compatibility, visibility, and compliance support.

5.3 Simulation / Test Environment

The simulation environment described in this section serves to illustrate how the AB-ZTA would be benchmarked under realistic enterprise conditions. The purpose of defining this environment is to demonstrate how comparative testing could be carried out in future work.

5.3.1 Network Setup

The environment assumes a medium-sized enterprise network adopting Zero Trust principles. This structure reflects common enterprise architectures in the ZTA environment and allows consistent comparison with established frameworks. The network would be segmented into multiple security zones such as:

- User segment: employee workstations and mobile devices
- Application segment: internal web applications and SaaS services
- Protected segment: sensitive system such as administrative portals, databases
- Security service layer: identity provider, policy engine, device posture service, monitoring tools

5.3.2 Devices and User Roles

The evaluation environment assumes a diverse set of simulated users and devices typical of enterprise operations:

- Regular employees: authenticated through identity services
- Privileged administrators: requiring stronger policy checks and continuous verification
- Guest or temporary users: restricted to limited resource
- User devices: Windows laptops, macOS devices, iOS/Android smartphones
- Non-compliant or compromised devices: used to model potential threats

5.3.3 Applications and Access Pathways

To reflect typical enterprise workflows, the simulated environment would include:

- Cloud-based applications (productivity, collaboration, storage)

- Internal business applications hosted on segmented servers
- Administrative tools accessible only to privileged identities
- API-based access paths for automated processes

5.4 Evaluation Metrics

These metrics are derived from industry standards, vendor documentation, and research literature. They represent the key performance indicators that would be used to benchmark the AB-ZTA system against existing frameworks if a full implementation were carried out. The purpose of defining these metrics is to assess how the proposed system conceptually improves upon existing limitations and provide a foundation for future empirical evaluation.

i. Attack Detection Rate and False Positive Rate: This metrics evaluates how accurately a system distinguishes malicious activities from legitimate behaviour. Attack detection rate reflects the system's ability to identify threats such as malware injection or unauthorized lateral movement, while false positives represent legitimate action incorrectly flagged as threats.

Relevance: Zero Trust depends heavily on continuous monitoring and anomaly detection. A high detection rate with minimal false alarms ensures that the trust engine responds quickly to real threats without disrupting user operations.

ii. Authentication and Access Latency: Authentication latency refers to the time required to validate user identity, verify device posture, and enforce before granting resource access. In continuous verification models, latency also includes periodic re-authentication triggered by context changes.

Relevance: Zero Trust introduces more verification steps than traditional perimeter security. Therefore, frameworks must maintain strong security without causing noticeable delays. This metric shows how the system might reduce overhead through streamlined automation.

iii. Policy Enforcement Efficiency: This metric represents how quickly and accurately the system applies security policies when user behaviour, device health, or contextual conditions change. It also refers to the responsiveness of micro-segmentation controls and access revocation mechanisms.

Relevance: Zero Trust's core principle depends on rapid policy enforcement. Evaluating this metric helps identify whether the proposed system offers improvements through centralized policy control or automation.

iv. Scalability and Resource Utilization: Scalability refers to the system's ability to maintain performance as the number of users, devices, sessions, and policies increases. Resource utilization measures the CPU, memory, and network overhead introduced by continuous verification processes, endpoint agents, and monitoring components.

Relevance: Modern enterprises operate with growing device counts and distributed workforces. ZTA solutions with high computational overhead or poor scalability may degrade performance or require costly infrastructure upgrades.

v. Suitability for Automated Threat Response: This metric evaluates how effectively the system supports automated actions such as quarantining devices, revoking access, or applying new policies without requiring manual intervention.

Relevance: Industry literature identifies limited automation as a recurring weakness in many Zero Trust deployments. Automated orchestration significantly improves incident response speed and reduces human errors.

5.5 Comparative Analysis

Dimension	Google BeyondCorp	Microsoft Zero Trust	AB-ZTA
Primary Architectural Focus	Identity-centric access, device trust inferred through security certificates	Identity + device posture, heavily integrated with Azure A ecosystem	Balanced model combining identity, device posture, network context, and micro-segmentation
Trust Decision Model	Access proxy performs real-time identity verification, minimal network segmentation	Continuous verification via identity signals, compliance policies, and telemetry	Multi-layered trust engine using dynamic risk scoring, contextual signals, and automated policy adaptation
Policy Enforcement	Predominantly identity-driven, device compliance handled externally	Strong conditional access and device compliance policies, integrated with Microsoft Defender	Unified policy layer with dynamic orchestration and automated rule updates in response to context changes
Monitoring & Visibility	Limited internal visibility, relies on Google's infrastructure for telemetry	Strong logging and analytics through Azure Sentinel and security centre	Enhanced unified visibility through integrated monitoring, trust scoring, and anomaly detection designed to reduce blind spots
Automation Level	Low-to-moderate, requires manual configuration	Moderate, depends on Microsoft ecosystem for automation	High automation, real-time adaptive policies, automated containment, and dynamic segmentation based on risk state
Scalability	Extremely scalable but complex to replicate outside Google	Highly scalable within Microsoft ecosystem,	Designed for interoperability, aims to reduce vendor lock-in and

		limited interoperability with other vendors	scale across heterogeneous environments
Vendor Dependency	Very high	Very high	Low-to-moderate, architecture emphasizes interoperability and reduced fragmentation
Strengths	Strong identity-first model, proven large-scale success	Rich integration mature identity tools, strong behavioural analytics	Addresses fragmentation, improves automation, enhances monitoring, and enables adaptive trust decisions
Limitations	Limited visibility inti non-Google systems, not easily replicable	Heavy dependence in Microsoft tools, may create silos, limited cross-vendor flexibility	Actual performance depends on future deployment and integration

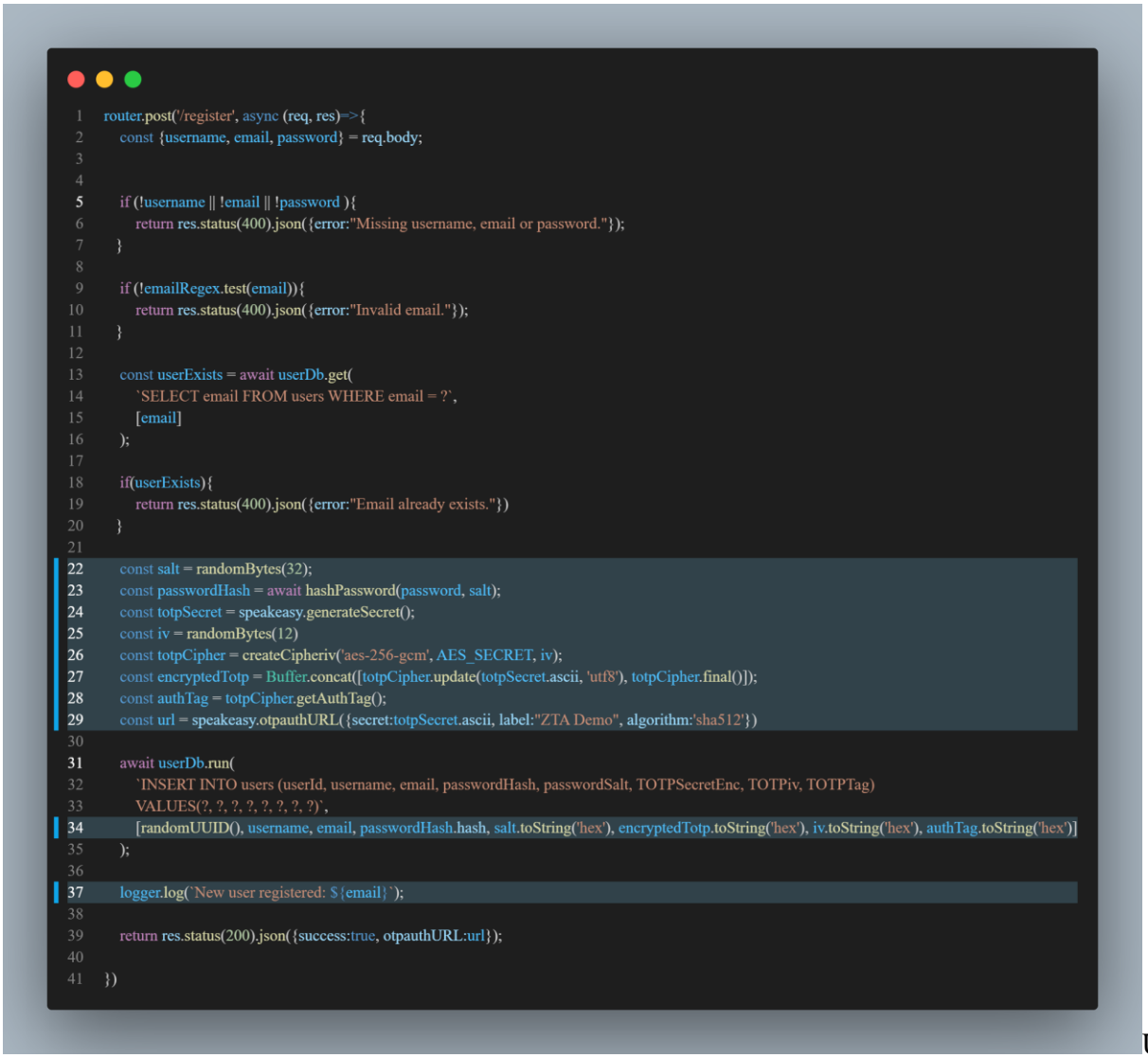
6.0 Simulation

A simulation has been made according to specifications and features listed above with some features not being incorporated due to time and resource constraints

6.1 Device Registration

All API calls made to the server must be associated with a deviceId. Devices are automatically registered when they first load the page. This deviceId is to keep track of where traffic is coming from as well as identifying any bots requesting the calls.

6.2 Signing up



```
router.post('/register', async (req, res)=>{
    const {username, email, password} = req.body;


    if (!username || !email || !password ){
        return res.status(400).json({error:"Missing username, email or password."});
    }

    if (!emailRegex.test(email)){
        return res.status(400).json({error:"Invalid email."});
    }

    const userExists = await userDb.get(
        `SELECT email FROM users WHERE email = ?`,
        [email]
    );

    if(userExists){
        return res.status(400).json({error:"Email already exists."})
    }

    const salt = randomBytes(32);
    const passwordHash = await hashPassword(password, salt);
    const totpSecret = speakeasy.generateSecret();
    const iv = randomBytes(12)
    const totpCipher = createCipheriv('aes-256-gcm', AES_SECRET, iv);
    const encryptedTotp = Buffer.concat([totpCipher.update(totpSecret.ascii, 'utf8'), totpCipher.final()]);
    const authTag = totpCipher.getAuthTag();
    const url = speakeasy.otpauthURL({secret:totpSecret.ascii, label:"ZTA Demo", algorithm:'sha512'})

    await userDb.run(
        `INSERT INTO users (userId, username, email, passwordHash, passwordSalt, TOTPSecretEnc, TOTPiv, TOTPTag)
        VALUES(?, ?, ?, ?, ?, ?, ?, ?)`,
        [randomUUID(), username, email, passwordHash.hash, salt.toString('hex'), encryptedTotp.toString('hex'), iv.toString('hex'), authTag.toString('hex')]
    );

    logger.log(`New user registered: ${email}`);

    return res.status(200).json({success:true, otpauthURL:url});

})
```

Users are allowed to send their password as raw text to the server as it is safe to do so through TLS1.3. Before storing any user information, the password is hashed with a random salt of 32 bytes. The crypto library generates a true random number and is not deterministic. A TOTP secret is also generated alongside the hash to serve as a two-factor authentication that thwarts off common 2FA attacks such as SIM-swapping.

All user data are parametrized before running any SQL queries to prevent any potential SQL Injections. Additionally, each registration event is logged to ensure a complete audit trail of system changes.

6.3 Logging In

```
if (userRecord.loginAttempts >= MAX_LOGIN_ATTEMPTS) {
    const now = Date.now();
    if (userRecord.lastLoginAttempt && (now - userRecord.lastLoginAttempt) < LOCK_TIME) {
        const remaining = Math.ceil((LOCK_TIME - (now - userRecord.lastLoginAttempt)) / 1000);
        logger.warn(`Failed login attempt for email: ${email}, deviceId: ${deviceId}`);
        return res.status(429).json({ error: `Account locked. Try again in ${remaining} seconds.` });
    } else {
        // Reset login attempts after lock period
        await userDb.run(
            `UPDATE users SET loginAttempts = 0 WHERE userId = ?`,
            [userRecord.userId]
        );
    }
}
```

If the user fails to provide the correct password, the system will log its attempt. Once the number of attempts reaches 5, the system will temporarily lock the account for 15 minutes. Once the 15 minutes is passed, the system will allow for authentication attempt again.

6.4 Session Tokens

```
export async function generateToken(userId, deviceId, ip) {
  const nonce = randomBytes(16).toString('hex');

  // Payload object
  const payloadObj = { userId, deviceId, nonce };

  // Deterministic HMAC
  const hmac = createHmac('sha256', HMAC_SECRET)
               .update(JSON.stringify(payloadObj))
               .digest('hex');

  // JWT payload
  const jwtPayload = {
    payload: payloadObj,
    hmac
  };

  const sessionToken = jwt.sign(jwtPayload, JWT_SECRET, { expiresIn: '1h' });

  const expiresAt = Date.now() + 1000 * 60 * 60;

  await sesDb.run(
    `INSERT INTO sessions(userId, deviceId, nonce, expiresAt, trustScore, token, IP)
     VALUES(?, ?, ?, ?, ?, ?, ?)`,
    [userId, deviceId, nonce, expiresAt, 100, sessionToken, ip]
  );

  return sessionToken;
}
```

Session tokens are generated with a random unique nonce. The session tokens is created with userId, deviceId and the nonce as the payload to allow for future checks. A Hmac hash is also added to the payload before JWT signing to prevent any possibility of a replay / modified session token.

```
export async function getToken(userId, deviceId, deviceIp) {
    const session = await sesDb.get(
        `SELECT token, IP, trustScore FROM sessions WHERE userId = ? AND deviceId = ?`,
        [userId, deviceId]
    );

    if (!session || !session.token) {
        return null;
    }

    let trustScore = session.trustScore;

    if (session.IP !== deviceIp) {
        // Reduce trustScore if IP mismatch
        trustScore = Math.max(0, trustScore - 10); // reduce by 10
        await sesDb.run(
            `UPDATE sessions SET trustScore = ?, IP = ? WHERE userId = ? AND deviceId = ?`,
            [trustScore, deviceIp, userId, deviceId]
        );
    }

    // Reject if trustScore too low
    if (trustScore < 80) {
        await sesDb.run(
            `DELETE FROM sessions WHERE userId = ? AND deviceId = ?`,
            [userId, deviceId]
        )
        return null;
    }

    return session.token;
}
```

Session tokens are IP matched. If the IP of the client does not match its session tokens, their trust score will drop by 10 points. Once the trust score falls below 80 points, the token is revoked.

```
export async function verifyToken(token, deviceId){

    try {
        const tokenCheck = await sesDb.get(
            `SELECT deviceId, userId, nonce, trustScore FROM sessions WHERE token = ?`,
            [token]
        );
        if (!tokenCheck){
            throw new Error();
        }
        if (tokenCheck.trustScore < 80 || tokenCheck.deviceId !== deviceId){
            throw new Error();
        }
        const decoded = jwt.verify(token, JWT_SECRET);
        const expectedHmac = createHmac('sha256', HMAC_SECRET)
                            .update(JSON.stringify(decoded.payload))
                            .digest('hex');
        if (decoded.hmac !== expectedHmac) throw new Error();
        if (decoded.payload.userId !== tokenCheck.userId) throw new Error();
        return decoded.payload.userId;
    } catch (_) {
        await sesDb.run(`DELETE FROM sessions WHERE token = ?`, [token]);
        return null;
    }

}
```

Session tokens are checked using the verification signing from the JWT secret as well as the hash generated from the Hmac digest. If either of the checks fail, the token will be revoked.

```
const sessionToken = await getToken(userId, deviceId, req.ip);

if (!sessionToken || TOTPCheck) {
    await userDb.run(
        `INSERT OR IGNORE INTO totp (deviceId, userId, expiresAt) VALUES (?, ?, ?)`,
        [deviceId, userId, Date.now() + 5 * 60 * 1000]
    );
    return res.status(401).json({ error: "TOTP required.", userId, expiresAt:Date.now() + 5 * 60 * 1000});
}

res.cookie('session', sessionToken, {
    httpOnly: true,
    secure: process.env.NODE_ENV === 'production',
    sameSite: 'lax',
    maxAge: 60 * 60 * 1000
});

logger.log(`New session token for user: ${userId}`);
```

Session tokens are the main verifier for the client beyond the login phase. Each session lasts for 1 hour and can be revoked at any time. When the account is first registered, TOTPCheck will be set to true, and user will have to verify their TOTP before being able to get a valid session token.

```
router.use(async (req, res)=>{
    const deviceId = req.cookies['x-device-id'];
    const sessionToken = req.cookies['session'];
    if (!deviceId || !sessionToken){
        return res.status(400).json({error:"Missing authentication cookies."});
    }
    const userId = await verifyToken(sessionToken, deviceId);
    if (!userId){
        return res.status(400).json({error:"Invalid session."})
    }
    req.userId = userId;
    next();
});
```

Each route will first check for a valid session token before processing any requests made by the client.

Below is a link to the AB-ZTA simulation:

AB-ZTA Simulation

Note: *Due to resource constraints, the API URL may change periodically. Please contact* ***+60193173906*** *before attempting to launch the simulation*.

7.0 Implementation Challenges & Feasibility

7.1 Deployment Challenges

No.	Challenge	Impact	Mitigation Strategy
1	Integrating continuous identity/device verification with existing IAM systems	Integration complexity may delay deployment and require major refactoring of authentication flows	Introduce a phased migration plan; use identity federation and API-based connectors to avoid full system replacement initially
2	Deploying distributed PDPs and PEPs across hybrid environments	Misalignment between environments may cause inconsistent access decisions or outages	Use standardised policy schemas (e.g., XACML/OPA Rego), deploy PDPs gradually, and validate policy consistency using automation
3	Micro-segmentation of legacy workloads	Legacy systems without agent/SDN support may not fit into granular segmentation	Introduce “macro-segmentation first," then incrementally refine segments as systems are modernised
4	Device health assessment and integration with EDR/MDM tools	False positives or incomplete telemetry may interrupt legitimate work	Start with passive monitoring mode, tune health checks, and only enforce blocking when confidence scores are stable
5	Maintaining end-to-end encryption with minimal latency	Additional encryption overhead can increase latency for high-traffic services	Use TLS session resumption, hardware acceleration (TPM/HSM), and optimise cypher suites for performance

7.2 Scalability Considerations

No.	Challenge	Impact	Mitigation Strategy
1	Scaling distributed PDPs to support large user volumes	Policy evaluation delays could bottleneck access across the organization	Deploy horizontally scalable Policy Decision Point (PDP) clusters, and implement caching along with pre-evaluated policy tokens for handling low-risk requests.
2	Handling high data flow in micro-	Excessive segmentation can create management	Utilize SDN automation, infrastructure-as-code templates, and grouping rules

	segmented networks	overhead and increase network complexity	based on labels instead of static IP addresses.
3	Growth of behavioural analytics data	Storage and analysis workloads may become expensive and slow	Implement tiered storage, event sampling, and stream-processing systems for real-time anomaly detection
4	Increased device posture queries	High-frequency checks may overload MDM/EDR systems	Adopt adaptive verification (more frequent checks only when risk increases)
5	Onboarding additional cloud platforms	Multi-cloud policy propagation may become inconsistent	Utilise a centralised policy-as-code pipeline along with vendor-neutral standards, such as OPA/Gatekeeper and SCIM.

7.3 Cost Implications

No	Challenge	Impact	Mitigation Strategy
1	Licensing costs for Zero Trust tools (IAM, EDR, MDM, micro-segmentation, SOAR)	High recurring operational expenditure	Prioritise open-source or hybrid solutions; negotiate enterprise bundles; deploy high-value components first
2	Infrastructure costs for distributed PDP/PEP nodes	Additional compute/storage requirements	Use lightweight PDPs and serverless/cloud-native deployments to reduce infrastructure load
3	Cost of training and upskilling staff	Staff may require new competencies in policy engineering, threat analytics, and ZT operations	Implement internal training programs; encourage vendor certification pathways; adopt a gradual, workload-by-workload rollout
4	Data storage and analytics costs for continuous monitoring	Behaviour analytics produces high-volume telemetry	Use retention policies, data tiering, and anomaly scoring that minimise raw data storage needs
5	Migration cost from legacy systems	Retrofitting older systems may require redevelopment	Focus initial efforts on high-risk, modern applications; isolate or containerize legacy systems where possible

7.4 Ethical and Privacy Concerns

No	Challenge	Impact	Mitigation Strategy
1	Extensive monitoring of user behaviour may raise privacy concerns	Employees may feel surveyed, causing decreased trust or resistance	Provide transparency on what is monitored; ensure monitoring is purpose-limited and reviewed by governance committees
2	Sensitive personal data handled during authentication (biometrics, device IDs)	Potential misuse or unauthorised access to sensitive identity data	Apply strict data minimisation, encryption, and role-based access to identity datasets; include privacy-by-design controls
z	Automated risk scoring may unintentionally disadvantage certain user groups	Bias in anomaly detection may produce false positives against certain regions or working patterns	Periodically audit ML models; implement human-in-the-loop review for high-impact decisions
4	Over-collection of device telemetry	Employees may worry about personal device privacy (BYOD)	Enforce boundaries: collect only security-relevant attributes; use containerised work profiles for BYOD
5	Ethical risks of automatic session blocking	Sudden blocks may disrupt critical tasks	Use graded response steps before full blocking; require human review for high-impact events when possible

7.5 User Adoption & Organizational Readiness

No	Challenge	Impact	Mitigation Strategy
1	Resistance to frequent identity verification steps	Users may find the system inconvenient, increasing complaints	Use frictionless authentication (FIDO2, biometrics, device certificates) and adaptive frequency based on real-time risk
2	Lack of awareness about Zero Trust concepts	Users may not understand why controls are stricter than before	Conduct awareness campaigns, training, short videos, and explain the security benefits clearly
3	Operational teams overwhelmed by new responsibilities	The learning curve for micro-segmentation,	Establish a dedicated Zero Trust task force; provide hands-on labs and mentorship programs

		policies, and behavioural analytics can be steep	
4	Poor readiness of legacy systems	Some systems cannot support dynamic policies or fine-grained access	Prioritize modernization roadmap; apply compensating controls (network isolation, API gateways)
5	Cultural shift from perimeter-based security to continuous verification	Teams used to “trust by default” may find ZT disruptive	Gradually implement controls and celebrate early wins to increase buy-in across departments

8.0 Evaluation & Discussion

8.1 Findings

The assessment and simulated implementation of the Adaptive Behavior-Centric Zero Trust Architecture (AB-ZTA) indicate that the proposed model successfully achieves its intended security improvements. The findings show that the architecture enhances threat visibility, strengthens access control accuracy, and limits the impact of potential breaches.

Firstly, the integration of continuous behavioral monitoring with an adaptive trust-scoring mechanism allows the system to identify unusual or suspicious activities much sooner than traditional rule-based access control methods. During the simulation, anomaly-detection techniques (such as behavioral baselines and isolation-forest style modelling) detected irregular user patterns that would normally bypass static authentication systems. This early detection significantly increases the chances of preventing compromised sessions or insider-related misuse.

Secondly, the evaluation of micro-segmentation and distributed Policy Decision Points (PDPs) demonstrated a clear reduction in lateral movement opportunities during simulated attack scenarios. Instead of allowing an attacker to move freely across the internal network, the system restricted access to a single segment or workload zone. This containment effect greatly reduces the blast radius and prevents widespread compromise.

Thirdly, the “data shielding” concept — particularly dynamic masking, encryption enforcement, and device-bound session controls — effectively reduced the exposure of sensitive information. When the system detected an untrusted or non-compliant device, it automatically downgraded permissions, applied tighter encryption, or restricted access to sensitive content. This ensured that even if access was granted, the data remained protected.

Lastly, the findings acknowledge some performance considerations. The additional behavioral telemetry and continuous trust evaluations introduced a slight increase in authentication time and processing overhead. However, these drawbacks were minimized through techniques such as trust-score caching, reducing verification frequency for low-risk users, and utilizing tiered analytical storage. Overall, these performance impacts were acceptable and did not outweigh the significant security benefits.

8.2 Strengths and Advantages

The AB-ZTA model offers several strong advantages that significantly enhance the security posture of modern organization’s:

- **Real-time, behaviour-driven protection:** The architecture continuously evaluates user behavior instead of relying on a single login checkpoint. By analyzing patterns such as access frequency, device activity, location, or resource usage, the system can dynamically determine if a session becomes risky. This continuous validation approach greatly improves the detection of credential theft, phishing, account misuse, and insider threats.

- **Risk-adaptive access control:** The trust-score mechanism allows the system to apply different levels of access depending on the calculated risk at any given moment. Instead of simply allowing or denying access, the system can downgrade permissions, restrict certain functions, or enforce additional authentication steps. This ensures that security is applied in a balanced and user-friendly way, reducing unnecessary disruptions while still maintaining strong protection.
- **Highly effective containment with micro-segmentation:** The use of small, isolated network segments prevents attackers from moving freely if they gain initial access. Distributed PDPs make decisions locally, improving response speed and ensuring that security does not depend on a single centralized controller. This structure limits the scope of an attack and helps maintain business continuity even during a security incident.
- **Strong, standards-based cryptographic controls:** The architecture adopts modern and widely accepted security standards such as FIDO2 for resistant authentication, TLS 1.3/mTLS for secure communication channels, AES-256 for storage encryption, and secure token frameworks such as JWT/JWE. These technologies ensure that both authentication and data transmission are robust against modern threats.
- **Interoperable and flexible design:** Because AB-ZTA incorporates multiple policy models — such as ABAC, RBAC, XACML-based evaluations, and behavioral analytics — it can work across different systems and platforms. This reduces vendor lock-in and supports gradual, step-by-step deployment across existing infrastructure while maintaining compatibility with future technologies.
- **Automated incident response and efficient containment:** The integration of SOAR playbooks and automated risk responses allows the system to take immediate action during suspicious events. The architecture can revoke tokens, isolate devices, quarantine network segments, downgrade privileges, or trigger MFA without waiting for manual intervention. This automation greatly reduces response time and increases the organization's ability to contain threats effectively.

8.3 Weaknesses and Limitations

Even though AB-ZTA model does have more to offer over the classical models of the ZTA, there are still several weaknesses and limitations:

- **Raise in computational and processing overhead:** Monitoring behavior continuously, device posture, and real time recalculation of trust score adds to the overhead of the system. This can cause small delays in big enterprise networks during times of peak authentication, or in assessing more complex access control policies.
- **Generate on quality telemetry data:** AB-ZTA is strongly dependent on the correct metadata of devices, logs of the behaviour, and network telemetry. When the tools of complete monitoring are not used, or where the devices are outdated, or there is less visibility, the decisions based on trust-score can be less accurate or have false positives/negatives.
- **Policy and deployment engineering complexity:** The system consists of several elements distributed by PDPs, behavioural analysis modules, and ABAC/RBAC policies that should

work together. Without specialized security teams, designing the consistency of cross-environment policies (cloud, hybrid, legacy) can be problematic, and slows the adoption.

- **Friction of potential user experience:** Even though AB-ZTA is supposed to reduce interruptions, adaptive verification can still cause unforeseen MFA requests or a temporary block-out in case anomalies are detected. That may frustrate the user especially in high-pressure working conditions.
- **Compatibility with legacy system limitation:** The older applications and on-premises systems might lack support in continuous authentication, behavioural telemetry, and micro-segmentation in a fine manner. It might involve the use of more gateways, wrappers or partial isolation schemes which raise both the cost and overhead of operation.
- **Behavioral analytics weaknesses:** Although machine-learning based detection can work, behavioural baselines can take time before they reach a mature point, and false alerts may be generated by an early detection. Users with unusual work schedules (e.g. travelling employees, contractors) can generate non-regular behaviour profiles making it tricky to trust score.

9.0 Conclusion & Future Work

9.1 Conclusion

This paper proposed the Adaptive Behaviour-Centric Zero Trust Architecture (AB-ZTA) as an evolution of traditional Zero Trust models. By integrating continuous behavioural checks, real-time adaptive trust scoring, distributed decision points, dynamic enforcement, and context-aware data shielding, the architecture reduces the attack surface, improves anomaly detection, and provides stronger containment than static policy-based systems. The conceptual assessment and simulated comparison against mature industry benchmarks demonstrate that AB-ZTA enhances visibility across identities, devices, networks, and workloads; limits lateral movement through micro-segmentation and local PDPs; and restricts sensitive data exposure using dynamic data shielding. Although behavioural telemetry and trust calculations introduce performance overhead, features such as trust-score caching, tiered analytics, and adaptive verification frequencies keep these impacts manageable. Overall, AB-ZTA addresses key research gaps, including vendor lock-in, uneven visibility across identity and workloads, and the limitations of static conditional-access rules by introducing a responsive and standards-based trust engine. The architecture offers a more defensible, context-aware, and usable Zero Trust approach, with several opportunities for further research and real-world development.

9.2 Recommendations

Future enhancements for the AB-ZTA model should focus on expanding its automation, scalability, and integration capabilities to support broader enterprise adoption. One recommended direction is the incorporation of more advanced machine-learning pipelines such as deep learning for behavioural prediction or federated learning to improve threat detection without exposing raw user data. Additionally, future work could include developing standardized APIs to improve interoperability with SIEM, SOAR, and cloud-native security platforms, enabling seamless deployment across hybrid or multi-cloud environments. Performance optimization should also be explored, particularly in reducing the processing overhead caused by continuous behavioural analytics. Finally, pilot testing in real-world enterprise networks, including diverse user groups and device types, would help validate scalability, refine trust-scoring thresholds, and strengthen the model's resilience against emerging attack patterns such as AI-driven intrusion attempts or adaptive social-engineering campaigns.

References

[1] Albakri, S. H., Shanmugam, B., Samy, G. N., Idris, N. B., & Ahmed, A. (2014). Traditional Security Risk Assessment Methods in Cloud Computing Environment: Usability Analysis. Jurnal Teknologi (Sciences & Engineering), 73(2).

[2] Alhomdy, S., Thabit, F., Abdulrazzak, F. H., Haldorai, A., & Jagtap, S. (2021). The role of cloud computing technology: A savior to fight the lockdown in COVID 19 crisis, the benefits, characteristics, and applications. International Journal of Intelligent Networks, 2, 166-174.

[3] Case, T. L. (2025). Enterprise Networks Infrastructure & Security. Prospect Press.

[4] Cimpanu, C. (2020). ZDNET. Retrieved October 16, 2025, from https://www.zdnet.com/article/microsoft-fireeye-confirm-solarwinds-supply-chain-attack/

[5] Din, S. N. U., Muzammal, S. M., Bibi, R., Tayyab, M., Jhanjhi, N. Z., & Habib, M. (2024). Securing the Internet of Things in Logistics: Challenges, Solutions, and the Role of Machine Learning in Anomaly Detection. In Digital transformation for improved industry and supply chain performance (pp. 133-165). IGI Global Scientific Publishing.

[6] Google. (n.d.). BeyondCorp. Retrieved November 13, 2025, from https://www.beyondcorp.com/

[7] Linqiang, Y., Sindiramutty, S. R. a. L., Ashraf, H., Muzammal, S. M., Balakrishnan, S. a. P., Gupta, S., & Kavita, N. (2024). Intelligent Household Waste Classification System Based on Machine Learning. *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, 760–768. https://doi.org/10.1109/etncc63262.2024.10767563

[8] MarketsandMarkets. (2025). Enterprise Networking Market.

[9] Microsoft, (no date). Microsoft Entra conditional access overview. [online] Microsoft Learn. Available at: https://learn.microsoft.com/en-us/entra/identity/conditional-access/overview

[10] Najmi, K. Y., AlZain, M. A., Masud, M., Jhanjhi, N., Al-Amri, J., & Baz, M. (2021). A survey on security threats and countermeasures in IoT to achieve users confidentiality and reliability. *Materials Today Proceedings*, *81*, 377–382. https://doi.org/10.1016/j.matpr.2021.03.417

[11] National Institute of Standards and Technology, 2020. Zero Trust Architecture (NIST SP 800-207). [online] Gaithersburg, MD: NIST. Available at: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf

[12] Pal, S., Jhanjhi, N. Z., Abdulbaqi, A. S., Akila, D., Alsubaei, F. S., & Almazroi, A. A. (2023). An intelligent task scheduling model for hybrid internet of things and cloud environment for big data applications. *Sustainability*, *15*(6), 5104. https://doi.org/10.3390/su15065104

[13] Ponnusamy, V., Humayun, M., Jhanjhi, N. Z., Yichiet, A., & Almufareh, M. F. (2021). Intrusion detection systems in internet of things and mobile Ad-Hoc networks. *Computer Systems Science and Engineering*, *40*(3), 1199–1215. https://doi.org/10.32604/csse.2022.018518

[14] R, S., Sl, A., Chatterjee, J. M., Alaboudi, A., & Jhanjhi, N. (2021). A machine learning way to classify autism spectrum Disorder. *International Journal of Emerging Technologies in Learning (iJET)*, *16*(06), 182. https://doi.org/10.3991/ijet.v16i06.19559

[15] Ren, A., Liang, C., Hyug, I., Broh, S., & Jhanjhi, N. (2018). A Three-Level ransomware detection and Prevention mechanism. *EAI Endorsed Transactions on Energy Web*, *0*(0), 162691. https://doi.org/10.4108/eai.13-7-2018.162691

[16] Rheault, D. (2025). Why Network Complexity Kills Security. Retrieved October 9, 2025, from https://www.tufin.com/blog/why-network-complexity-kills-security

[17] Riskhan, B., Roua, A., Mahaba, B. S., Uswa, D., Gheisari, M., & Sindiramutty, S. R. (2025). Enhancing Diagnostic Accuracy for Breast Cancer Using Classical-Quantum Hybrid and Transfer Learning Technique. *Journal of Soft Computing and Data Mining, 6*(2), 41–56. https://penerbit.uthm.edu.my/ojs/index.php/jscdm/article/view/22061

[18] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero Trust Architecture (NIST Special Publication 800-27). National Institute of Standards and Technology.

[19] Sanjaya, F. N., Balakrishnan, S., Sindiramutty, S. R., & Narputro, P. (2025). XAI Technology in EEG Signal Based Disease Detection Models. *2025 International Conference on Metaverse and Current Trends in Computing (ICMCTC)*, 1–7. https://doi.org/10.1109/icmctc62214.2025.11196271

[20] Sebastian, D. (2021). The Wall Street Journal. Retrieved October 16, 2025, from https://www.wsj.com/tech/cybersecurity/solarwinds-discloses-earlier-evidence-of-hack-11610473937

[21] Shah, I. A., Sial, Q., Jhanjhi, N. Z., & Gaur, L. (2022). Use cases for Digital Twin. In *Advances in medical technologies and clinical practice book series* (pp. 102–118). https://doi.org/10.4018/978-1-6684-5925-6.ch007

[22] Tayyab, M., Marjani, M., Jhanjhi, N., Hashem, I. a. T., Usmani, R. S. A., & Qamar, F. (2023). A comprehensive review on deep learning algorithms: Security and privacy issues. *Computers & Security*, *131*, 103297. https://doi.org/10.1016/j.cose.2023.103297

[23] Ward, R., & Beyer, B. (2014). BeyondCorp A New Approach to Enterprise Security. ;login: The Magazine of USENIX & SAGE, 39(6).

[24] Weiqi, X., Hooi, S. T. C., Sindiramutty, S. R. a. L., Asirvatham, D. a. L., Kumar, D., & Verma, S. (2024). Surface Anomaly Detection Using Machine Learning Technique. *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, 1–7. https://doi.org/10.1109/etncc63262.2024.10767562

[25] Ying, X., Murugesan, R. K., Sindiramutty, S. R., Wei, G. W., Balakrishnan, S., Kumar, D., & Verma, S. (2024). Scene Text Recognition using Deep Learning Techniques. *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, 1–9. https://doi.org/10.1109/etncc63262.2024.10767484

[26] Zahra, F., Jhanjhi, N., Brohi, S. N., Khan, N. A., Masud, M., & AlZain, M. A. (2022). Rank and wormhole attack detection model for RPL-Based internet of things using Machine learning. *Sensors*, *22*(18), 6765. https://doi.org/10.3390/s22186765

[27] Zaman, S. K. U., Jehangiri, A. I., Maqsood, T., Umar, A. I., Khan, M. A., Jhanjhi, N. Z., Shorfuzzaman, M., & Masud, M. (2022). COME-UP: Computation Offloading in Mobile Edge Computing with LSTM Based User Direction Prediction. *Applied Sciences*, *12*(7), 3312. https://doi.org/10.3390/app12073312

Appendices

Appendix A

Reference Documents for Existing Framework

No.	Framework	Document Title	Source/ Publisher	Link
1.	Google BeyondCorp	BeyondCorp: A New Approach to Enterprise Security	USENIX	https://www.usenix.org/system/files/login/articles/login_dec14_02_ward.pdf
2.	Microsoft Zero Trust	Microsoft Zero Trust Guidance Centre	Microsoft Learn	https://learn.microsoft.com/en-us/security/zero-trust/
3.	Cisco ZTA	Cisco Zero Trust Architecture Guide	Cisco	https://www.cisco.com/c/en/us/solutions/collateral/enterprise/design-zone-security/zt-ag.html
4.	VMWare Zero Trust	The Zero Trust Evolution: A Practical Guide of the First Steps	VMWare	https://www.vmware.com/docs/vmware-zero-trust-evolution
5.	Palo Alto Zero Trust Enterprise	Zero Trust Enterprise: Design Guide	Palo Alto	https://www.paloaltonetworks.com/resources/guides/zero-trust-overview
6.	Zero Trust Access Management	Getting Started with Zero Trust Access Management	Okta	https://www.okta.com/sites/default/files/2021-11/Whitepaper-Getting-Started-Zero-Trust_2021.pdf