Implementing Zero Trust in Federated Identity Systems Using Hardware Attestation and Adaptive Multi-Factor Authentication (MFA)

by

MD Monzurul Islam
22101124
Jaima Zahin
22101129
Aditya Das Swaccho
24141269
Salfi Ahsan
22101142
Mahtab Uddin Al Hasan
24141273

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
January 2026.

Declaration

It is hereby declared that

1. The final year thesis submitted is our own original work while completing degree at Brac University.
2. The final year thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The final year thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

MD Monzurul Islam

22101124

Jaima Zahin

22101129

Aditya Das Swaccho

24141269

Salfi Ahsan

22101142

Mahtab Uddin Al Hasan

24141273

Approval

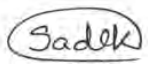
The final year thesis titled "Implementing Zero Trust in Federated Identity Systems Using Hardware Attestation and Adaptive Multi-Factor Authentication (MFA)" submitted by

1. MD Monzurul Islam (22101124)
2. Jaima Zahin (22101129)
3. Aditya Das Swaccho (24141269)
4. Salfi Ahsan (22101142)
5. Mahtab Uddin Al Hasan (24141273)

of Fall, 2025 have been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in Final Year.

Examining Committee:

Supervisor:
(Member)



Dr. Md Sadek Ferdous

Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In an era defined by dynamic cyber threats, old perimeter-style security models are becoming increasingly obsolete. Federated Identity Systems (FIS), where users authenticate across multiple domains based on a single identity provider (IdP), offer convenience but are accompanied by significant security threats, including token theft, IdP compromise, and static session validation. At the same time, Zero Trust Architecture (ZTA) has emerged as a strong framework that applies continuous verification and least-privilege access regardless of user location or network trust. This research proposes a comprehensive solution for deploying the Zero Trust approach to Federated Identity Systems, which adjusts authentication needs based on real-time risk evaluations. It integrates Adaptive Multi-Factor Authentication (MFA) and hardware attestation, like TPM (Trusted Platform Module), to detect and register devices that establish trust, ensuring that compromised and untrusted devices do not enter the federation. The system designs and deploys a safe, privacy-preserving federated identity system utilizing OpenID Connect and dynamic policy verification. After threat modeling, requirements analysis, and performance testing, the proposed system demonstrates increased protection against identity-based attacks and remains user-friendly and interoperable. Combining device trust, which is supported by hardware and adaptive authentication with contextual risk assessment, this architecture provides a comprehensive trust model that helps to mitigate threats of credential abuse, device spoofing, and token misuse. The approach increases security in federated environments and minimizes friction among users to provide an effective, scalable secure digital identity solution.

Keywords: Zero Trust Architecture, Federated Identity System, Adaptive MFA, Authentication, Trusted Platform Module (TPM), Hardware Attestation, Hardware Security Module (HSM), OpenID Connect, Security, Token, Performance Evaluation.

Acknowledgement

First of all, all praise and gratitude to Almighty Allah, by whose grace and blessings we have successfully completed this thesis without any major obstacles. We are deeply grateful to our supervisor, Dr. Md Sadek Ferdous Sir, for his invaluable guidance, constant support, and insightful advice throughout this journey. His encouragement and timely assistance were instrumental in overcoming challenges in our work. We also extend our sincere thanks to Yeasin Sir for his mentorship, constructive feedback, and unwavering support, which greatly contributed to the completion of this research. Lastly, we owe our deepest gratitude to our parents for their endless love, sacrifices, and prayers. Without their unwavering support and encouragement, this achievement and our journey toward graduation would not have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The remote work and cloud services digital era is upon us, and so is the question of the remote work setting. The standard, castle-and-moat security does not apply anymore [38]. Consumers use systems in personal devices, geographically, and using third-party services; therefore, it does not matter whether security is perimeter-centric. In order to ease identity control in this. Federated identity systems have been developed, making it possible to authenticate and transparently use various services in different domains [37]. Although this makes the user experience easier, it also brings about new security threats. Federated systems are extending trust that an organization is not directly in charge of. Interesting targets of credential theft, phishing, token theft, and insider threats. These new risks require a change of mindset where there is not an implicit trust, but an adaptive, verification-driven model. This is where Zero Trust Architecture (ZTA) comes in and plays its role. It needs to ensure every request to access is always validated, irrespective of the source or the location, and denies the principle of default trust [23].

ZTA implements strict access control and continuous authentication regardless of the user's place or network context, which is governed by the principle of never trust, always verify [23]. IdPs and Service Providers operate under federations of trust dependence relationships; ZTA avoids risk by classifying all access. Untrusted requests are considered authenticated until they are [10]. Normal Multi-Factor Authentication (MFA) systems always use fixed security policies, resulting in tradeoffs of usability or failure to take into account adaptive response to new developing attacks [20].

In order to accomplish this, adaptive MFA adapts the authentication requirements according to the change warnings of danger in real-time, including user behavior, device posture, and geolocation [34]. Applied to a Zero Trust-enabled federation structure, Adaptive MFA gives organizations the ability to deliver fine-grained security without compromising on user experience. As an example, a low-risk access request by a secure device can take the form of a password only, whereas a risky situation (e.g., an unfamiliar location) causes step-up authentication [16].

Zero Trust Architecture (ZTA) in isolation has already received extensive research in previous studies and Multi-Factor Authentication (MFA) and networks [39] in centralized systems [28]. Nonetheless, the adaptive MFA integration into a federated

environment based on Zero Trust is an open challenge. This paper fills that gap by suggesting a coherent structure that implements continuous and risk-assessment authentication across distributed trust boundaries without affecting usability and scaling.

1.1 Background

With remote working, cloud computing, and cross-domain digital collaboration that were growing in scale, the old security model based on perimeter protection became more and more redundant. Federated Identity Systems (FIS) came up as the solution to the requirement of transparent authentication across domains to enable users to log in. The adoption of a central Identity Provider (IdP). Although this makes it easier to use, it also brings forth great threats, including the stealing of tokens, excessive dependency on IdP, and not verifying sessions. The current literature has investigated federated protocols, risks of identity federation, and static multi-factor authentication (MFA), yet it has had the tendency to ignore dynamic and changing threat environments in federated environments.
One possible solution that could make use of the maxim is that of Zero Trust Architecture (ZTA), "Never trust, always verify". ZTA has the capability to support ongoing and context-dependent access control, which is dependent on user actions, on the condition of the devices, and the environment. ZTA can provide it by integrating Adaptive Multi-Factor Authentication (AMFA), which varies authentication requirements based on real-time actual risk.

The gap is filled in this work through a proposed solution integrated to harmonize ZTA, FIS, and AMFA to implement real-time federated risk-aware authentication between the borders. The solution will help to enhance security, preserve interoperability, and deliver user experience with highly distributed, sophisticated environments.

1.2 Research Motivation

The traditional "trust but verify" security model has proved to be inadequate for the globalized digital world of today. The "trust but verify" approach, which is typically used inside a secured perimeter, allows initial access based on presumed confidence while carrying out ongoing or event-based verification. When the perimeter collapses, networks are vulnerable to insider threats and lateral movement since this paradigm assumes that internal actors and systems are safe [23]. In the era of widespread cyberattacks, data loss, and advanced attack vectors, enterprises cannot depend on perimeter-based models that automatically trust internal network traffic due to prior investments in network infrastructure and software. Hackers used credentials taken from a third-party HVAC provider to infiltrate Target's network in 2013. The attackers proceeded laterally and grabbed data from over 40 million credit and debit cards without being detected since the system, by default, trusted internal traffic [9]. In addition, T-Mobile agreed to a 500 million USD settlement following a string of data breaches that occurred between 2021 and 2023 and pledged to use Zero Trust Architecture to restore customer confidence [36]. To stop future

invasions, the settlement called for tighter access limits, ongoing monitoring, and stronger identity verification.

In a globalised digital world, the use of static systems of implicit trust is no longer enough. These consist of the manipulation of identities across realms [3], the execution of standardised security rules [17], and preventing cross-lateral mobility threats. Having obtained an entry point by one of the federation members. Having the philosophy of never trust, always verify, the Zero Trust architecture based on OpenID Connect offers an upgrade of identity-based access and context management. Zero Trust within federated systems is still very underresearched, notwithstanding the fact that it has been widely researched in single-organization studies. In federated systems where trust will have to be negotiated between autonomous domains, the current methods often assume the existence of a single control of network resources [30].

A motivation to address this gap is a primary reason behind this thesis. Considering the distributed nature of federated systems, such signs of policy variation, and the necessity of safe but smooth cooperation, this research aims to research how the principles of Zero Trust Architecture(ZTA) could be implemented in them successfully.

Addressing these concerns, the research will offer a valuable framework that a company may apply to implement Zero Trust in federated systems that ensure the presence of high-level security without affecting operational independence and trust relationships. Furthermore, a solution to this issue will support secure and trusted exchanges of identity between identity and service providers, enhance inter-organization interaction, and enhance trustworthiness, reputation, and legal compliance in federated settings.

1.3 Problem Statement

Since federated identity systems (FIS) enable users to log in to multiple services using a single identity provider (IDP), they have become an important part of the modern management of digital identities. This method enhances convenience and reduces password fatigue; there are several security weaknesses that undermine the confidentiality and integrity of a digital identity. The token theft or token misuse (i.e., SAML assertion) that may be gained by means of token capture, or through token exposure via attacker extensions, unsafe token cache, or cross-site scripting (XSS) is among the most important risks [15]. The actual events, such as the 2020 SolarWinds attack, have proven that once stolen, such tokens can be reused in order to gain unauthorized access [25].

Secondly, the federated systems tend to be over-reliant on the use of IdP as an authenticating authority and identity verifier. When the IdP is compromised or poorly configured, then attackers can broadcast malicious statements to the many service providers (SPs) and sabotage the entire chain of trust [13].

Lastly, the federated architecture that is traditionally used usually does not provide continuous session evaluation, which assigns long-lived session tokens without checking user or device context after authentication. Such a static approach is incompatible with the dynamic environments, where risk profiles evolve very fast, exposing systems to session hijacking or subsequent lateral movement by adversaries [23].

1.4 Objective

This study is mostly about the creation of a Federated Identity system, which is integrated with Zero Trust Architecture. This entails coming up with a secure authentication system which takes a security analysis and performance analysis to address the current loopholes of a Federated Identity system.

RO1 Examination of existing authentication systems: With regard to the past authentication mechanisms, we aim to recognize the way they operate in the modern security protocols. This also includes the review of existing authentication processes in terms of ZTA in an effort to determine whether they have possible security or privacy vulnerabilities.

RO2 Research the effectiveness of ZTA and MFA in Federated Identity: Our target is to find out how the concepts of ZTA and adaptive MFA may be combined to form holistic and privacy-centered authentication in Federated Identity systems.

RO3 Design and develop the proposed system: We aim to design and deploy an adaptive authenticated system of Federated Identity, which is based on ZTA principles. Objectivity is gained by breaking the existing barriers without compromising the authentication requirements.

RO4 Performance testing and evaluation: We aim to test the developed system properly with the help of diverse analytic tools, including security, performance, and compatibility. This includes carrying out attack testing and evaluating the system performance. Assure and confirm that industry standards, cryptographic security, and defined engineering practices of the system adhere to the security and privacy standards.

RO5 Identify and mitigate risks: We also aim to identify and mitigate security and privacy threats in federated identity authentication flows by integrating adaptive policies that align with Zero Trust principles and preserve user data confidentiality.

1.5 Methodology in Brief

This work is based on an incremental method to develop and validate a secure federated identity system based on Zero Trust Architecture (ZTA) and Adaptive Multi-Factor Authentication (AMFA). It begins by establishing the basic problems in current federated identity models, such as token misuse and lack of continuous

authentication. An extensive literature review supports structuring the work and describing how ZTA and AMFA would overcome these deficits.

Threat modeling using the STRIDE model is performed in order to forecast potential attacks and use as guidance for security design. These lead to the development of functional and security requirements that, in turn, result in the design of a federated system capable of integrating ZTA using OpenID Connect and AMFA.

A prototype is being created to demonstrate the purported solution, subsequently tested for performance, usability, and security. Every activity is consistent with the research objectives, so the framework works, is scalable, and performs in actual federated environments.

1.6 Scopes and Challenges

This research focuses on Zero Trust architecture in Federated Systems environments where several independent organizations collaborate while maintaining separate governance and infrastructure. The OpenID Connect is included in this research in federated system architecture, which needs dynamic and secure microservice-to-microservice communication. The study investigates technical identity management, further perpetuation, and network segmentation solutions in such an environment, and under stringent security controls, autonomy is maximized.

Organizational challenges and technical challenges are also present. In technical terms, Zero Trust in federation brings about a scalability issue, interoperability between heterogeneous infrastructures, distributed enforcement of policy, and preservation of privacy through correlation of identities. Organizational factors include establishing an initial trust, policy disagreement management, incident response coordination, and shared responsibility models, and must have high-fidelity governance mechanisms and collaborative processes. Zero Trust also needs to serve insider threat, cross-domain attacks, subtle key management, and integrated audit trail needs without the loss of information sovereignty.

Operationally, a transition to Zero Trust can be potentially disruptive to business continuity, expensive to undertake, and it exposes competence gaps across the partner organizations. Research-wise, constructing realistic test structures, developing across-organization measures, addressing ethical dilemmas, and conducting long-term studies are complicated but inevitable. This thesis aims at overcoming challenges through the creation of reference architectures, establishing evaluation frameworks, industry collaboration, and continuously improving its methods as it gets feedback in real-life situations.

1.7 Report Structure

In this paper, every chapter has its own purpose of leading the reader throughout the paper's journey.

Chapter 1: This chapter sets the grounds wherein it reveals the issue, research purposes, and the rationale in delving into the area of secure federated identity systems.

Chapter 2: The chapter is devoted to the background knowledge, which provides a critical review of the literature on the existing authentication techniques, Zero Trust Architecture (ZTA), including adaptive Multi-Factor Authentication (MFA) systems.

Chapter 3: It describes the system specifications identified through the determination of functional and security requirements and inherent limitations of the system. It is based on the STRIDE threat modeling framework to detect the vulnerability like spoofing and elevation. Additionally, it presents the system's adherence to industrial codes and practices.

Chapter 4: This chapter entails a description of the proposed system architecture, which connects federated identity protocols to the Zero Trust paradigms. It outlines the design procedure, isolating the framework as the User End (client device with TPM) and the System End (backend services and IdP). Here, the workflow of continuous verification, adaptive access control, and hardware-based authentication is also configured.

Chapter 5: This chapter describes the work of the protocols and the formal verification of the model. It outlines the data model and protocol flows used to register, authenticate, and access TPM-based registration resources. In addition, it offers mathematical validation to guarantee that the system follows the security invariants, including device integrity and compliance with the policy.

Chapter 6: This chapter deals with the results and analysis of system performance. It provides empirical information on the testing Response Time, Delay, and Throughput, between 300 and 1,500 operating cycles. Here, a comparative analysis of the research with state-of-the-art models is provided, and how the research objectives and requirements were achieved is discussed. In addition, the research findings are presented with respect to the drawbacks of current federated systems and the effectiveness of the proposed ZTA-FIS-AMFA combination.

Chapter 7: This chapter has the conclusion and a roadmap for future work on possible improvements, such as machine learning in risk assessment and production deployment.

Chapter 2

Literature Review

In this chapter, we present a comprehensive overview of the fundamental concepts and previous studies that influence this research. We hope to establish a sound theoretical foundation for the planned integration of Zero Trust Architecture (ZTA), Federated Identity, and Adaptive Multi-Factor Authentication (AMFA) into a combined access control framework.

Section 2.1 describes the key technologies and frameworks that are relevant for this research, e.g., ZTA, Federated Identity Systems, and authentication models. These building blocks constitute the backbone of reasoning for the subsequent chapter's design decisions.

Section 2.2 provides an overview of available academic and industry publications on federated identity, trust models, access control, and adaptive authentication. This includes comparative analyses, implementation strategies, and the identification of open issues in the field.

Finally, Section 2.3 presents the key conclusions of the literature that has been reviewed, recognizes gaps, and outlines how this study will address those gaps with a new, integrated approach.

2.1 Preliminaries

In this chapter, we first describe the relevant technologies used in this research. In the next section, we review the existing literature on the topic and provide a comprehensive analysis of that literature. We have also prepared a comparative table 6.2 on this in the next section.

2.1.1 Zero Trust Architecture (ZTA)

Zero Trust (ZT) is a security paradigm that assumes no implicit trust, regardless of network location, and continuously evaluates access requests based on multiple contextual factors. It is designed to reduce uncertainty in enforcing least-privilege access decisions on a per-request basis, especially in environments that are presumed to be compromised [5]. A Zero Trust Architecture (ZTA) operationalizes these principles across enterprise infrastructure and workflows, treating enterprise-owned

environments as no more trustworthy than external networks. The main aspects of ZTA are powerful identity verifications, health verifications of devices, granular access control, and constant monitoring. Instead of relying on the old-fashioned perimeter protection, ZTA uses logical segmentation, policy dynamism, and behavior analytics in reducing the blast radius of possible breaches [22]. Zero Trust enterprise incorporates these concepts into both physical and virtual infrastructures and notes that it is necessary to have adaptive and risk-aware access at all levels of the system [23].

2.1.1.1 ZTA Tenets

A zero trust architecture is designed and deployed with adherence to the following zero trust basic tenets[23];

- All data sources and computing services are considered resources.
- All communication is secured regardless of network location.
- Access to individual enterprise resources is granted on a per-session basis.
- Access to resources is determined by dynamic policy, including the observable state of client identity, application/service, and the requesting asset, and may include other behavioral and environmental attributes.
- The enterprise monitors and measures the integrity and security posture of all owned and associated assets.
- All resource authentication and authorization are dynamic and strictly enforced before access is allowed.
- The enterprise collects as much information as possible about the current state of assets, network infrastructure, and communications, and uses it to improve its security posture.

2.1.1.2 ZTA components

ZTA depends on several components to impose its security principle. Here are the components of ZTA:

Policy Engine (PE): The Decision Maker - It evaluates each access decision based on policy and other information sources, whether to grant, deny, or revoke.
Policy Administrator (PA): The Doer - It communicates with the PEP and generates an access token or credentials. Also, it tells the PEP to allow or disallow access.
Policy Enforcement Point (PEP): The Gatekeeper - It is a core component in Zero Trust Architecture that acts as the gatekeeper between users and protected resources.

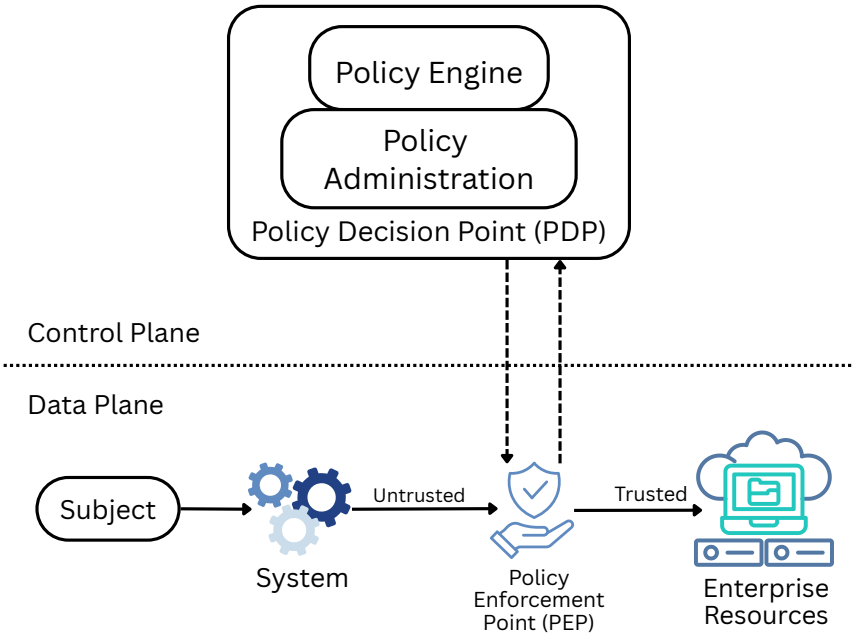


Figure 2.1: ZTA Components

At the core of Zero Trust Architecture (ZTA) lies a coordinated set of components that collectively enforce dynamic, least-privilege access decisions. The Policy Engine (PE) is the decision-making authority. As shown in Figure 2.1, PE evaluates access requests based on enterprise-defined policies, contextual information (such as user identity, device posture, and network conditions), and real-time threat intelligence [23]. After PE makes a decision, the Policy Administrator (PA) takes up his role. The PA, as the executor of the decision, pushes the decision forward by setting access controls and enabling the secure communication process between the requester and the resource [23]. At last, there is the Policy Enforcement Point (PEP), which is the gatekeeper that occupies the gate perimeter between the operators and secured resources. It makes sure that only authorized traffic that is verified by PE and approved through PA is allowed to pass through [22]. A combination of these elements forms a trust evaluation pipeline that will provide constant access evaluation and alignment with organizational security objectives.

2.1.2 Identity Models

Modern digital identity systems operate under several disparate models, each shaped by the evolving balance between security, usability, and control. In Siloed Identity models, users deal with numerous individual logins for each service, a frustrating experience all too common to all, which also increases security risks. Centralized models streamline this by funneling all the credentials through a single authority, but the convenience comes at a cost: one breach potentially compromises all. To enable simplicity of access across domains, Federated Identity models enable logging into

multiple systems with one trusted identity provider, think logging into third-party apps with your Google or Facebook account. Most recently, Decentralized Identity models have entered the scene, giving individuals management of their digital identity with cryptographically secure wallets, reducing reliance on central authorities. Many organizations pursue Hybrid Identity models, combining centralized and federated models to meet operational as well as security demands in transitions. The workflow of these models is shown in Figure 2.2 except the Federated Identity model, which has been shown in Figure 2.4. Every model represents a different philosophy about who should own, control, and trust digital identity in an interconnected world [17].

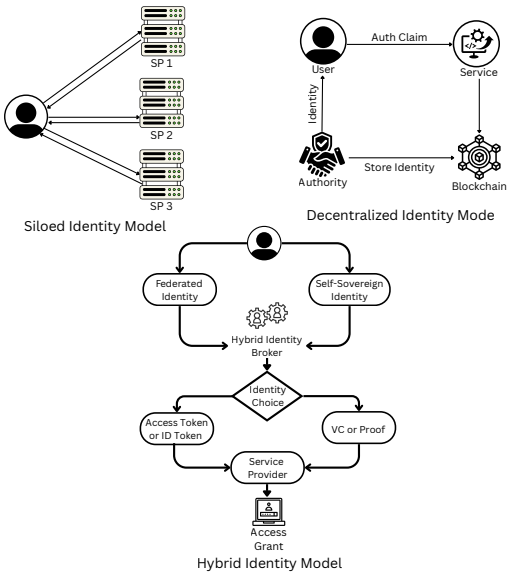


Figure 2.2: Different Identity Models

2.1.2.1 Federated Identity

Identity models define how user identities are handled, authenticated, and authorized within and across systems. The most prevalent ones are the centralized identity model with one central authority (like an internal directory or IdP) possessing all the identities, which is efficient but open to single points of failure. The federated identity model allows for home users from different organisations to access services based on home credentials using trust between identity providers (e.g., SAML or OpenID Connect). The decentralized or self-sovereign identity model puts control in users' hands, with digital identities managed by them through blockchain-based verifiable credentials, enhancing privacy and user control. The two models give different tradeoffs in scalability, control, and trust, and the choice is context-dependent.

A Federated Identity System allows users to authenticate across multiple independent domains using a single set of credentials, typically managed by a central Identity Provider (IdP) [14]. Instead of registering separately for each service, users authenticate once through the IdP, which then issues trusted identity assertions to various Service Providers (SPs). A Federated Identity System allows users to log in for

multiple independent domains with one set of credentials, normally managed by one central Identity Provider (IdP) [14]. Registration with all services is unnecessary: they log in through the IdP, which makes trusted identity claims to multiple Service Providers (SPs). Users and service providers are therefore exempt from the drudge of handling credentials, enhancing user experience and administrative ease of handling. A case is a university student who can log into an external digital library with campus credentials. The digital library, as SP trusts the university's IdP for verification of the user and his or her access rights. Schemes for implementation of this scheme include SAML 2.0, OpenID 2.0, and OpenID Connect (OIDC), which standardize the format and security of identity claims. There are a number of risks involved in this scheme. In case of breach of IdP, over-reliance on the IdP may result in system-wide compromise. Also, inadequate implementation can result in interception or misuse of tokens, and too many user attributes can cause privacy issues, including linkability and unintended profiling across services [2]. In addition, most federated systems do not verify after authentication and believe that a fixed session is valid.

2.1.2.1.1 Key Concepts of a Federated Identity System

A Federated Identity System is a system that comprises several basic elements that facilitate organizational operational security and scalable identity management:

Identity Provider (IdP): This is the authoritative body that performs user authentication and identity assertion, or token generation. Identity Providers that are well-known include both commercial (e.g., Google, Microsoft) and institutional systems (e.g., university services of logins) [1].

Service Provider (SP): A Service Provider is dependent on the IdP to confirm the user identity and grant access to secured resources. Examples include academic portals, cloud-based services, or content providers' credentials [8].

Authentication Protocols: Federated protocols are required to use a standard identity to ensure the secure identity transfer between the IdP and SP. The most widely used protocols are OpenID 2.0, OpenID Connect (OIDC), and Security Assertion Markup Language (SAML 2.0); they provide the steps for obtaining, granting, and verifying identity and access tokens [16].

Tokens: Tokens with cryptographic signatures, such as SAML assertions, access tokens, or ID tokens, are issued by the IdP upon successful authentication. These tokens encapsulate user claims and are presented to service providers (SPs) for authorization decisions. The token formats and scopes depend on the underlying authentication protocol [16].

In Figure 2.3, a visual representation of the relation between these components is given.

2.1.2.1.2 How It Works

In the connected digital world of today, people regularly engage with services that cut across several organisational boundaries. Managing separate credentials for each

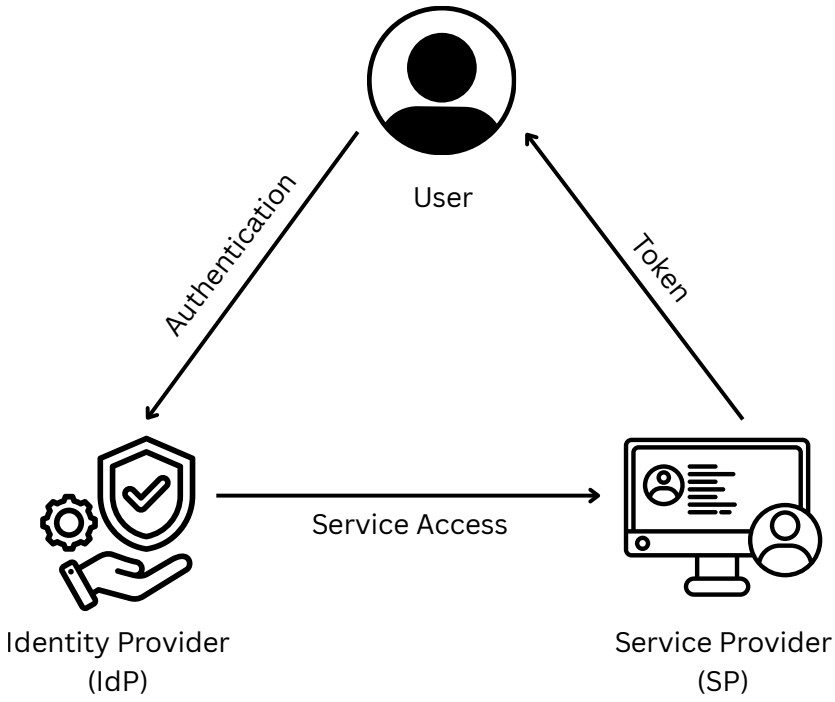


Figure 2.3: Relation Between IdP, SP, and User

service quickly becomes impractical and insecure. Federated Identity Systems address this by allowing users to authenticate once with a trusted provider - called an Identity Provider (IdP) and then access multiple independent services - called Service Providers (SPs)- without the need for separate logins [1]. In 2.4, the workflow is shown. This model improves user experience, besides enhancing security posture when it is coupled with contemporary protocols such as OpenID 2.0 and OpenID Connect. With these standards, organizations do not have to deal with sensitive password databases and transition to the trust-based authentication processes [8].

Figure 2.4: FIS Workflow

2.1.3 Authentication

Authentication refers to the process of establishing the identity of a user, a device, or a system and then granting them access to digital resources. It serves as the initial defense barrier in any secure environment and ensures that the persons with

allowed access are the only ones that can interact with the services or data under protection. It verifies an asserted identity using credentials, i.e., passwords, tokens, or biometrics. Authentication assists in avoiding unauthorized access in digital systems and creates trust [27].

Traditionally, authentication was based on static methods such as usernames and passwords, but these methods are becoming more and more vulnerable to cyber threats as the threats are becoming more and more sophisticated. In response to this, modern systems are moving toward dynamic, context-aware, and multi-layered authentication systems. This modern authentication system provides mechanisms for continuous authentication instead of verifying identity at the point of access. Efficient authentication methods maintain a balance between security, usability, and privacy.

2.1.3.1 Types of authentication

Based on the type of factor used to verify a user's identity, authentication is differentiated into different techniques [18],

Something the individual is aware of: This includes private information such as PINs, passwords, and security question answers. Only the user is expected to retain this information.

Something the individual owns: Think of this as a digital key that is portable. Smart cards, USB tokens, and even portable devices that generate one-time codes are a few examples.

An attribute of the person (static biometrics): This includes physical characteristics like fingerprints, face features, or retinal patterns that are difficult to alter.

The individual's actions (dynamic biometrics): Recognition by handwriting features, typing rhythm, and speech patterns are a few examples.

2.1.3.2 Adaptive Multi-Factor Authentication

Adaptive Multi-Factor Authentication is the process of dynamically modifying authentication requirements according to user behaviour, location, and device type. Adaptive MFA analyses real-time risk to select the most appropriate authentication processes that balance security and user experience. The system might provide access with minimum verification stages in low-risk situations, whereas high-risk situations may require the use of more authentication levels. This method reduces risks by asking for additional verification when abnormalities are found. Research has shown that adaptive MFA systems are capable of adapting to different situations based on requirements without affecting usability [11]. It often uses machine learning to improve the accuracy and adaptability of the risk assessment models.

2.2 Review of Existing Research

The secure extension of Enterprise Zero Trust (ZTE) frameworks to federate with external partners of different trust and competence is discussed by Simpson et al. [26]. They offer six federation models, ranging from ad hoc person-to-person sharing to full ZTE federation, and assess each one for scalability, security depth, and trust compliance. In order to support authentication, authorisation, confidentiality, and accountability, the paper makes a compelling case for security and cryptography using technologies like Public Key Infrastructure (PKI), Security Assertion Markup Language (SAML) tokens, Transport Layer Security (TLS) encryption, and Security Token Services (STS). The vulnerability and usability of each of the models are measured through a comparison methodology, which comprises an essential survey of the way the vulnerabilities escalate as the federation expands. The performance analysis, as well as practical implementation, is not included in the scope of this paper, yet it will be one of the main issues of our research. Nevertheless, it provides a practical strategic framework for developing secure federated systems based on Zero Trust principles.

In his article on Zero Trust in Cloud, Himanshu Sharma expounds on the implementation of the Zero Trust principles on cloud infrastructure [30]. He provides a comprehensive security analysis that contains risk assessments and quantitative values that indicate vulnerabilities, intrusion, and compliance failure after installation. Through comparing Zero Trust to traditional methods of perimeter-based protection, the research provides a comparative assessment framework and points to the advantages of identity-based policies, micro-segmentation, and continuous validation. Cryptographic analysis and performance evaluation, which stand as two of the most important components that will be discussed in our research, are not covered in this paper. Despite this paper providing helpful recommendations and addressing topical technologies such as IAM systems, SIEM, MFA, andcloud-native solutions (AWS, Azure, Google Cloud).

The article by Fernandez et al. provides an intelligent and systematic examination of the Zero Trust model according to the traditional concept of security and the security patterns [29]. The authors, instead of assuming that ZTA is what it is, deconstruct its elements and concepts, such as least privilege, complete mediation, and trust evaluation, and look at the way of how these concepts are applied to business structures, such as those of BeyondCorp by Google and NIST. The study introduces an initial Security Reference Architecture (SRA) and evaluates ZTA by using practical questions based on overhead, feasibility, and threat coverage. Notably, this research contains a conceptual analysis and identifies the flaws in the current implementations, in particular, in the domain of trust computation and governance, but fails to provide empirical performance and cryptographic metrics. It also advocates principle-based, application-focused security architecture and condemns the excessive use of such techniques as microsegmentation. The fact that the abstract security patterns are used as an evaluation and guidance tool in the development of ZTA makes the work to be unique. Although it is mostly theoretical, it offers valuable practical recommendations and enumerates a broad spectrum of possible research directions, which makes it a valuable addition to the list of contributions

to the academic and industry stakeholders who want to apply the concept of Zero Trust to build more secure systems.

Kerman et al. [22] go beyond Zero Trust as an idealized concept and offer an in-depth practical solution to deploying Zero Trust Architecture (ZTA) in real-world environments. Formal performance does not occur as part of their work, nor does cryptographic analysis, but rather provides practical deployment strategies based on commercially available technologies, including identity providers, policy enforcement elements, and monitoring systems. The article does not provide any comparison analysis of competing ZTA solutions but concentrates on complete system construction that will show interoperability, legacy system integration, and ongoing verification procedures. One of the fundamental issues is security, which is implemented by identity assurance and access control, and threat mitigation procedures. Despite such cryptographic tools as TLS and JWTs form part of the architecture; they do not get into detail. The key contribution is in the very implementation - the provision of technical plans, architectural mappings, and workable insights that organizations may evolve depending on their own Zero Trust maturity.

In his article, Kanaka Maheswara Rao Chennuri [32] investigates adaptive multi-factor authentication (MFA) systems, which vary in authentication demands. Applying dynamic risk evaluation to overcome the shortcomings of traditional authentication tools in the current cybersecurity environment. The author provides a comprehensive solution that involves behavioral biometrics, contextual risk analysis, and AI-based anomaly detection. The system architecture will entail continuous biometric verification, geospatial authentication, and device health monitoring. Supervised and unsupervised techniques of machine learning are used to identify user behaviour and adjust the level of authentication. Experiment results across different industries show high user satisfaction (up to 94%) and almost 99% reduction in account attacks. The study fills a significant gap by merging different authentication factors inside the adaptive framework. Improved detection rates, shorter authentication times, and increased system reliability are the most notable contributions of the paper. However, a crucial factor for AI-driven security systems is adversarial robustness against spoofing and model evasion attacks, which is not properly evaluated in this paper. The article proposes additional studies on privacy-sensitive AI models and quantum-resistant protocols.

Chadwick [3] examines Federated Identity Management (FIM) in detail in order to address the emerging problems of identity verification, among various systems. The paper explains how the federation of identity providers and service providers enables users to authenticate only once and access different services. Systems evaluation, such as that of Shibboleth, CardSpace, or Athens, is also among the important tools that pay attention to protocols such as SAML and mechanisms such as attribute release policies. All these systems will operate on the model of "7 Laws of Identity" as presented by Kim Cameron, which enhances user permission, limited disclosure, and interoperability. The writer shows the effectiveness of FIM in defeating the shortcomings of centralized systems. It provides scalability by means of pseudonymous authentication. The assurance level integration, support of multiple identity providers, and privacy fine control are some of the major contributions. Though

the paper covers the problem of interoperability and phishing, it fails to analyze performance trade-offs in real-life implementations statistically.
Ghazizadeh et al. [7] present an extensive analysis of security issues with federated identity management (FIM) in cloud computing. It mainly focuses on identity theft and the lack of credibility between entities. The paper analyzes protocols like SAML, OAuth, and OpenID and their flaws, such as XML signature wrapping and logical issues of Single Sign-On (SSO). The paper lacks real-world model deployment, but it analyzes previous research and points out trust issues in federated entities. Trusted Computing (TC) technologies are proposed by the authors to improve identity assurance while decreasing misuse by implementing hardware-based trust anchors. The vital role of platform-level trust in federated identity ecosystems is mentioned here, which was overlooked in the previous research. It represents TC as a mitigation mechanism for reducing phishing and identity misuse. The paper does not provide any detailed implementation technique of the proposed TC solution. The lack of standard trust frameworks and the slow adoption of Trusted Computing (TC) are the limitations provided in this. It is suggested that future research should be based on developing scalable, privacy-preserving identification models with real-world deployment strategies.

Dodanduwa and Kaluthanthri [24] explore the crucial trust relationships that underpin the secure functioning of OAuth 2.0 and OpenID Connect. Their work doesn't propose new protocols but carefully breaks down five key trust connections, such as between the resource owner and client, and between the client and authorization server, that are essential yet often implicit in practice. They also emphasize the fact that the main purpose of OAuth 2.0 is to support third-party access to secured resources, which is highly dependent on the trust assumptions that are not part of it. In comparison, OpenID Connect builds upon OAuth 2.0 with the addition of user authentication, rendering human user participation an obligatory step, complicating the task of trust management. The paper is useful in clarifying the role of these trust dependencies in defining security in federated identity systems and becomes a guide for use to researchers and implementers who are interested in gaining more insight about the human and organizational aspects of identity management.

Naik and Jenkins [17] give a brief, but very informative comparison of three key federated identity protocols, such as SAML, OAuth 2.0, and OpenID Connect, in terms of how to choose the most appropriate one according to the application requirements. The authors provide a framework of decision-making, rather than just the technical overview, based on actual deployment situations. They claim that the management of digital identity depends on the context, depending on the authentication vs. authorization requirements, enterprise-based system, cloud-based system, or mobile. SAML is a strong XML protocol that could serve enterprise-level SSO. OAuth 2.0 is known to be flexible in authorization processes, especially in API-based architecture, but it lacks user identity. OpenID Connect becomes the most contemporary and flexible, through the addition of authentication over OAuth 2.0 with standardized ID tokens. What is unique about the paper is that it provides practical guidelines on how to match the protocol selection with architectural and security requirements, as opposed to trendy ones. Although the study is abstract and does not provide any empirical standards, it is a beneficial one as it assists

system designers to make successful decisions that are context sensitive to adopt federated identity.

In reference to Zero Trust Architecture (ZTA), Teerakanok et al. [43] conduct an extensive review of the concept with a focus on how it is moving beyond its model of perimeter-based security to a model where trust is not assumed, and all access requests are dynamically checked. Their analysis cites major issues in migrating to ZTA, such as integration with legacy systems, complexity in the architecture, and organizational resistance. It is also interesting to note that the authors recommend a gradual adoption approach, together with alignment of the stakeholders to facilitate their adoption. This observation justifies the improvement of federated identity systems directly, in which reliance on Identity Providers (IDPs) is excessive, misuse of a token, and static validation of a session are severe problems. Federated systems can have a stronger posture of security by incorporating the principles of ZTA, including continuous authentication and least-privilege access. The paper provides a critical base upon which the study of how ZTA can address the fundamental weaknesses in identity federation, particularly those associated with visibility of authentication flows and policy enforcement, will be conducted.

Rahman et al. [35] discuss how the concept of Zero Trust Security (ZTS) can be implemented in the enterprise architecture of Micro, Small, and Medium Enterprises (MSMEs), where the main focus is to enhance cybersecurity in those settings with scarce resources and expertise. The research approach is based on an enterprise-architecture-oriented approach that incorporates the analysis of stakeholders, design of both application and technology architecture, incremental implementation, and post-implement evaluation. ZTS is realized in architectural controls, including multi-factor authentication (MFA), micro-segmentation, intrusion detection and prevention (IDPS), and encrypted communication, and is focused on an organizational implementation, not protocol-level design. The presented result shows that the number of reported security incidents was significantly lower after the adoption of ZTS. Overall, the number of incidents dropped to 53, as opposed to the 105 incidents before adopting ZTS, which represents about 45-50 percent of the decrease and an increase in employee cybersecurity awareness. The authors also determine critical implementation issues such as budgetary restrictions, training needs, and cultural adjustment, and suggest the gradual implementation and affordable security plans adapted to MSMEs.

Hasan [33] discusses the Zero Trust Architecture (ZTA) as an all-encompassing enterprise-security framework that seeks to curb vulnerabilities and insider attacks by continuously providing the least possible access. In the paper, a systematic ZTA implementation framework is provided, focusing on identity and access management (IAM), MFA, micro-segmentation, and continuous monitoring and behavioral analytics. Instead of suggesting a specific implementation of the system, the research summarizes the results of the industry reports and real-life case studies to assess ZTA effectiveness. The positive results reported are as follows: phishing-related breaches were reduced by 40 percent, the lateral movement was decreased by 35 percent, and insider-threat incidents dropped significantly in the financial sector, healthcare sector, and technology sector after adopting ZTA. Such findings can support the

effectiveness of ZTA in minimizing the attack surfaces and enhancing enterprise security posture, and the difficulties in integration complexity, scalability, and cost of deployment are also mentioned in the paper, especially in the case of smaller organizations.

Lewis and Lewis [4] focus on how Security Assertion Markup Language (SAML) is used to provide web-based SSO, emphasizing how it helps to strengthen federated authentication procedures and lessen credential fatigue. Versions 1.0 and 2.0 of SAML, a mature standard created by OASIS, have improved support for browser-based SSO, allowed for single logouts, and strengthened cryptographic safeguards like encryption and digital signatures. Due to the modularity of SAML, assertions, bindings, and identities can be used by developers to tailor authentication processes, as the article brings this out. Due to metadata sharing, Identity Providers (IdPs) and Service Providers (SPs) can cooperate without making mistakes. It is established as well that the security capabilities of SAML, including the assertions which are time-bound and strong cryptography enforcement, reduce the security threats such as replay and man-in-the-middle attacks. The paper also cites more recent works that attempt to make SAML more resilient, as dual artifact systems and secure binding increase. In general, the research makes SAML a cornerstone of federated identity systems, which is easy to use and secure with scalable solutions to the current web authentication requirements.

2.3 Summary of Key Findings

The reviewed literature indicates that there is an increasing attention to the use of Zero Trust principles to federated identity systems, and most of the studies are sound theoretical frameworks, computer applications, and practice. As the comparison is evident in Table 6.2, however, there are still big gaps in many dimensions. In particular, there are not many studies that involve the analysis of performance in detail, although Zero Trust affects the responsiveness and scalability of the system to a considerable extent. These studies also lack hardware security, protocol flow and threat modeling. Besides, several studies mention tools and technologies (e.g., IAM systems or cloud-native platforms) as usable, but few of them offer end-to-end systems implementations, of course, not in federated ones. Comparative evaluation frameworks are used sporadically, and empirical evidence is largely absent, which limits the capability to benchmark proposed models. This research addresses these gaps by suggesting the design and implementation of a Zero Trust-based federated identity system, analyzing it across security, performance, and cryptographic resilience dimensions, and providing a structured comparison against state-of-the-art models to determine its effectiveness and practical viability.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

The given section determines the specification of the system by identifying the functional and security requirements and inherent constraints of the suggested solution. It also presents the compliance requirements, industry standards, and codes of practice that will be used in designing and implementing the system. Together, they make the proposed solution secure, reliable, scalable, and consistent with the Zero Trust principles and address the operational and regulatory expectations as well.

3.1.1 Threat Modeling

Through our identity federation system, threat modeling avoids security vulnerabilities from happening due to SP-to-IdP trust relationships. To enable us to enumerate the potential threats in our identity federation system, we utilize Microsoft's well-trusted STRIDE threat modeling framework [42], which defines security vulnerabilities into six categories of threat relevant for such systems.

T1 - Spoofing: The attacker can masquerade as a valid user, identity provider (IdP), or service provider (SP) with the goal of illicit access or deceiving users in federation identity systems.

T2 -Tempering: Tempering with authentication tokens (e.g., SAML assertion, OAuth token), with metadata, or with the SP and the IdP exchanged messages, with the possibility of evading validation tests.

T3 - Repudiation: Since federated systems often have centralized authentication at the IdP, users or entities may deny having performed specific actions if proper logging and traceability aren't enforced.

T4 - Information Disclosure: Sensitive information such as user properties or token information may be disclosed negligently through improper configuration, token exposure, or verbose protocol metadata in the federation protocols.

T5 - Denial of Service: The Denial of Service attack on the IdP will deny all legitimate users from authenticating against all SPs that are connected, thereby rendering the federation's access disabled.

T6 - Elevation of Privilege: If SPs fail to properly validate claims or access scopes, attackers can manipulate tokens to gain higher-level privileges than assigned by the IdP.

Apart from STRIDE, there are some additional threats in a Federated Identity system,

T7 - Insecure Metadata Configuration: Federated systems rely on metadata (e.g., entity IDs, endpoints, certificates), and insecure or unsigned metadata can be tampered with to redirect authentication flows or disable verification mechanisms. Attackers can exploit this to introduce rogue IdPs or SPs into the trust fabric.

T8 - Weak Token Binding: If tokens are not cryptographically bound to the device or session, they can be stolen and reused elsewhere. This breaks session context and violates Zero Trust principles of identity-asset binding.

A federated identity system exchanges a range of user traits and authentication information across trust boundaries because it incorporates several interacting entities, such as Identity Providers (IdPs), Service Providers (SPs), and individuals. This data flow presents a number of possible privacy risks in the absence of sufficient privacy controls or user permission procedures. Based on this understanding, the identified privacy threats are as follows:

T9 - Unintended Attribute Exposure: Federated identity systems often transmit user attributes (e.g., email, name, group) from the IdP to the SP. If not properly minimized or scoped, unnecessary personal information may be disclosed.

T10 - Linkability Across Services: When the same unique identifier (like a subject ID or email) is reused across multiple SPs, it allows those services or attackers to correlate user activity across platforms.

3.1.2 System Requirement

Before even designing a system, understanding and identifying the requirements are important. Through requirement analysis, we would be able to define the functional and security controls that are important to build a proper secure system. As we are adapting the ZTA principles, the requirements are designed according to that to ensure a strong defense against threats that we are trying to mitigate.

Functional Requirements:

Functional requirement describes the essential functions and features that a system requires to build safely.

F1 - Federated Identity Authentication: The system should authenticate users through trusted Identity Providers (IdPs) using OpenID 2.0 or OpenID Connect.

F2 - Per-Session Authorization: Each request to a service has to be authorized individually without any reliance on session persistence or SSO.

F3 - Adaptive Multi-Factor Authentication (MFA): The system needs to impose MFA dynamically based on risk context (e.g., device, location, behavior anomalies).

F4 - Context-Aware Policy Evaluation: Access control decisions should consider identity, device security posture, environment, and behavioral signals.

F5 - Attribute Disclosure Control: The system needs SP-specific policies and user consent to allow user attributes.

Security Requirements:

Security requirement explains the system's defence for the data and resources in terms of potential threats.

S1 - Secure Communication Channels: All interactions between users, IdPs, and SPs must be protected using strong encryption (e.g., TLS 1.3). They help to secure against message tampering during transit (T2), exposure of confidential information (T4), and diversion attacks from insecure metadata settings (T8).

S2 - Dynamic Access Policies: Access must be granted or denied in real-time based on dynamic policy evaluation, not static roles or network zones. These reduce the likelihood of privilege escalation (T6), impersonation by unauthenticated devices or entities (T1), and information leakage by extending access based on currently existing risk environments(T4).

S3 - Token Security and Binding: Tokens must be encrypted, signed, and bound to the requesting device/session to prevent theft or replay attacks. This prevents those who reuse and steal tokens (T9), ensures that token tampering is identified (T2), blocks spoofed identities (T1), and reduces the threat of unauthorized privilege escalation (T6).

S4 - Minimal Trust and Continuous Validation: No entity (even within the network) is trusted by default; all identities and devices must be continuously verified. The model protects against impersonation attacks (T1), counters the impact of Denial of Service attacks by minimizing the trust dependencies (T5), and reduces the likelihood of unauthorized privilege gain (T6) or token misuse (T9).

S5 - Monitoring and Audit Logging: The system must log all access and identity transactions to enable behavioral analysis and threat detection. This facilitates traceability and non-repudiation (T3), can recognize DoS or suspicious patterns of

activity early (T5), assists in the discovery of privilege misuse (T6), and facilitates recognition of tracking threats across services (T8).

3.2 Societal Impact

The proposed federated identity system with hardware attestation, which is built on the principles of Zero Trust and adaptive MFA, can improve the security of the enterprise and inter-organizational settings by lowering identity theft, malicious usage of tokens, and unauthorized access to the resources. This helps to secure user identities, sensitive enterprise data, and valuable assets of the organization.
In organizational terms, the research facilitates responsible digital transformation by facilitating safe cooperation between independent actors without using informal trust. It is especially important in industries where data transfer among institutions is relevant, including healthcare networks and financial systems, and a data breach may lead to disastrous impacts on society. The system seeks to enhance resilience and accountability in digital ecosystems by improving trust relationships by consistently verifying them instead of making assumptions that may be constant.
In general, the study will help to develop more reliable digital identity infrastructures, which will give users confidence, safeguard individual rights, and help them engage in digital activities safely in an already more interconnected society.

3.3 Environmental Impact

The proposed system has a low environmental impact and is environmentally friendly as it is mainly software-based, and it uses enterprise infrastructure. It uses existing hardware features like tpm so it avoids extra implementation of hardware and reduces electronic waste. A continuous verification system ensures efficient use of computational resources and reduces unnecessary energy consumption. Also, federated identity management promotes more efficient resource management and minimizes the redundancy of infrastructure. So, overall, the proposed system promotes environmentally sustainable security practices.

3.4 Ethical Issues

This research does not involve personal data collection. As no interviews or user-provided statements were taken during the design of the system, it raises no ethical issues. The work is done by following standard academic and engineering practices.

3.5 Constraints, standards, and codes of practice

Several limitations define the proposed system, both in its design and possible implementation. The availability of a TPM-enabled device limits hardware attestation and, thus, cannot operate when it is not available. The prototype has been limited to the controlled testing environment and does not include the complexities involved in large-scale production deployment. Additionally, the issue of resource limitations,

which most importantly is the computational cost of the ongoing authentication process and cryptographic operations, has been factored into the analysis and is assumed to be within reasonable performance limits in the current research.

The design of the system is aligned with the NIST Special Publication 800-207 [23] on Zero Trust Architecture that outlines fundamental principles such as continuous authentication, least privilege, and policy-based access control. The federated authentication mechanisms are based on the protocols specified by OpenID Connect and OAuth 2.0, so the protocols are both interoperable and secure. Every cryptographic activity complies with best-practice recommendations by NIST and OWASP [41], with the use of sound encryption, key management, and token security measures. The general data-protection principles, such as data minimisation and data limitation of purpose, direct privacy, and are also aligned with the General Data Protection Regulation (GDPR) [12], but a full regulatory compliance evaluation falls outside the scope of this research.

3.6 Project Management Plan

This section describes the project management strategy used to warrant the successful implementation of the proposed Zero Trust-based Federated Identity System within the planned time. The project management plan specifies the resources needed, the schedule of development, the budget, and the creation of the progress monitoring mechanisms and plan adjustment in case of need.

3.6.1 Resource Allocation

The project mainly involves the use of technical software resources. Technical resources consist of a development workstation that can run a containerized environment, virtual machines, and cryptographic operations. The devices registered through TPM must have TPM-capable devices or software that mimics TPM functionality. The software resources consist of open-source software like identity management platforms (e.g., OpenID Connect-compatible IdPs), policy engines, containers, and security testing tools. There is no need to have proprietary software licenses, so financial overhead is reduced. Documentation and reporting tools are necessary to help with the constant progress tracking and thesis preparation.

3.6.2 Project Timeline

It is a progressive way of undertaking the project so that it can be implemented systematically and on time. For this, we have divided the whole project into three parts. The first stages involve background research, modeling threats, and requirements. The next stages are system design, federated authentication, integration of TPM-based device attestation, and adaptive mechanisms of MFA. The last stages include testing of the system, performance analysis, documentation, and finalization of reports.

The progress is checked regularly, and the schedule can be modified in case of technical difficulties or unforeseen delays. Such a step-by-step approach will allow meeting

the key milestones without lowering the quality of the system and research goals.

3.7 Risk Management

The aspect of risk management is essential to the project because federated identity systems are sensitive to security, and the implementation of Zero Trust is technically complex. In this section, possible risks in the project are identified, the effects are evaluated, and mitigation measures are described.

3.7.1 Project Risk Identification

Various risks can bear on the effective implementation of the project. There are technical difficulties in the integration of federated identity protocols, adaptive MFA mechanisms, and TPM supporting attestation. System responsiveness can also be affected by performance overhead due to constant verification and cryptographic processes. Also, real-world validation may be limited by access to physical TPM-enabled equipment.

In terms of project management, time is a risk factor in case project implementation or debugging is more labor-intensive than expected. Hardware attestation or policy enforcement mechanisms may cause further development delays because of knowledge gaps.

3.7.2 Monitoring and Reduction Plans

As a means of reducing technical risks, the system is developed in stages, with each part being tested individually, and only after that, the entire system is integrated. Lightweight policy appraisal and selective use of continuous verification are the responses to the risks of performance.

The time-related risks are taken care of by the introduction of buffer periods to the project schedule and emphasis laid on the core functionality of the system instead of optional additions. Academic supervision at regular progress reviews assists in the identification of problems at an early stage and in the determination of corrective actions. The use of documentation and version control helps even more to lessen the chances of rework or mistakes in the implementation.

The risks are constantly observed during the project lifetime. When a risk occurs, then corrective actions through rescheduling of tasks, changing the scope, or implementing different strategies are undertaken. This proactive risk management strategy would make sure that the project is on schedule and will be accomplished within the required time without jeopardizing security goals and the quality of research.

3.8 Economic Analysis

The proposed system is economical because resource management and use of the existing infrastructure are efficient. It uses integrated Trusted Platform Modules

(TPMs) which are available in modern devices. The use of integrated TPMs helps to reduce the cost of additional hardware. The use of an HSM simulator instead of physical HSM devices helps to reduce costs during development and evaluation.

The reduction of unnecessary overhead of authentication is done through MFA by the invocation of dynamic higher security controls in scenarios relating to high risk. The solution provided has a tradeoff between efficiency in costs for operations and enhanced security.

Chapter 4

System Methodology and Design Specification

This chapter outlines the structured approach taken to create and evaluate a secure federated identity system incorporating Zero Trust Architecture (ZTA) and Adaptive Multi-Factor Authentication (AMFA). The research approach seeks to incrementally address the research objectives through the combination of theoretical research and practical development. Each stage, from problem identification to system evaluation, paves the way towards the development of a scalable, privacy-preserving, and context-aware authentication system in federated settings.

Section 4.1 describes the research methodology, identifying the steps taken throughout the research. The steps are for the purpose of developing the proposed system while keeping it coherent with the research aims.

Section 4.2 provides a technical overview of the system specification, defining the architecture, modules, components, and technologies utilized to implement the proposed Zero Trust-enabled federated identity framework.

4.1 Research Methodology

This research follows a systematic methodology to design, implement, and evaluate a secure federated identity system aligned with Zero Trust principles. The methodology comprises the following stages:

Problem Identification: Identified the limitations of traditional federated identity systems, such as static trust, token misuse, and weak privacy controls.

Literature Review: Reviewed existing work on federated identity protocols, Zero Trust Architecture, and known security/privacy vulnerabilities.

Research Objective: Aimed to design a Zero Trust-based federated identity system that enhances security and privacy without relying on SSO.

Threat Modeling: Applied the STRIDE model to identify and classify potential security and privacy threats in the proposed system.

Requirement Analysis: Defined functional and security requirements guided by Zero Trust principles and threat modeling insights.

Architecture Design: Designed a system using OpenID/OIDC, adaptive MFA, and dynamic policy evaluation to enforce zero-trust-based access control.

Protocol Design: Defined secure communication rules and adaptive authentication flows aligned with Zero Trust principles within the federated identity framework.

Implementation: Developed the system integrating federated protocols with Zero Trust components like risk-aware MFA and continuous validation.

Testing and Evaluation: Evaluated the system's resilience to threats and its ability to enforce fine-grained, privacy-preserving authentication flows.

In the Figure 4.1, a visual representation of methodology is given.

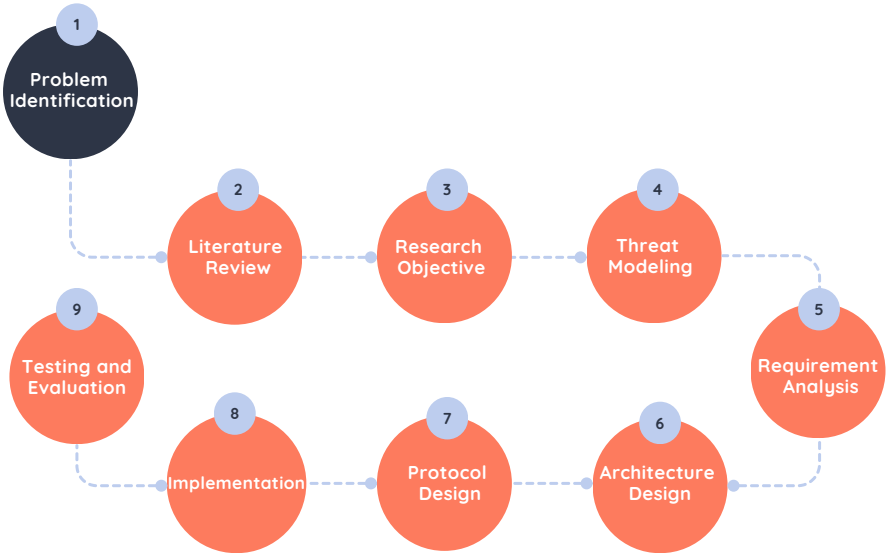


Figure 4.1: Methodology

4.2 System Design

The proposed system is build with several prominent components that play specific security and functional roles. The internal mechanisms enforce continuous, hardware-backed device attestation, adaptive multi-factor authentication, and dy-

namic, policy-driven access control aligned with Zero Trust principles, while the decision controller centrally manages all access decisions. Finally, the design clearly separates the architecture into two ends, the user end, which consists of the client device and browser that carry out cryptographic operations as well as attestations, and the system end, comprising backend services that handle the identity verification, device trust, policy checking, and resource allocation, thereby making the federated identity system scalable, secure, and modular and performing adaptive as well as continuous validation of trust.

4.2.1 Architecture

The proposed architecture is shown in Figure 4.2. It combines federated identity protocols, attestation service, and Zero Trust Architecture (ZTA) principles into a single framework. It includes two principal areas: the User End and the System End. The User End includes the user device and the trusted hardware; on the other hand, the System End handles the authentication, authorization, and policy implementation.

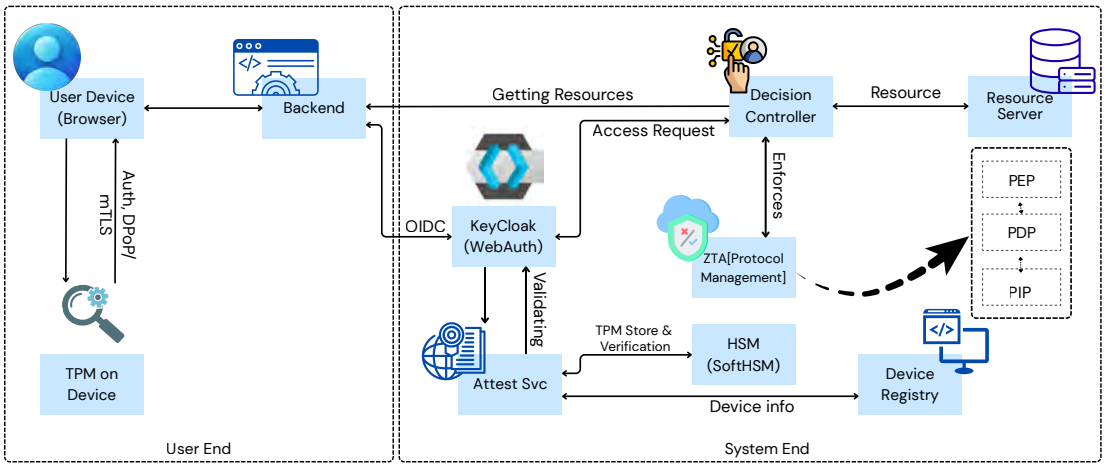


Figure 4.2: System Architecture

User Device

The User Device works as the client side from which the access request is made. A browser or client application will run on a device that has a Trusted Platform Module (TPM). The TPM creates evidence of attestation in order to identify the integrity of the device before the exchange of identities. Mutual TLS (mTLS) is used to provide protection to communication to ensure that the data sent is confidential and authentic [40]. Moreover, Demonstration of Proof-of-Possession (DPoP) is used to prevent the replay attack of a credential through binding the tokens to a particular client key [31]

Identity and Trust Services

Authentication, device trust evaluation, and access policy enforcement are provided. Keycloak is an open-source identity and access management platform that provides OpenID Connect (OIDC) and WebAuthn interfaces that fulfill the roles of the Identity Provider (IdP). Keycloak authenticates the user, provides the security tokens, and allows the integration of federated identity protocols.

Attestation Service is used to determine device trustworthiness by comparing the measurements provided by TPM with a reference configuration in a Device Registry. Moreover, this helps to make sure that only the valid devices are involved in the authentication process.

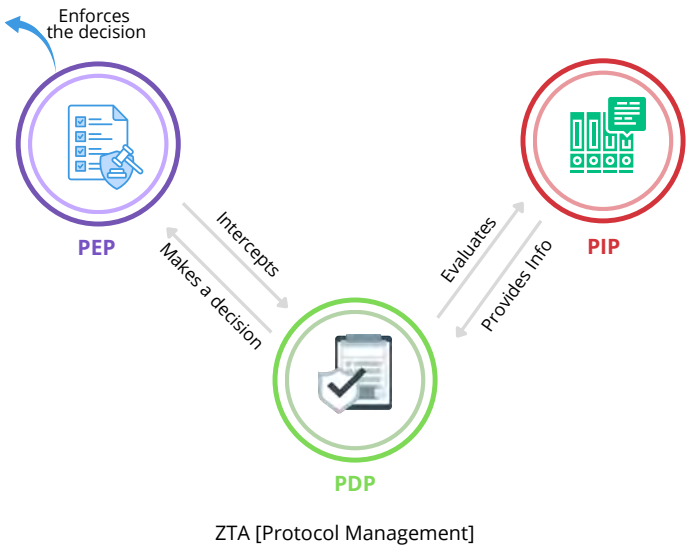


Figure 4.3: ZTA [Protocol Management]

The decision controller mediates access requests and communicates with the Zero Trust Policy Engine shown in Figure 4.3. It is a canonical Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Information Point (PIP) engine [6]. The PEP accepts incoming requests, the PDP compares them with the policies, and the PIP retrieves other contextual features of the user role [21]. This design introduces advanced adaptive authorization in the context of Zero Trust by combining enforcement, decision making, and information retrieval [19].

Resource Server Environment

The protected resources are stored in the Resource Server. The control of access to these resources is external and is provided upon authorization by the Decision Controller if the protocol management validates the request and permits the decision controller. Otherwise, resources won't be accessible. This continuous verification process helps to prevent unauthorized access to resources.

4.2.2 Workflow

The operation starts with the registration of a device when a user attempts to access the system, in this case, Figure 4.2 for resources. A new device produces TPM-based cryptographic keys and attestation evidence. After that, this evidence is forwarded to Keycloak, which then passes it to the Attestation Service. After that, the Attestation Service verifies this evidence. After successful verification, a new record of a device is written in the Device Registry, and a device-specific credential is made available for safe use. An HSM simulator (SoftHSM) was also used to store and manipulate the keys generated by the TPM, rather than physical hardware, to enable development and evaluation to be cost-effective.

In the authentication process, when the user tries to log in, they are redirected to Keycloak using an OIDC request. The WebAuthn challenge is issued by Keycloak and signed by the TPM. It is sent back as an assertion. Following this, the Attestation Service verifies the integrity of the device. If the verification is successful, Keycloak issues short-lived tokens tied to the device (DPoP-bound tokens) that provide proof-of-possession.

Access to resources is verified when the user requests access and the request is passed through the decision controller to the Policy Enforcement Point (PEP) of protocol management. It passes the token and DPoP proof to the Policy Decision Point (PDP). As the process continues, the request is assessed by the PDP, which is the Decision Controller. It evaluates based on the information gathered by the different Policy Information Points (PIPs), such as the Device Registry, Attestation Service, and Keycloak for token analysis.

Depending on this contextual knowledge- level of trust of a device, validity of the token, the PDP makes a decision (permit, deny, or conditional). This decision is then imposed by the PEP to the decision controller. In succession, the resources will be accessible by the decision controller if it gets validation from the protocol management. This endless loop of validation, decision-making, and enforcement represents the Zero Trust principles and ensures that no implicit trust is provided in any of the stages.

This architecture integrates hardware-enforced assurance of the device within a zero-trust model. Federated identity helps to simplify the authentication process between organizational domains with the help of token exchange [19]. Moreover Implementation of Keycloak as the IdP offers a standards-based federated authentication and authorization platform [21]. As well as ZTA, brings continuous verification and contextual access control to fight against modern threats. Overall, this combination of steps can provide a high level of security and maintain its usability, which is the foundation of the implementation and assessment in the following parts.

Chapter 5

Protocol Flow and Development

Here, we demonstrated the use cases for our suggested architecture together with their corresponding protocol flows. First, we identified every cryptographic notation that was used during the protocol flow in Table 5.1.

Notation	**Description**
D	User Device (with TPM)
TPM	Trusted Platform Module on the user device
TPM_{K_E}	Endorsement Key of TPM
TPM_{K_A}	Attestation Key of TPM
P_R	Platform Configuration Registers
M_{Data}	Metadata
N	Nonce
R	Resource
A_{SEV_R}	Attestation Service Report
$A_{SEV_{PL}}$	Attestation Service Payload
A_{SEV_S}	Attestation Service Status
D_R	Device Registry
D_{R_S}	Device Registry Status (e.g., active, suspended, revoked, pending)
$Dinfo$	Device information
$V_{(\cdot)}$	Token status (e.g., valid, invalid, expired)
DEC_{AC}	Access decision (e.g., permit, deny, conditional)
SA_{Ind}	Indicator of whether extra authentication is required
A	Admin of the system
U	End User
ID_U	User ID
ID_D	Device ID
UR	User role descriptor
ID_C	Credential ID
ID_R	Resource ID
R_{typ}	Resource type
RP_{IR}	Relaying Party Identifier
T	Timestamp
ID_P	Policy ID

Continued on next page

Notation	Description
IdP	Identity Provider for a system
T_{OIDC}	OIDC-issued token
T_{DPoP}	Demonstrating Proof of Possession token
$[\cdot]_{K\|HTTPS}$	Communication encrypted with key K or via HTTPS
K_U	Public key
K_U^{-1}	Private key
$\{\}_{K_U}$	Encryption using public key
$[\cdot]_{K_U^{-1}}$	Signature using private key
URL	Web Uniform Resource Locator
URL_R	Resource URL
RSN	Reason
N_{Info}	Network information
$(\cdot)_L$	Trust level (e.g., high, medium, low)
$D(\cdot)_L$	Device trust level
$C_{(\cdot)}$	Compliance level (e.g., yes, no, unknown)
$S(\cdot)_L$	Sensitivity level
$CORS$	Whether request originates from a cross-origin
DEC_{Res}	Reason for PDP access decision
ST_o	Opaque state value for CSRF prevention
Sc	Scope (permissions or access requested by user)
$Flag$	Boolean flag (yes/no indicator)

Table 5.1: Notations

5.1 Data Model

Table 5.2 demonstrates the data model formed for the proposal of the framework. The data model describes how requests (req) and responses (resp) operate in an authentication and authorization system, by TPM, OIDC, and Attestation services. Many requests and the associated responses can be handled by the framework. The requests are registerDeviceReq, attestDeviceReq, credentialReq, oidcAuthReq, dpopProofReq, policyEvalReq, resourceAccessReq, registryCheckReq, and accessReq, and the responses are registerDeviceRes, attestDeviceRes, credentialRes, oidcAuthRes, dpopProofRes, policyEvalRes, resourceAccessRes, registryCheckRes, and accessRes. Every request is responsible for the specific responses and verifies the essential according to the needs.

To register for the service, request the registerDeviceReq contains D_{info} (Device info) along with TPM_{K_E}(Endorsement Key of TPM) and the TPM_{K_A} (Attestation Key of TPM). Then the system responds by registerDeviceRes along with ID_D (Device ID) and D_{R_S} (Device registry status). Thus, finish the user registration procedures.

To authenticate the system, a request by the oidcAuthReq that holds the ID_U (user ID) and ID_D (device ID) along with the authParams. On the other side for response, oidcAuthRes gives T_{OIDC} (OIDC provided token) along with customClaims. The

system is being secured with the OpenID Connect-based authentication and proof validation.

The Resource Access Model provides a secure and policy-based resource fetching. The user requests are authenticated by KeyCloak and authorized by the Decision Controller with the assistance of demonstrating a T_{DPoP} (Proof of Possession Token). If approved, the Decision Controller retrieves the resource from the Resource Server. This solution ensures that the centralized authentication system is well-authorized and provides controlled access to protected resources.

Data Model
req($deviceRegReq, attestDeviceReq, credentialReq, oidcAuthReq,$ $dpopProofReq, policyEvalReq, resourceAccessReq, registryCheckReq, accessReq$)
resp($deviceRegRes, attestDeviceRes, credentialRes, oidcAuthRes,$ $dpopProofRes, policyEvalRes, resourceAccessRes, registryCheckRes, accessRes$)
deviceRegReq($D_{info}, TPM_{KA}, TPM_{KE}$)
deviceRegRes(D_{id}, DRS)
attestDeviceReq($ID_D, ASEV_{PL}$)
attestDeviceRes($attestStatus, ASEV_R$)
attestPayload($ID_D, TPM_{KE}, TPM_{KA}, PR, MData, [\ldots]_{K_U^{-1}}$)
attestStatus($V(\ldots)$)
credentialReq(ID_U, ID_D, N)
credentialRes($ID_U, ID_D, clientData, authenticatorData, [web]_{K_U^{-1}}$)
clientData($URL, N, type, CORS$)
authenticatorData($RPIR, [web]_{K_U^{-1}C}, Flag$)
oidcAuthReq($ID_U, ID_D, authParams$)
oidcAuthRes($T_{OIDC}, customClaims$)
authParams($ID_C, N, STo, URL_R, Sc, UR$)
customClaims($ID_D, DRS, D(\ldots)_L, UR, C(\ldots), SA_{Ind}$)
dpopProofReq($T_{DPoP}, [DPoP]_{K_U}$)
dpopProofRes($T_{DPoP}, proofStatus$)
proofStatus($V(\ldots)$)
AI(r, w, d, ex, A)
resourceAccessReq($ID_U, ID_D, ID_R, ai\ AI$)
resourceAccessRes($accessResult, auditInfo$)
auditInfo($ID_U, ID_D, ID_R, T, DEC_{AC}, ID_P, RSN, NInfo$)
policyEvalReq($ID_U, ID_D, resourceInfo, T_{OIDC}$)
policyEvalRes($DEC_{AC}, DEC_{Res}, SA_{Ind}$)
resourceInfo($ID_R, R_{typ}, URL_R, S(\ldots)_L$)
accessReq($T_{OIDC}, resourceInfo, attestStatus$)
accessRes($DEC_{Res}, conditions, ID_R, R$)
conditions("read-only","valid for 5 min","re-auth required")
registryCheckReq(ID_U, ID_D)

Continued on next page

Data Model
registryCheckRes(DRS, ti_{TL}, ci_{CS})
TL(H_L, M_L, L_L)
CS(Cy, CN, Cun)

Table 5.2: Data Model

5.2 TPM-based Federated Identity Registration Protocol

In this phase, device registration is managed through an administrator, initiating a deviceRegReq from a user device. The TPM of the device creates its attestation keys and produces a signed attestation statement. The attestation is sent to the attestation service, authenticating device integrity and provenance. When the attestation is successfully authenticated, the Device Registry flags the device as active and enrolls it for further authentication and resource access. Lastly, Keycloak(K_C) produces credentialRes, which is securely sent back to the TPM and stored. This distributed approach ensures device enrollment and only admits hardware-backed devices into the system.

A possible protocol flow of this phase can also be viewed from Figure 5.1 and such an exchange of messages results from Table 5.3 We detail the possible protocol flow as follows.

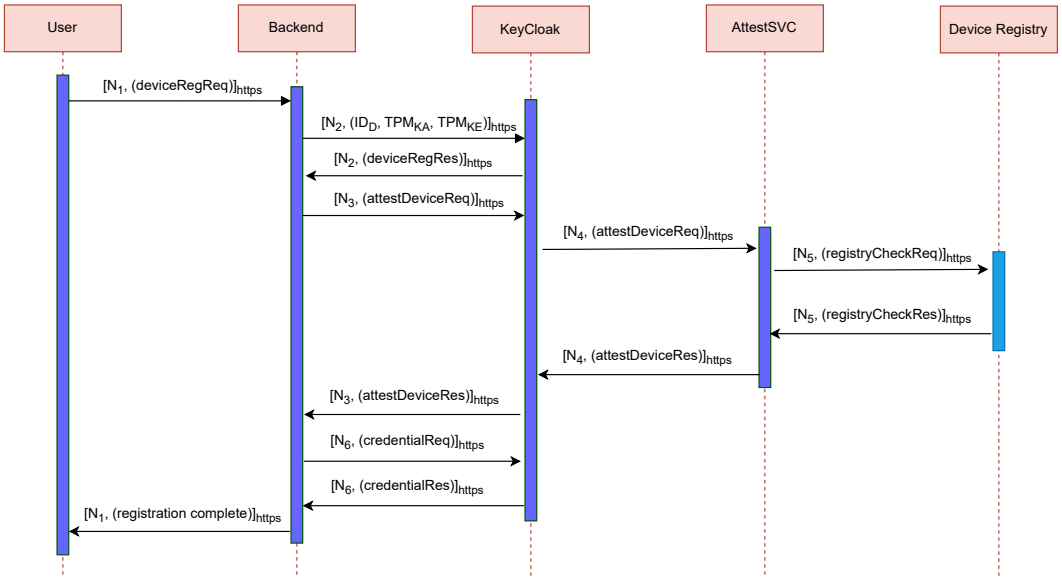


Figure 5.1: Registration

M	Description
M1	$U \;\; B_{end} : [N1, (deviceRegReq)]_{HTTPS}$
M2	$B_{end} \;\; K_C : [N2, (ID_D, TPM_{K_A}, TPM_{K_E})]_{HTTPS}$
M3	$K_C \;\; B_{end} : [N2, (deviceRegRes)]_{HTTPS}$
M4	$B_{end} \;\; K_C : [N3, (attestDeviceReq)]_{HTTPS}$
M5	$K_C \;\; A_{SEV} : [N4, (attestDeviceReq)]_{HTTPS}$
M6	$A_{SEV} \;\; D_R : [N5, (registryCheckReq)]_{HTTPS}$
M7	$A_{SEV} \;\; D_R : [N5, (registryCheckRes)]_{HTTPS}$
M8	$A_{SEV} \;\; K_C : [N4, (attestDeviceRes)]_{HTTPS}$
M9	$K_C \;\; B_{end} : [N3, (attestDeviceRes)]_{HTTPS}$
M10	$B_{end} \;\; K_C : [N6, (credentialReq)]_{HTTPS}$
M11	$K_C \;\; B_{end} : [N6, (credentialRes)]_{HTTPS}$
M12	$B_{end} \;\; U : [N1, (registration\ complete)]_{HTTPS}$

Table 5.3: TPM-based Federated Identity Registration Protocol

i. A user(U), aiming to register a new device, goes to the device registration page and starts the process of registering a device by clicking "Register device." It makes the backend(B_{end}) build a deviceRegReq with the device and device-generated TPM keys.

ii. Within the device, the Trusted Platform Module produces TPM_{K_A} (Attestation Key of TPM) and signs a freshness challenge from the back-end(B_{end}), and thus produces an attestation status.

iii. It takes the TPM's attestation quote, device details, and the TPM's public keys. Then it passes these as an attestDeviceReq to the attestation service for standalone attestation verification.

iv. The attestation service takes in the Attestation Service Payload ($A_{SEV_{PL}}$), checks the device's measurements and platform state against approved security baselines, and if successful with the validation checks, creates a new D_{R_S} (Device Registry Status) entry with an active status in the D_R (Device Registry).

v. When receiving device registration confirmation from the registry, the back-end sends Keycloak(K_C) credential issuance request with a credentialReq that holds the user and device identifiers and nonce as proof of request freshness.

vi. Keycloak(K_C) or the credential authority issues a device-bound credential and securely delivers it to the backend(B_{end}), where it is provisioned into the TPM on the device.

vii. Last of all, the device registry page also notifies the end user that the device has successfully registered and appears as an active device under the D_R (Device Registry) and can perform subsequent authentication and access operations.

5.3 TPM-based Federated Identity Authentication Protocol

Authentication leverages the OIDC protocol, supplemented by hardware-bound assertions. Authentication flows are triggered by the end-user through oidcAuthReq, with Keycloak challenging the device using WebAuthn/TPM mechanisms. The device's TPM signs the challenge and sends back the response for verification. Keycloak(K_C) checks both the A_{SEV_R} (Attestation Service Report) and D_{R_S} (Device Registry Status) to determine that the device continues to be trusted, verifying the stored state and valid attestation. With successful verification and authentication, Keycloak(K_C) returns T_{OIDC} (OIDC provided token) and T_{DPoP} (Demonstrating Proof of Possession Token). These session tokens grant the end user seamless access to federated resources with the guarantee that only an enrolled and attested device has access to the tokens.

The protocol flow suggested in this phase is represented as Figure 5.2 the protocol of it can be seen from Table 5.4 . The Authentication protocol flow that we are going to discuss next.

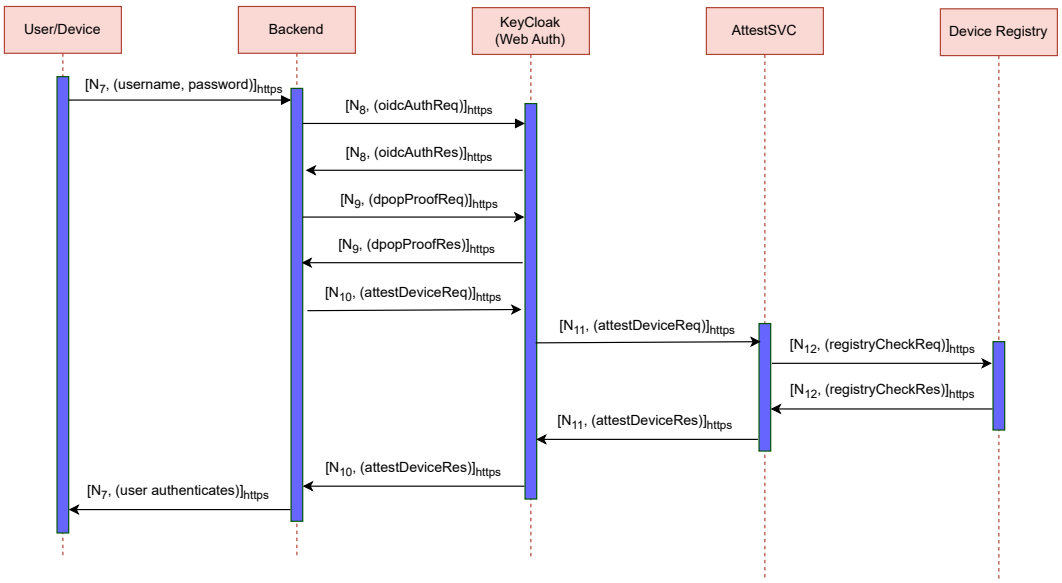


Figure 5.2: Authentication Flow

M	Description
M1	$U \;\; B_{\text{end}} : [N_7, (\text{username, password})]_{HTTPS}$
M2	$B_{\text{end}} \;\; K_C : [N_8, (\text{oidcAuthReq})]_{HTTPS}$
M3	$K_C \;\; B_{\text{end}} : [N_8, (\text{oidcAuthRes})]_{HTTPS}$
M4	$B_{\text{end}} \;\; K_C : [N_9, (\text{dpopProofReq})]_{HTTPS}$
M5	$K_C \;\; B_{\text{end}} : [N_9, (\text{dpopProofRes})]_{HTTPS}$
M6	$B_{\text{end}} \;\; K_C : [N_{10}, (\text{attestDeviceReq})]_{HTTPS}$
M7	$K_C \;\; A_{\text{SEV}} : [N_{11}, (\text{attestDeviceReq})]_{HTTPS}$
M8	$A_{\text{SEV}} \;\; D_R : [N_{12}, (\text{registryCheckReq})]_{HTTPS}$
M9	$D_R \;\; A_{\text{SEV}} : [N_{12}, (\text{registryCheckRes})]_{HTTPS}$
M10	$A_{\text{SEV}} \;\; K_C : [N_{11}, (\text{attestDeviceRes})]_{HTTPS}$
M11	$K_C \;\; B_{\text{end}} : [N_{10}, (\text{attestDeviceRes})]_{HTTPS}$
M12	$B_{\text{end}} \;\; U : [N_7, (\text{user authenticates})]_{HTTPS}$

Table 5.4: TPM-based Federated Identity Authentication Protocol

i. User(U) wishes to gain access to a secured app and initiates the login process by visiting the app's login page. The backend(B_{end}) redirects the user to Keycloak(K_C), triggering an OIDC authentication flow through an oidcAuthReq from the device and the user identifiers.

ii. Keycloak(K_C) issues a WebAuthn or TPM challenge to the browser of the user. The browser issues a challenge to the TPM of the device, and the TPM computes an attestation and a cryptographic signature using its private key. The browser returns the assertion signed and attestation data back to Keycloak(K_C).

iii. Keycloak(K_C) checks the attestation sent through an attestDeviceReq and also checks device status at the Device Registry(D_R) through a registryCheckReq. It checks that the device remains valid, active, and not revoked or suspended.

iv. The Attestation Service(A_{SEV}) returns Keycloak(K_C) attestation validation results, attestDeviceRes. If device attestation succeeds, Keycloak(K_C) continues; otherwise, Keycloak(K_C) rejects the authentication request with an optional request for further authentication, step-up authentication.

v. When registry checking and attestation are successful, Keycloak(K_C) produces T_{OIDC} (OIDC provided token). For proof-of-possession, T_{DPoP} (Demonstrating Proof of Possession Token) is sent as a verification assertion with a device key, so that only the registered device can consume tokens.

vi. The backend(B_{end}) receives the authentication response, completing the login process for the user. The session is now strongly tied to both the user and their registered device, establishing a secure foundation for subsequent resource access operations.

5.4 ZTA-based Resource Access Protocol

When the user (U) tries to access the protected service, the device includes the access token received above and a DPoP proof, signed by the device's private key, in the resource request (resourceAccessReq). The Decision Controller (D_{Ctrl}) verifies the validity of the DPoP proof, ensuring that the requesting device owns the token. Then, the policy engine matches the access request (policyEvalReq) with predefined policies, the latest device attestation status, registry status, and user attributes. The outcome determines the service provider's response. All authorization decisions, token verifications, and device interactions are logged for compliance and audit purposes.

The protocol flow for this phase is suggested to be shown in Figure 5.3 and the corresponding protocol messages are shown in Table 5.5 We now consider the resource access protocol flow.

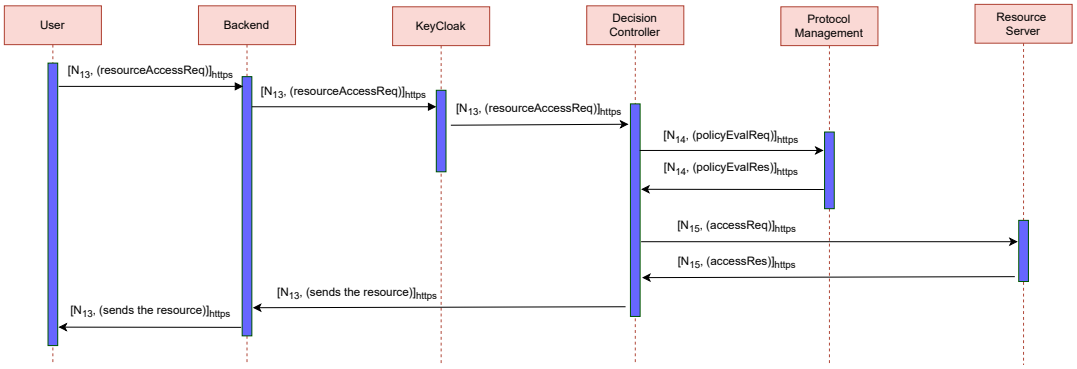


Figure 5.3: Resource Access Flow

M	Description
M1	$U \quad B_{\text{end}} : [N_{13}, (\text{resourceAccessReq})]_{HTTPS}$
M2	$B_{\text{end}} \quad K_C : [N_{13}, (\text{resourceAccessReq})]_{HTTPS}$
M3	$K_C \quad D_{\text{Ctrl}} : [N_{13}, (\text{requestAccessReq})]_{HTTPS}$
M4	$D_{\text{Ctrl}} \quad Ptcl_{\text{Mnt}} : [N_{14}, (\text{policyEvalReq})]_{HTTPS}$
M5	$Ptcl_{\text{Mnt}} \quad D_{\text{Ctrl}} : [N_{14}, (\text{policyEvalRes})]_{HTTPS}$
M6	$D_{\text{Ctrl}} \quad R_{\text{SVR}} : [N_{15}, (\text{accessReq})]_{HTTPS}$
M7	$R_{\text{SVR}} \quad Ptcl_{\text{Mnt}} : [N_{15}, (\text{accessRes})]_{HTTPS}$
M8	$D_{\text{Ctrl}} \quad B_{\text{end}} : [N_{13}, (\text{sends the resource})]_{HTTPS}$
M9	$B_{\text{end}} \quad U : [N_{13}, (\text{sends the resource})]_{HTTPS}$

Table 5.5: ZTA-based Resource Access Protocol

i. After a successful authentication, the user's agent desires access to a secured resource by sending a request to the backend. This request contains the obtained

access token somewhere along with a DPoP proof in the request's header, which provides proof-of-possession for the token. This is the starting point for a resourceAccessReq, which holds the user ID, the ID of the device, the access token, DPoP proof, identifier for the resource, and corresponding audit information.

ii. The backend forwards the resourceAccessReq to the Decision Controller(D_{Ctrl}), which uses a policyEvalReq to evaluate a policy for the access request. This request contains the output of the token introspection, DPoP verification, device ID, attestation summary, resource ID, and scope of accesses being requested.

iii. The Decision Controller(D_{Ctrl}) makes requests to auxiliary services when needed. These are a request for verification of a token and scope from Keycloak(K_C), a new quote for a TPM from the Attestation Service (A_{SEV}) if the policy requires updated evidence, and compliance or revocation state from the D_R (Device Registry). The results returned are attestation results, up-to-date registry state (active, revoked, or suspended), and token validity information.

iv. The Policy Decision Point (PDP) processes the request based on all information it has collected and returns a policyEvalRes object with its decision: permit, deny, or conditionally permit access. Decision reasons and obligations might be returned, too.

v. The Decision Controller (D_{Ctrl}) applies the PDP decision. On denial, the controller returns a "denied" indication and logs the occurrence for auditing purposes. On allowed, the controller applies the terms that have been defined and retrieves corresponding resource information from the Resource Server (R_{SVR}).

vi. The Resource Server (R_{SVR}) forwards the received data (payload, metadata) to the Decision Controller, which forwards it to the backend, then the application interface of the user application. The status, any policy conditions that are being enforced, and audit information are saved for tracking and compliance.

This phase provides for strict control over resource access with dynamic, granular policy verification, which binds together the attestation state of the device, active validation of tokens, and live policy evaluation at transaction time.

5.4.1 Cryptographic Resource Access Protocol

This section presents a cryptographic resource access protocol aligned with the proposed Zero Trust architecture. The protocol operates between a *User End* and a *System End* and treats authorization as a cryptographic artifact rather than a derived session state.

Entities

- U: End user
- D: User device equipped with TPM

- TPM: Trusted Platform Module on D
- IdP: Identity Provider
- DR: Device Registry
- PDP: Policy Decision Point
- PEP: Policy Enforcement Point
- R: Requested resource

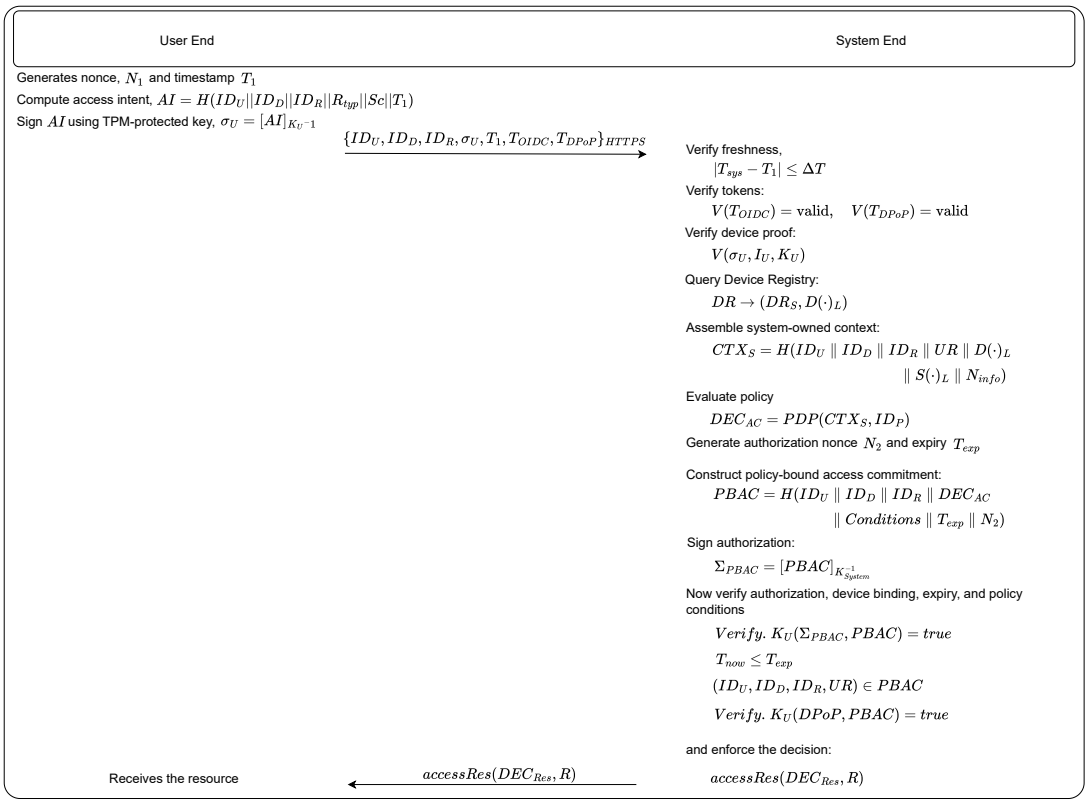


Figure 5.4: Cryptographic Resource Access Flow

The proposed cryptographic resource access flow adopts a two-end Zero Trust design, explicitly separating the responsibilities of the User End and the System End, as illustrated in Figure 5.4. At the User End, the device generates a fresh nonce N_1 and timestamp T_1, which are used to compute an access intent AI that binds the user identity, device identity, requested resource, access type, scope, and temporal context. The access intent is signed using a TPM-protected private key, ensuring that the request is device-bound and resistant to forgery, and is transmitted to the system together with OIDC and DPoP tokens over a secure channel.

Upon receiving the request, the System End performs a sequence of independent verifications. First, it validates request freshness by comparing the user-provided timestamp T_1 against the system-observed receipt time T_{sys}. It then verifies the legitimacy of the OIDC and DPoP tokens and validates the TPM-backed signature to

confirm device authenticity. The system subsequently queries the device registry to retrieve the device's trust status and assembles a system-owned contextual representation incorporating identity attributes, device trust level, request intent, sensitivity classification, and network context. This context is evaluated by the policy decision point to derive an access decision.

If access is permitted, the system generates a fresh authorization nonce N_2 and an explicit expiration time T_{exp}, which are embedded into a policy-bound access commitment (PBAC). The PBAC cryptographically encodes the authorization decision, binding it to the user, device, requested resource, policy conditions, and validity window, and is signed by the system authority. During enforcement, the system verifies the PBAC signature, binding constraints, expiration, and proof-of-possession before enforcing the final decision and returning the protected resource to the user.

5.5 Model Verification

Here, we would provide the mathematical verification of the proposed architecture.

5.5.1 Preliminaries

We define the core sets and tuples derived from the architecture:

Users: Let U be the set of users in the system.

$$U = \{u_1, u_2, \ldots, u_n\} \tag{5.1}$$

Devices: Let D be the set of devices in the system.

$$D = \{d_1, d_2, \ldots, d_m\}, \quad \text{where each } d_i \text{ has TPM-backed attestation} \tag{5.2}$$

Resources: Let R be the set of resources that are to be accessed in the system.

$$R = \{r_1, r_2, \ldots, r_k\} \tag{5.3}$$

Capabilities: Let C be the set of capabilities that can be present in the system.

$$C = \{c_1, c_2, \ldots, c_p\} \quad where \quad C_j = \{r_j, ops_j, constraints_j\} \tag{5.4}$$

here r_j = resource a user/device can access, ops_j = operations (read, write, execute, delete, etc.), and $constraints_j$ = contextual conditions (time, location, device state).

Device Registry: Let DR be the device registry where d_i denotes each device, $attestStatus(d_i)$ denotes the attestation status of each device, and TL_i represents the trust level.

$$DR = \{d_i, attestStatus(d_i), TL_i\}, \quad TL_i = \{H_L, M_L, L_L\} \tag{5.5}$$

Access Requests: Let AR be the access request done by each user or device, where each request consists of user ID(ID_U), device ID(ID_D), resource(R), and access intent(AI).

$$AR = \{ar_1, ar_2, \ldots, ar_q\}, \quad ar_i = \{ID_U, ID_D, R, AI\} \tag{5.6}$$

5.5.2 State Variables

$$attestStatus(d_i) \in \{Valid, Invalid\} \tag{5.7}$$

denotes the outcome of TPM-based attestation for the device d_i, indicating whether the device integrity has been successfully verified.

$$token(ID_U, d_i) \text{ issued if } attestStatus(d_i) = Valid \tag{5.8}$$

specifies that a user-device bound authentication token is issued only when the associated device has passed integrity attestation.

$$policyDecision(ar_i) \in \{Allow, Deny\} \tag{5.9}$$

represents the authorization verdict produced by the policy decision engine for access request ar_i.

$$accessResult(ar_i) \in \{Granted, Denied\} \tag{5.10}$$

captures the final enforcement outcome of the access request ar_i after integrating authentication, attestation, and policy evaluation results.

5.5.3 Thresholds and Logical Symbol

TL_{min} - minimum trust level required to consider a device trustworthy for access.

$(\forall) universalquantifier, (\exists) existentialquantifier, (\wedge) logicaland, (\vee) or,$
$(\Leftrightarrow) logicalequivalence, (\subseteq) subset.$

5.5.4 System Invariants

To capture the essential security guarantees of the system, the following formal properties are defined:

Device Integrity Enforcement:

$$\forall d_i \in D,\ \forall token(ID_U, d_i) \Rightarrow attestStatus(d_i) = Valid \tag{5.11}$$

- Let D the set of all registered devices.
- Let $d_i \subseteq D$ represent a particular device.
- Let $token(ID_U, d_i)$ denote a token issued to user U for device d_i.
- Let $attestStatus(d_i)$ represent the attestation result of the device, whether it's trusted or not.

So, a token can only be issued if that device's attestation status is valid.

Registry Consistency

$$\forall d_i \in D,\ DR(d_i) \text{ is up-to-date with the latest attestation.} \tag{5.12}$$

- Let DR denote the Device Registry.
- Each d_i in D must have it's attestation data synchronized in DR.

The Device Registry must always hold the most recent attestation record for each registered device.

Policy Compliance

$$\forall ar_i \in AR,\ accessResult(ar_i) = Granted \Rightarrow policyDecision(ar_i) = Allow \tag{5.13}$$

- Let AR be the set of all access requests.
- Let $ar_i \subseteq AR$ represent a specific access request.
- Let $policyDecision(ar_i)$ denote the evaluation result from the Policy Decision Point(PDP).
- Let $attestResult(ar_i)$ be the final enforcement outcome at the Policy Enforcement Point(PEP).

So, if access is *granted*, it must have been explicitly allowed by the policy engine (protocol management).

Capability Enforcement

$$ar_i \text{ satisfies } c_j \Rightarrow requestedOps \subseteq ops_j \wedge constraints_j \text{ met} \tag{5.14}$$

- Let $C = c_1, c_2, ...c_p$ represent the set of all capabilities.
- Let $c_j \subseteq C$ be one capability associated with a resource.
- Let $requestedOps$ denote the operations (read, write, delete, etc.) requested in ar_i.
- Let ops_j be the permitted operations under capability c_j.
- Let $constraints_j$ be any contextual conditions (e.g., time, device health).

So, access is only valid if the requested operations fall within the allowed set, and all contextual constraints are met.

Zero Trust Enforcement

$$\forall ar_i \in AR,\ accessResult(ar_i) = Granted \Rightarrow \big(token(ID_U, ID_D) \text{ is valid} \wedge\ attestStatus(ID_D) = Valid \wedge\ DR(ID_D).TL \geq TL_{min}\big) \tag{5.15}$$

- Let ID_U represent the user identity, and ID_D the device identity.
- Let $DR(ID_D).TL$ denote the device's trust level recorded in the Device registry.
- Let TL_{min} be the system's minimum trust threshold.

So, access can be granted only if the token binding user and device is valid, the device's attestation status is verified, and the trust level of that device meets or exceeds the minimum required threshold.

These equations formalize the system's security invariants across five dimensions - device integrity, registry consistency, policy correctness, capability enforcement, and adherence to zero-trust principles. All granted access requests must pass these conditions to maintain end-to-end trust assurance.

5.5.5 Access Request Proof Flow

Consider an access request:

$$ar_i = (ID_D, ID_U, R, AI) \tag{5.16}$$

The process of authorization and validation proceeds through the following steps:

Step 1: Device Attestation

The backend retrieves the device attestation result from the attestation service:

$$attestStatus(ID_D) = Valid \tag{5.17}$$

Step 2: Keycloak Token Issuance

Keycloak issues a token only if the device attestation status is valid:

$$token(ID_D, ID_U) \Rightarrow attestStatus(ID_D) = Valid \tag{5.18}$$

Step 3: Policy Evaluation

The Decision Controller evaluates the access intent (AI) using the Policy Decision Point (PDP) and Policy Information Point (PIP):

$$policyDecision(ar_i) \in \{Allow, Deny\} \tag{5.19}$$

Step 4: Capability Check

Access is allowed only if the requested operations conform to the capability constraints:

$$ar_i \text{ satisfies } c_j \Rightarrow requestedOps \subseteq ops_j \wedge constraints_j \text{ met} \tag{5.20}$$

Step 5: Registry Verification

The device must maintain a trust level above the system-defined threshold:

$$DR(ID_D).TL \geq TL_{min} \tag{5.21}$$

Step 6: Formal Guarantee

$$\forall ar_i \in AR,\ \mathsf{accessResult}(ar_i) = \mathsf{Granted} \iff \begin{cases} \mathsf{attestStatus}(ID_D) = \mathsf{Valid}, \\ \mathsf{token}(ID_U, ID_D) \text{ is valid}, \\ \mathsf{policyDecision}(ar_i) = \mathsf{Allow}, \\ ar_i \text{ satisfies } c_j, \\ \mathsf{DR}(ID_D).\mathsf{TL} \geq TL_{\min} \end{cases} \tag{5.22}$$

This formal proof guarantees that device integrity, authenticated identity, policy compliance, capability constraints, and registry-based trust are all enforced before granting access.

5.6 Implementation

The study of the suggested Zero Trust-based federated identity system implies the combination of a hardware-based trust chain with innovative federated protocols. It is implemented based on a modular architecture with security responsibilities in separate server environments and observance of the concept of Zero Trust.

5.6.1 Hardware Attestation

The system provides the integrity of the devices through the Trusted Platform Module (TPM) of the user device. In the registration stage, the system records distinct hardware identifiers to have a long-term connection between the device and the security infrastructure:

EK Hash and TPM Hash: The Endorsement Key (EK) hash is obtained as a non-migratable identifier of the hardware as a special hash of the device TPM shown in Figure 5.5.
HMAC Sealing through HSM: To ensure that such identifiers are not altered in the database, EK hash is sealed with the help of an HSM simulator (SoftHSM). The system calculates HMAC of EK hash to make sure that in case of the registry

compromise, the identity of a device cannot be spoofed without knowledge of the keys in the HSM.
Re-verification: Each login will send a pre-authentication test in which a new HMAC is calculated and compared with the stored value with a timing safe equality test to ensure hardware modifications are detected, or devices are replicated.

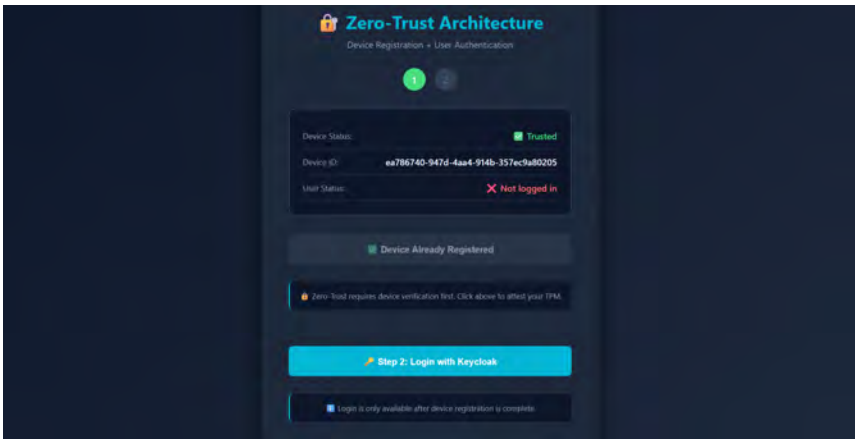


Figure 5.5: TPM Registration

5.6.2 Federated Identity and OIDC

Identity management is not tied to the resource server and instead is done by a special Identity Provider (IdP).

Keycloak Integration: Keycloak is used in the system to manage the OpenID Connect (OIDC) flow and provide standardized and interoperable authentication across federated domains through login shown in figure 5.6.
Secure Exchange: It is based on PKCE (Proof Key for Code Exchange) to secure authorization code grant and DPoP (Demonstrating Proof-of-Possession) to bind the access tokens to the particular client key created throughout the session.
User-Device Binding: On a successful OIDC callback, the system binds the verified user identity to the particular Device ID value held in the registry, which means that the system does not grant access unless the user and the verified device are both present.

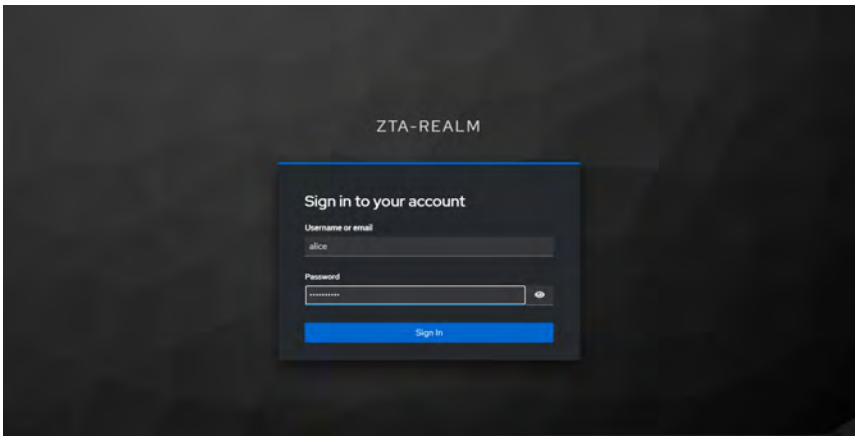


Figure 5.6: Keycloak Login

5.6.3 Contextual Risk and Adaptive MFA

Under the system, an Adaptive Multi-Factor Authentication (MFA) measure is used, where additional authentication layers are invoked as shown in Figure 5.7 and Figure 5.8 upon real-time risk evaluation.

Geolocation Detection: The application tracks the modification of IP addresses in order to identify changes that occurred in terms of geolocation.
Physical Location Tracking: The location also changes significantly, which is also monitored with the longitude and latitude values.
Step-up Authentication: When a location anomaly is identified, MFA is initiated, and the users are asked security questions that they were pre-enrolled with; these are saved in the form of cryptographic hashes to ensure user privacy.

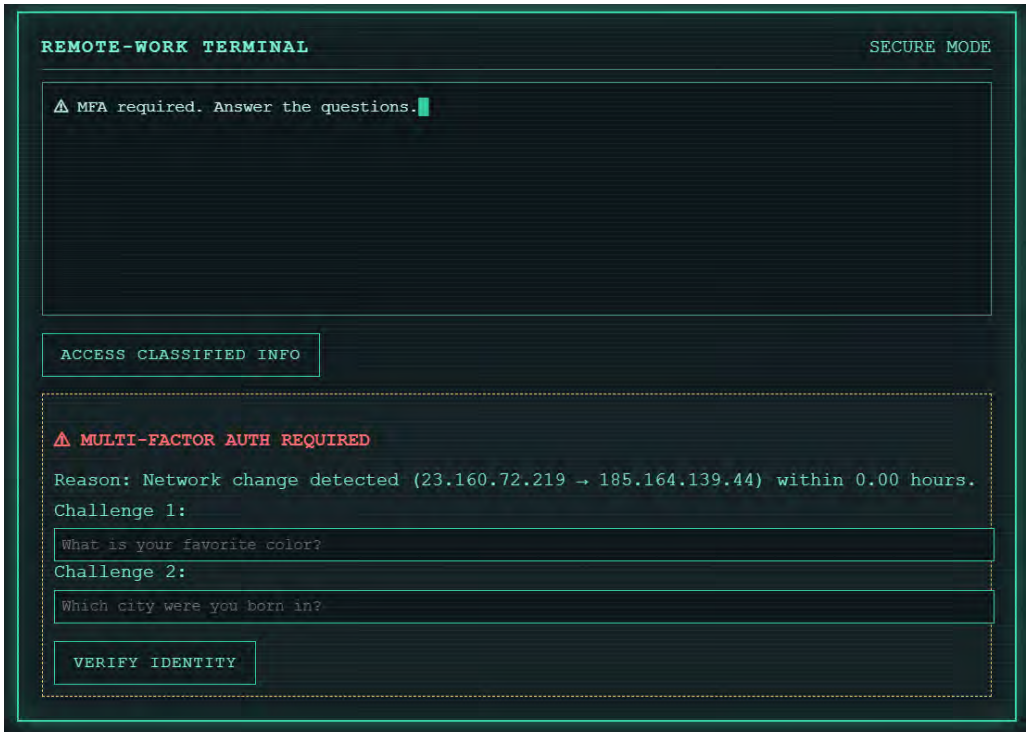


Figure 5.7: Adaptive MFA Activate

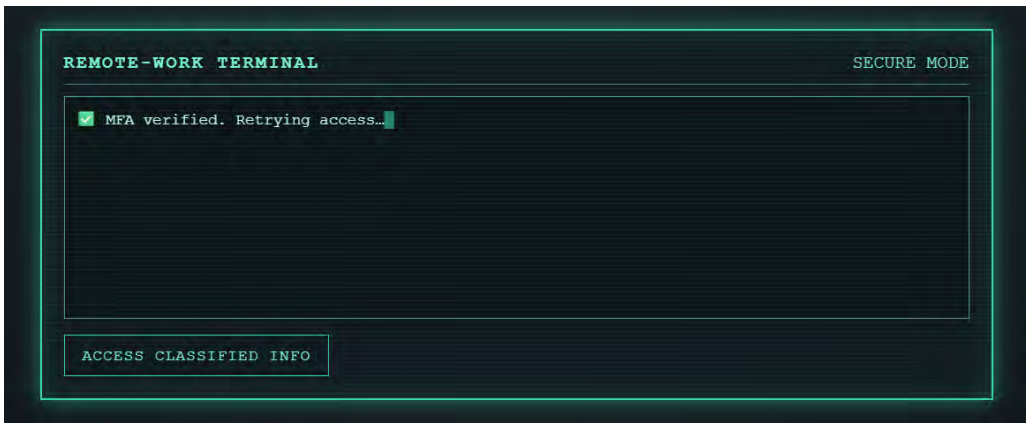


Figure 5.8: Adaptive MFA Verified

5.6.4 Distributed Server Architecture

The development environment is divided into distinct and specialized servers to reduce the attack surface and achieve single points of failure.

Backend Application Server: Handles the sessions, handles the attestation workflow and executes the final access decisions.
Identity Provider (Keycloak): It is managed by a different instance, and it is in charge of handling user credentials, roles, and issuance of OIDC tokens.
TPM Attestation Service: A dedicated service that runs on a separate host, like a Windows-based environment, which serves to interface with the physical TPM as well as provide hardware quotes.
Database (Firebase): It uses a separate database server as a persistent store, which is built on real-time to store device status, MFA enrollments, and audit logs.

Upon successful verification, if protocols are fulfilled system will let the user access the intended info as shown in Figure 5.9.

Figure 5.9: Access Resource

Access to the protected resource is granted only after the user successfully satisfies the Zero Trust Architecture (ZTA) evaluation. When contextual or risk-based conditions warrant additional assurance, the user must further complete the required Multi-Factor Authentication (MFA) challenges before authorization is finalized.

Chapter 6

System Performance

In this chapter, the significant empirical performance of the system could indicate that various critical parameters have been examined in detail. Moreover, the analysis may suggest that the proposed design solutions demonstrate important efficiency, and furthermore, the comparison among the relevant variables appears to illustrate key relationships. The chapter concludes with a discussion that might reveal the findings and could demonstrate that the significance for safe and scalable deployment of the system has been established.

6.1 Performance Evaluation

Figure 6.1 shows the response time and users graph for Zero Trust Architecture(ZTA). Nevertheless, between 300 to 900 users, response time may stay low and stable, from around 186 ms to 354 ms. However, at 1200 users, response time shows a slight increase to 795 ms. Given that the load increases, at 1500 users time increases drastically to 2580 ms. Thus, ZTA system may perform efficiently under regular moderate load but it goes through saturation under higher concurrency due to policy evaluation and verification.

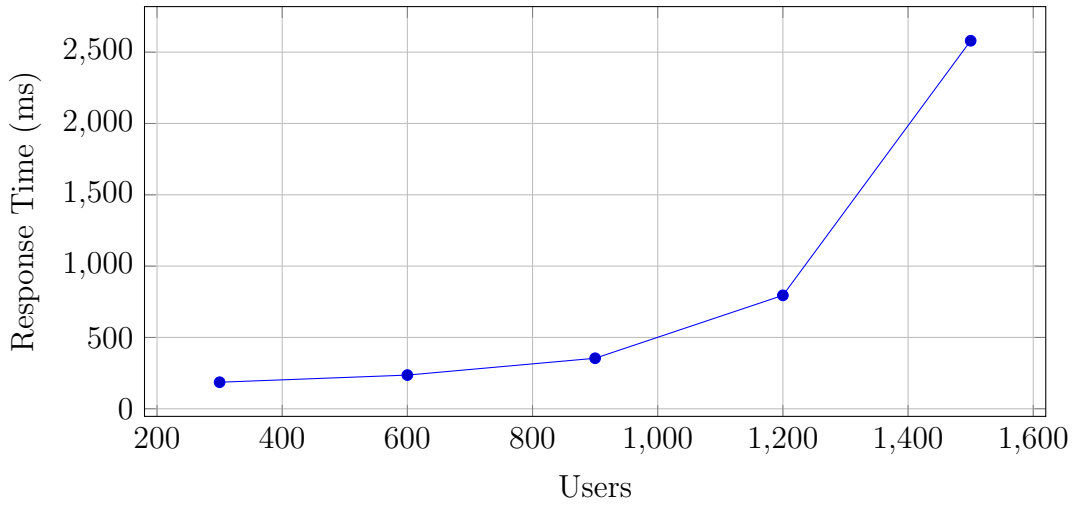


Figure 6.1: ZTA Response Time vs Users

Figure 6.2 shows delay measured for various cycle values in ZTA configuration. Additionally, delay might remain consistently low and stable up to 1200 users, and the

findings could suggest that request queuing and network overhead are low during normal operation. At 1500 users, a noticeable increase in delay appears, corresponding to response time spike in Figure 6.1. Therefore, this alignment may suggest internal authorization and decision processing overhead are main determinants of system latency once system is near reaching processing capacity.

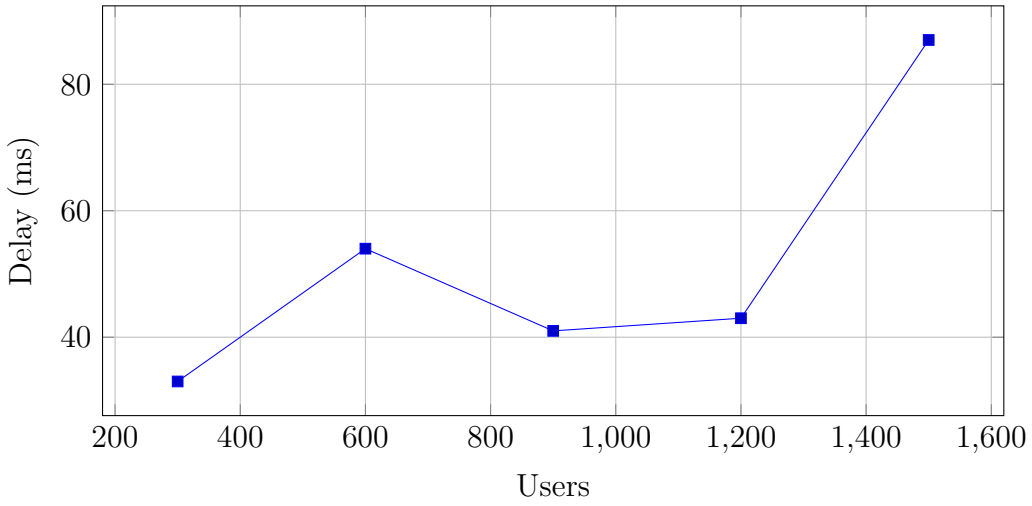


Figure 6.2: ZTA Delay vs Users

In light of the collected data, the significant throughput behaviour of the ZTA system could be illustrated in Figure 6.3. Moreover, throughput may rise fairly linearly from 300 to 1200 users, and the analysis might indicate that the system peaks at around 117 requests/second, showing important scalability under increasing load. At 1500 users, throughput drops to around 94 requests a second, indicating throughput degradation due to system saturation. However, this trend illuminates trade-off between secure evaluation and maximum sustainable throughput in ZTA design.

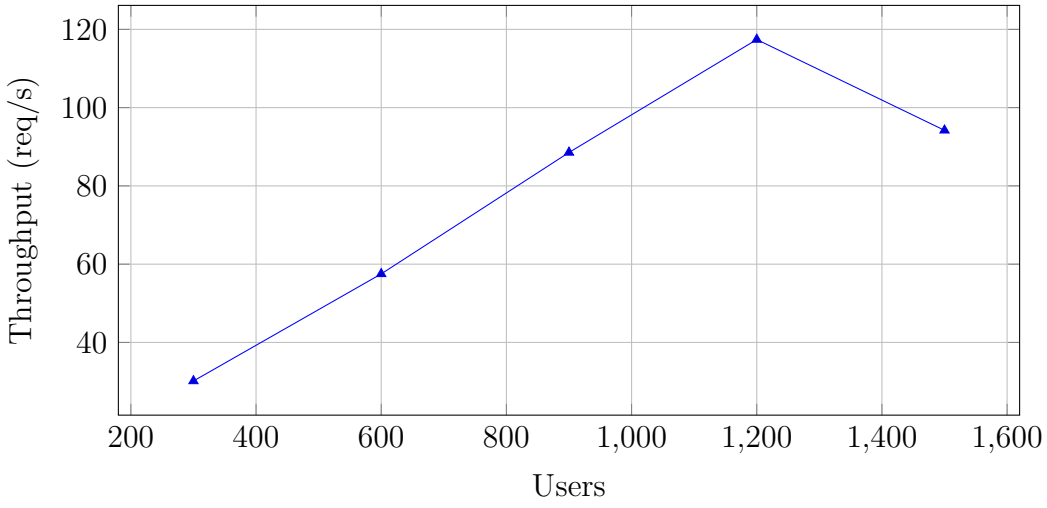


Figure 6.3: ZTA Throughput vs Users

Figure 6.4 could demonstrate that the response time performance of the MFA-enabled system appears fairly stable across all cycle values. Moreover, the MFA

response time may fluctuate between approximately 299 ms and 336 ms up to 1200 users. Furthermore, response time rises to about 570 ms at 1500 users, which might indicate moderate saturation effects. Thus, the MFA-enabled system shows smoother scaling behavior under high load compared to ZTA. Given that the decision logic is comparatively simpler, the system may maintain consistent performance characteristics.

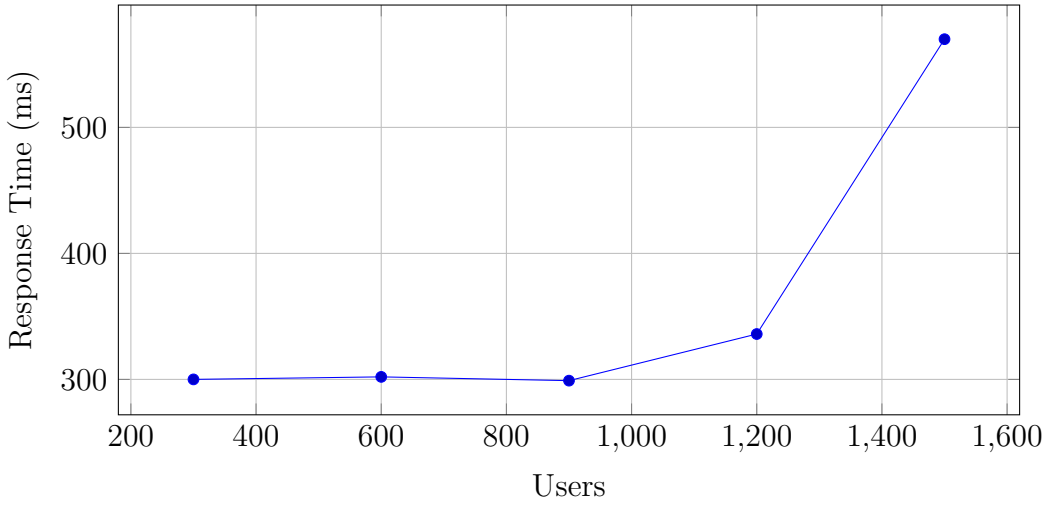


Figure 6.4: MFA Response Time vs Users

As shown in Figure 6.5, delay characteristics of the MFA-enabled system could follow the response time trend. However, delay values appear bounded for all cycle counts with no sudden spikes. Nevertheless, conditional MFA checks seem efficiently integrated and may not introduce excessive queuing or network-induced latency.

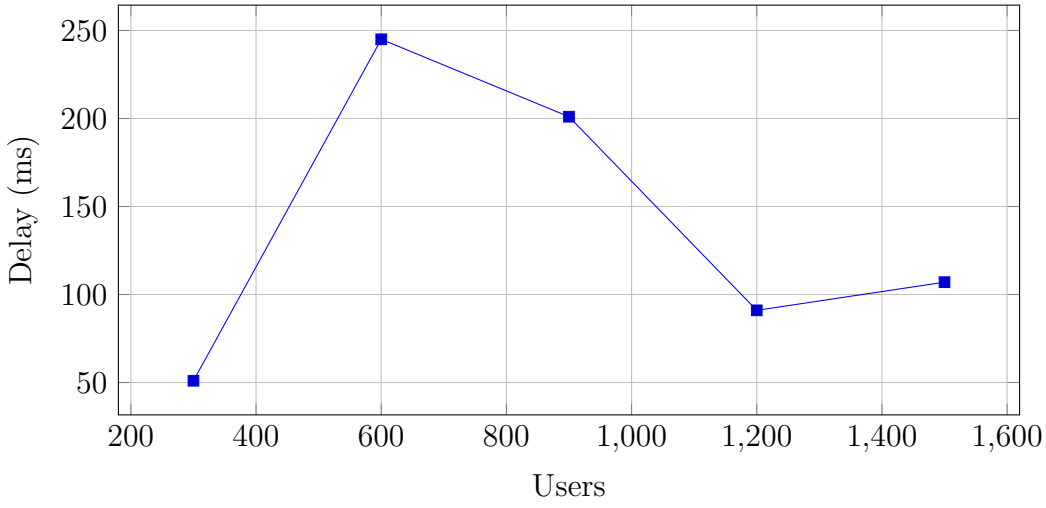


Figure 6.5: MFA Delay vs Users

Finally, Figure 6.6 might show throughput in the MFA-enabled configuration. Throughput could rise fairly consistently with cycles, peaking at approximately 109 requests per second at 1200 users. In light of higher concurrency levels, throughput slightly falls at 1500 users, which may suggest that MFA integration doesn't significantly reduce throughput and could retain stable request handling capacity.

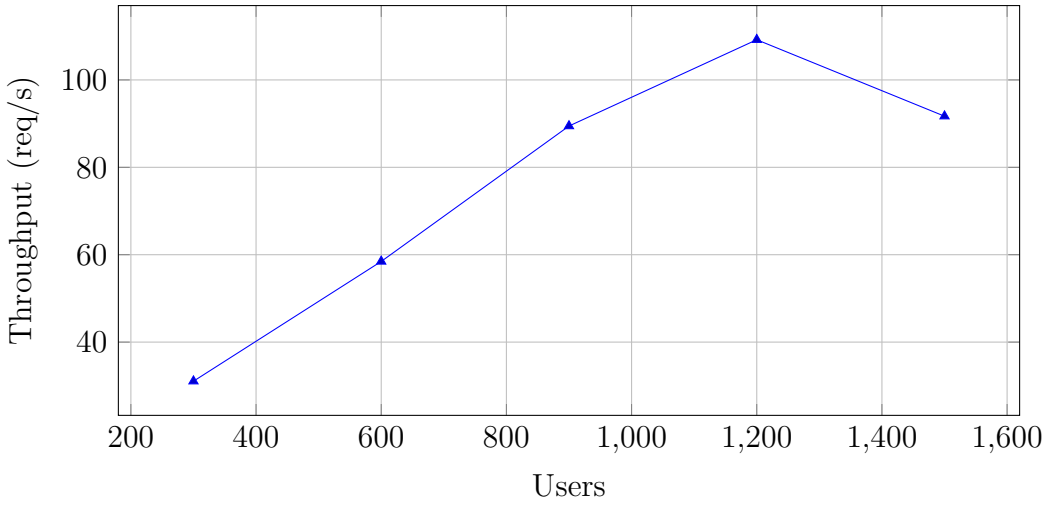


Figure 6.6: MFA Throughput vs Users

6.2 Design Solution Analysis

This part of the paper will examine the proposed design solutions and how the solutions effectively mitigate the discussed security issues with Federated Identity Systems through a Zero Trust Architecture. The system design consists of federated authentication based on the OpenID Connect protocol, device trust based on hardware attestation of TPM, and Adaptive Multi-Factor Authentication (MFA) to enable continuous verification and least-privilege access.

The implementation of OpenID Connect offers a standard and interoperable basis of federated authentication in various domains. The proposed design, in contrast to the traditional federation models, which are based on the strong assumptions of static trust, introduces a strong token validation process, temporary access tokens, and dynamic policy verification. This minimizes vulnerability to token replay attacks and session hijacking attacks that are typical vulnerabilities in traditional federated systems.

TPM-based registration and attestation of the device is also of great importance in improving the trust model of the system by locking authentication to authenticated devices. The system reduces threats like credential theft, device spoofing, and unauthorized access because it verifies the integrity of the device the end-user is registering with, as well as the integrity of the access requests, including the compromised endpoints. This method is compatible with the principles of Zero Trust because it does not presuppose implicit trust in devices and makes access decisions based on the identity and device posture.

Authentication is further enhanced by the integration of Adaptive MFA to dynamically modify the requirements of the verification according to the risk factors present in the context, like the trustworthiness of the device, the location of access, and the behavior of the session. This method is better resistant to phishing and credential theft than static MFA, and is more usable as it does not need extra authentication when faced with low-risk situations.

6.3 Statistical Analysis

The statistical analysis used to analyze system behaviour over periods of increasing operational users is based on the performance metrics recorded regarding the Zero Trust Architecture (ZTA) and the Adaptive Multi-Factor Authentication (MFA) configurations.

users	ZTA			MFA		
	Response T.	**Delay**	**Throughput**	**Response T.**	**Delay**	**Throughput**
300	186	33	30.14	300	51	31.05
600	236	54	57.51	302	245	58.46
900	354	41	88.54	299	201	89.45
1200	795	43	117.4	336	91	109.2
1500	2580	87	94.2	570	107	91.7

Table 6.1: Performance Comparison Between ZTA and MFA Across Test Users

6.3.1 Response Time and Delay Analysis

From table 6.1, the analysis of Response Time and delay of the two configurations, which are the Zero Trust Architecture (ZTA) and an Adaptive Multi-Factor Authentication (MFA), has had different performance attributes with an increment in cycle loads.

ZTA Trends: The original ZTA setup shows that there is a non-linear growth in the response time with the number of users. The performance remains relatively constant until 900 users, then it dramatically increases at 1200 (795 ms) and 1500 (2580 ms) users, indicating a significant processing overhead or possible saturation of continuous verification at high loads.

MFA Trends: The MFA configuration, on the contrary, has a much more stable response time profile with a clear peak in the range of 300 to 1200 users, with a median response time of about 300 ms. Response time decreases to 570 ms at 1500 users, but is still greatly reduced compared to the baseline ZTA when operated at the same load, implying that more optimized execution paths can be used by adaptive enforcement when it is operated on a sustained basis.

Delay values of the ZTA configuration are relatively low and steady, with maximum delay values of 87 ms as response times explode. In the MFA arrangement, delay exhibits a large variation with the highest point of 245 ms in 600 users, and then records a downward pattern as the number of users grows. This implies that MFA causes an initial latency through conditional challenges, but the system is effective in balancing such overheads as the load levels stabilize.

6.4 Comparisons and Relationship

This section compares and contrasts the proposed framework with the state-of-the-art research that has been found in the literature review. Using the comparative

metrics developed in table 6.2, the particular progress of this system and purposeful design decisions can be placed in the context of the existing systems.

Paper / Features	Hard. Security	Tech./ Tools	Perf. Analysis	Prot. Flow	Threat Mod.	Implem.	Compar. Eval.	Practical Recomm.
Naik et al. [17]	✗	✗	✗	✗	✗	✗	✓	✓
Foltz et al. [26]	✗	✓	✗	✗	✗	✗	✓	✓
Sharma [30]	✗	✓	✗	✗	✗	✓	✓	✓
Brazhuk et al. [29]	✗	✗	✗	✗	✗	✗	✓	✓
Kerman et al. [22]	✗	✓	✗	✗	✗	✓	✗	✓
Chennuri [32]	✗	✓	✓	✗	✗	✗	✓	✓
Chadwick et al. [3]	✗	✓	✗	✗	✗	✗	✓	✗
Dazki et al. [35]	✗	✓	✓	✗	✗	✓	✓	✓
Uehara et al. [43]	✗	✗	✗	✗	✗	✗	✗	✓
Lewis et al. [4]	✗	✗	✗	✗	✗	✗	✗	✓
Hasan [33]	✗	✗	✗	✗	✗	✓	✓	✓
Proposed System	✓	✓	✓	✓	✓	✓	✓	✓

Table 6.2: Feature Comparison of Selected Papers

This comparison shows that the practical aspect of bringing comparative evaluation, performance benchmarks, and hardware security features is paramount in ensuring that this system leads in secure practices of managing digital identity.

Hardware Security: In contrast to most of the current papers that only address protocol-level federation, this system employed TPM attestation and key sealing as implemented through HSM to guarantee that devices are intact.
Performance Analysis: Experiments were performed with 300 to 1500 users of operation at the device level, and the following performance values were taken: Response Time, Delay, and Throughput. This gives concrete data on the overhead of Zero Trust and MFA, which was perceived to be missing in much of the previous literature.

Implementation: The system includes an end-to-end implementation of Keycloak, Windows TPM services, and a Firebase backend, going out of the purely theoretical models that have been frequently introduced in previous research.
Comparative Evaluation: A systematic comparison with the models of state-of-the-art is also included in the work to identify the effectiveness and practical viability of the integrated approach.
Practical Recommendation: The work proposes practical insights on closing the state of trust gap across the federated domains and further development of the best practice in enforcing digital identity management.
Threat Modelling: A detailed threat modeling part is present here, which applies the STRIDE model to differentiate and categorize the possible security and privacy risks.

Protocol Flow: The protocol flows of registration, authentication, and access to resources are provided, giving a proper roadmap of what can be done to ensure secure communication in the federated framework.

6.5 Discussions

The results of this study demonstrate that the implementation of Zero Trust Architecture (ZTA) and Federated Identity Systems (FIS) is very effective in relation to resistance to identity-based attacks. The suggested system combines the hardware-supported trust through TPM attestation with adaptive authentication schemes to address the fatal flaws of the current federated models, in particular, the ones that are based on the notion of fixed trust relationships and on the assumptions about the perimeter. The system will address the challenge of authentication by implementing Adaptive Multi-Factor Authentication (MFA) and will adjust authentication needs according to real-time risk indicators (user behavior and device posture). An HSM simulator (SoftHSM) was employed in the system end to maintain the integrity of the cryptographic lifecycle by ensuring the protection of the keys in the TPM. The strategy will see to it that the attestation of devices is a fundamental aspect of reinforcing the authenticity judgments in inter-organizational contexts.

6.5.1 Requirement Analysis

The functionality and security requirements, which were established at the initial design phase, are effectively fulfilled by the suggested system. The following requirements have been met by combining hardware-based trust and federated protocols:

Federated Identity Authentication (F1): The system can successfully authenticate the users using the Keycloak system through the OpenID Connect (OIDC) protocol.

Per-Session Authorization (F2): The architecture makes use of short-lived tokens and dynamic policy evaluation with the help of the Decision Controller to provide each request with authorization.

Adaptive Multi-Factor Authentication (F3): The system will activate MFA dynamically based on risk indicators that the system identifies, including IP-based geolocation or physical coordinates changes.

Context-Aware Policy Evaluation (F4): The Policy Decision Point (PDP) uses the identity of the user along with the attestations of their device and environmental indicators to decide whether or not to grant access.

Attribute Disclosure Control (F5): The selective data sharing of user attributes is controlled according to the definite policies and the user's agreement.

Secure Communication (S1): Every communication between the User End and System End will be safeguarded with encrypted communication channels, namely HTTPS and DPoP.

Dynamic Access Policies (S2): Recent policy access is subject to real-time and dynamic granting or denial by the Decision Controller, due to live policy evaluation, and is not based upon the network zones as mitigation of risks of privilege escalation.

Token Security and Binding (S3): Tokens are provided to be bound to the particular registered device as part of DPoP (Demonstrating Proof-of-Possession) to avoid unauthorized reuse of a token.

Minimal Trust and Continuous Validation (S4): The principle of always checking with the never-trust policy is observed by TPM re-verification before any logging in and constant device integrity checking.

Monitoring and Audit Logging (S5): All access and identity operations are recorded to support behavioral analysis and traceability.

6.5.2 Objective Analysis

The study was able to fulfill the set aims (RO1-RO5) by creating a single framework implementing risk-aware authentication via distributed trust boundaries:

Analysing current authentication mechanisms (RO1): The research took into account the shortcomings of the traditional models, including stagnant trust and token abuse, offering a starting point to the zero-trust change.

Research on ZTA and MFA Integration (RO2): The study found out the ways in which hardware attestation and adaptive MFA can be combined to provide a privacy-oriented authentication process in federated systems.

Design and create a system based on our proposal (RO3): The proposed system has been implemented, where to enhance modularity and security, the Keycloak, TPM attestation, and an HSM-simulated backend were created as separate server environments.

Testing and Evaluating (RO4): During testing among 1,500 users, the system was tested to be stable and high throughput (91.7), including the overhead of ZTA and MFA.

Identification and Mitigation of Risks (RO5): The system prevents the risks of spoofing, credential theft, and unwarranted privilege escalation by way of threat modeling based on STRIDE and the use of HSM-sealed EK hashes.

6.5.3 Advantages

The suggested architecture has various important enhancements to the conventional identity management systems. That is, it offers a holistic view of trust that includes identity, device integrity, and contextual risk assessment. Adaptive MFA also enables the authentication strength to increase dynamically according to such contextual hazards, which decreases the attack surface and reduces user burden when risks are low. The geolocation sensitivity realized by means of the IP address tracking and coordinate monitoring (longitude/latitude) is an accurate method of identifying anomalies depending on the location. Also, registration through TPM restricts access to particular and verified devices, which is essentially effective in preventing the theft of credentials and spoofing of devices. Using OpenID Connect, the design ensures a standard and interoperable foundation of federated authentication as well as reducing the vulnerability to token replay and session hijacking with short-lived tokens and dynamically policy-validated.

6.5.4 Limitations

Although security was increased, some trade-offs and limitations characterize the scope of the system at the moment. Continuous verification and TPM attestation become embedded in the system, and it demands a high level of organization coordination in a federated environment at a high level. One of the main limitations of this study consists of the fact that an HSM simulator was used, and not physical hardware, to store TPM keys, which could fail to reflect the entire physical security picture of a dedicated hardware HSM. Also, hardware attestation is only made available to TPM-enabled devices, and the system cannot run as expected when such devices are not available or the hardware does not have a working module. Constant authentication and cryptographic operations also present a computational cost of their own that has to be included in the performance analysis. Lastly, the prototype was also constrained to a controlled test environment and would not capture the entire complexities or latencies of large-scale production deployments.

Chapter 7

Conclusion

In conclusion, a multi-factor authentication (MFA) approach that is adaptive and built into a Zero Trust Architecture (ZTA) has been suggested and applied to fill the existing security vulnerabilities of conventional federated identity systems. The study found that legacy models had major flaws, which included overreliance on static trust boundaries, susceptibility to token theft, and inadequate continuous session validation. The suggested framework will guarantee that all access requests are strictly checked, irrespective of the source, by changing to an integrated approach that would consolidate ZTA, Federated Identity System (FIS), and Adaptive Multi-Factor Authentication (AMFA).
The implementation was able to use the TPM-based hardware attestation and HSM-simulated key sealing to create a hardware-based trust chain, making it impossible to involve any devices outside the federation in the process. Moreover, the system was flexible to dynamic risk factors, such as changing the IP-based geolocation and physical coordinates, by requiring the triggering of step-up authentication challenges only in cases of anomalies. The proposed system has been tested for performance evaluation and it has provided a scalable and stable system with moderate throughput and latency. This research offers a solid, practical authentication framework to implement in a highly distributed environment.

7.1 Future Work

The principles of this study create a scaled model of a secure exchange of identities, although there are several potential ways to improve on this in the future:

Formal Security and Cryptographic Analysis: Future work will be on full formal security analysis to prove mathematically that it is resistant against more advanced attack vectors and the known cryptographic vulnerabilities.

Physical Hardware Integration: A shift in the existing HSM simulator and virtual environments to physical Hardware Security Modules (HSMs) and a more varied assortment of diverse TPM-enabled devices would assist in testing the physical security profile and even the real performance of the system.

Machine Learning to assess risk: The adaptive MFA engine can be enhanced by incorporating machine learning algorithms (both supervised and unsupervised) to

better the quality of anomaly detection and false positives of user behavior analysis.

Scaling Production Deployment: The testing will be scaled to a federated multi-organization setup to provide the opportunity to test the scalability, policy coordination, and incident response of infrastructures of heterogeneous nature.

Privacy-Preserving AI: Exploring Privacy-preserving AI models in risk evaluation will allow ensuring that data confidentiality is maintained to an extreme while also using the benefit of behavioral analytics.

Bibliography

[1] N. Ragouzis et al., *Security assertion markup language (saml) v2.0*, OASIS Standard, Mar. 2005.

[2] M. Hansen, A. Schwartz, and A. Cooper, "Privacy and identity management," *IEEE Security & Privacy*, vol. 6, no. 2, pp. 38–45, 2008.

[3] D. W. Chadwick, "Federated identity management," in *Foundations of Security Analysis and Design V*, ser. Lecture Notes in Computer Science, A. Aldini, G. Barthe, and R. Gorrieri, Eds., vol. 5705, Available: https://link.springer.com/chapter/10.1007/978-3-642-03829-7_3, Springer, 2009, pp. 182–196.

[4] K. D. Lewis and J. E. Lewis, *Web single sign-on authentication using saml*, Available: https://arxiv.org/abs/0909.2368, 2009.

[5] J. Kindervag, *Build security into your network's dna: The zero trust network architecture*, Online, Available: https://www.virtualstarmedia.com/downloads/Forrester_zero_trust_DNA.pdf, 2010.

[6] K. Z. Bijon, R. Krishnan, and R. Sandhu, "Risk-aware rbac sessions," in *Proc. 8th International Conference on Information Systems Security (ICISS)*, 2012, pp. 59–74.

[7] E. Ghazizadeh, M. Zamani, J. L. A. Manan, and A. Pashang, "A survey on security issues of federated identity in the cloud computing," in *Proc. 2012 IEEE 4th Int. Conf. CloudCom*, Taipei, Taiwan, 2012, pp. 562–568. DOI: 10.1109/CloudCom.2012.6427513.

[8] M. Jones et al., *Openid connect core 1.0*, OpenID Foundation, Nov. 2014.

[9] D. Nicks, "Target hackers accessed card data through a vendor," *TIME*, Jan. 2014, (accessed Jun. 17, 2025). Accessed: Jun. 17, 2025. [Online]. Available: https://time.com/2729/target-hackers-accessed-card-data-through-a-vendor/.

[10] M. S. Ferdous, G. Norman, A. Jøsang, and R. Poet, "Mathematical modelling of trust issues in federated identity management," in *IFIP Advances in Information and Communication Technology*, 2015, pp. 13–29. DOI: 10.1007/978-3-319-18491-3_2.

[11] A. K. Nag, A. Roy, and D. Dasgupta, "An adaptive approach towards the selection of multi-factor authentication," in *Proc. IEEE Symposium Series on Computational Intelligence (SSCI)*, 2015, pp. 463–472. DOI: 10.1109/SSCI.2015.75.

[12] European Union, *Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)*, https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng, Official Journal of the European Union, Apr. 2016.

[13] D. Fett, R. Küsters, and G. Schmitz, "A comprehensive formal security analysis of oauth 2.0," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Vienna, Austria, 2016, pp. 1204–1215.

[14] T. Hardjono and N. Smith, "Cloud-based commissioning of constrained devices using permissioned blockchains," in *Proc. of 2nd ACM Int. Workshop on IoT Privacy, Trust, and Security*, 2016. DOI: 10.1145/2899007.2899012.

[15] K. D. Lewis and J. E. Lewis, "Security assertion markup language (saml) for web-based single sign-on: Benefits, risks, and recent research," in *Proc. IEEE 10th Int. Conf. Semantic Comput.*, 2016, pp. 347–350.

[16] D. Fett, R. Kuesters, and G. Schmitz, *The web sso standard openid connect: In-depth formal security analysis and security guidelines*, https://arxiv.org/abs/1704.08539, 2017.

[17] N. Naik and P. Jenkins, "Securing digital identities in the cloud by selecting an apposite federated identity management from saml, oauth and openid connect," in *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, 2017. DOI: 10.1109/rcis.2017.7956534.

[18] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, 4th ed. Pearson, 2018.

[19] Y. R. Avuthu, *Federated identity and single sign-on (sso): Balancing security and usability in cloud iam implementations*, Available: https://www.jsaer.com/vol6/issue6/Paper3.pdf, Jan. 2019.

[20] A. Dasgupta and A. Roy, "Adaptive multi-factor authentication system with multi-user permission strategy to access sensitive information," Patent No. US20200014702A1, 2020. [Online]. Available: https://patents.google.com/patent/US20200014702A1/en.

[21] D. N. Divyabharathi and N. G. Cholli, "A review on identity and access management server (keycloak)," *International Journal of Security and Privacy in Pervasive Computing (IJSPPC)*, vol. 12, no. 3, pp. 1–14, Jul. 2020.

[22] A. Kerman, M. Souppaya, K. Scarfone, and S. Quincozes, "Implementing a zero trust architecture," National Institute of Standards and Technology, NIST Special Publication 800-207, 2020.

[23] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST, Tech. Rep., 2020, Available: https://doi.org/10.6028/NIST.SP.800-207.

[24] K. Dodanduwa and D. Kaluthanthri, *Role of trust in oauth 2.0 and openid connect*, 2021.

[25] SecurityWeek, *Continuous updates: Everything you need to know about the solarwinds attack*, Online, Available: https://www.securityweek.com/continuous-updates-everything-you-need-know-about-solarwinds-attack/, 2021.

[26] W. R. Simpson and K. E. Foltz, "Maintaining zero trust with federation," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 5, pp. 17–32, 2021. DOI: 10.46338/ijetae0521_03.

[27] X. Wang, Z. Yan, R. Zhang, and P. Zhang, "Attacks and defenses in user authentication systems: A survey," *Journal of Network and Computer Applications*, vol. 188, p. 103 080, 2021. DOI: 10.1016/j.jnca.2021.103080.

[28] J. Drga, I. Homoliak, J. Vančo, M. Perešíni, and P. Hanáček, "Detecting and preventing credential misuse in otp-based two and half factor authentication toward centralized services utilizing blockchain-based identity management," Nov. 2022, Available: https://arxiv.org/abs/2211.03490.

[29] E. B. Fernandez and A. Brazhuk, *A critical analysis of zero trust architecture (zta)*, Available: https://ssrn.com/abstract=4210104, 2022.

[30] H. Sharma, "Zero trust in the cloud: Implementing zero trust architecture for enhanced cloud security," *ESP Journal of Engineering Technology Advancements*, vol. 2, no. 2, pp. 78–91, Jun. 2022. DOI: 10.56472/25832646/JETA-V2I2P110.

[31] F. Fett, J. Lodderstedt, and B. Campbell, *Oauth 2.0 demonstrating proof-of-possession at the application layer (dpop)*, IETF Internet Draft, 2023.

[32] K. M. R. Chennuri, "Adaptive multi-factor authentication systems: A comprehensive analysis of modern security approaches," *International Journal of Computer Engineering and Technology*, vol. 15, no. 6, pp. 787–795, 2024, Available: https://iaeme.com/MasterAdmin/Journal_uploads/IJCET/VOLUME_15_ISSUE_6/IJCET_15_06_065.pdf.

[33] M. Hasan, "Enhancing enterprise security with zero trust architecture," *arXiv preprint*, 2024, arXiv:2410.18291.

[34] A. Lawrence, *Adaptive mfa: A smarter approach to authentication security*, Available: https://stytch.com/blog/adaptive-mfa/, Sep. 2024.

[35] A. Rahman, R. E. Indrajit, A. Unggul, and E. Dazki, "Implementation of zero trust security in msme enterprise architecture: Challenges and solutions," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 3, 2024.

[36] U. Shakir, "T-mobile promises to try not to get hacked again," *The Verge*, Sep. 2024. [Online]. Available: https://www.theverge.com/2024/9/30/24258763/t-mobile-fcc-settlement-cybersecurity-investment.

[37] *Streamlining operations with federated identity management*, Available: https://businesscasestudies.co.uk/what-is-federated-identity-management/, 2024.

[38] J. Tillson, *The crumbling castle*, Available: https://www.cyberdefensemagazine.com/the-crumbling-castle/, Feb. 2024.

[39] M. L. Gambo and A. Almulhem, *Zero trust architecture: A systematic literature review*, Available: https://arxiv.org/abs/2503.11659, 2025.

[40] J. Martin, *Encryption in transit in healthcare saas: The role of mutual tls*, Available: https://ssrn.com/abstract=5131928, Feb. 2025.

[41] OWASP Foundation, *OWASP Top Ten*, https://owasp.org/www-project-top-ten/, Accessed: 2025, Jul. 2025.

[42] Jegeib, *Threats - microsoft threat modeling tool - azure*, Microsoft Learn, Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats.

[43] S. Teerakanok, T. Uehara, and A. Inomata, "Migrating to zero trust architecture: Reviews and challenges," *Security and Communication Networks*, vol. 2021, Article ID 9947347.