

Identity Centric Cloud Security

Sonam Priya*, Subarna Shakya

Institute of Engineering, Pulchowk Campus, Tribhuvan University, Nepal

*Corresponding Author: sonampriya037@gmail.com

Abstract

Cloud computing's inherent scalability and flexibility necessitate robust Identity and Access Management (IAM) systems to ensure secure access to resources. This paper focuses on developing an identity-centric cloud security framework, with the primary goal of streamlining user authentication and access control. The project involved integrating Zendesk with the AWS console and implementing Single Sign-On (SSO) based on Role-Based Access Control (RBAC). AWS Cognito was employed to manage user identities, enabling secure authentication and authorization, while distinct roles (admin and basic) were defined to enforce granular access permissions. The implemented system successfully demonstrates secure user authentication and the effective use of RBAC to control resource access. By leveraging AWS services, the architecture provides scalability and facilitates real-time monitoring through CloudWatch, enhancing the ability to detect and respond to security incidents. The findings underscore the importance of an identity-centric security model in cloud environments and highlight the benefits of centralized identity management. The paper emphasizes the necessity of robust authentication and authorization mechanisms to secure cloud-based applications. The study advocates for the adoption of Identity Providers (IdPs) like AWS Cognito, which offer scalable and secure solutions for managing user identities. Furthermore, it recommends the careful definition of roles and permissions aligned with organizational security policies and continuous monitoring to maintain a strong security posture in the cloud.

Keywords: Cloud Computing, Cloud Security, AWS Cognito, Role Based Access Control (RBAC), Identity and Access Management (IAM), Identity Provider (IdP), Zendesk

1. Introduction

Cloud computing is a modern and powerful way to manage and use services over the internet, essentially an evolution of older computing systems where tasks were shared across different computers. It has brought huge advantages to the tech world by providing virtually endless storage and computing power [1]. What's more, cloud services operate on a "pay-as-you-go" basis, which makes it easy for companies to move their data and IT services to the cloud. This gives them on-demand access, wide network reach, and the ability to quickly scale up or down, all at a low cost. As a service-focused architecture, the cloud can deliver almost anything as a service, including software (SaaS), platforms (PaaS), and infrastructure (IaaS), along with more specialized services like databases (DbaaS) and identity management (IDaaS) [2]. Big players like Google, Amazon, and Salesforce are prime examples of companies that provide these convenient and affordable services to users. A critical part of any security system is access control, which verifies who is allowed to access what. When someone tries to access a facility, systems check their identity and authorize them based on credentials like PINs, passwords, or even fingerprints and smart cards. However, for companies moving to the cloud, managing identity and access becomes a significant challenge. Security and trust are major concerns when using distributed computing, and a strong identity and access management (IAM) system is essential to address these issues. With the growing need to handle external information and expand capacity, many organizations are struggling to extend their traditional access control methods beyond their own network firewalls into the cloud, especially for outsourced data [3]. Identity-centric

cloud security is the answer, focusing on securing digital identities and controlling access to cloud resources. This includes a robust IAM system for user authentication and role-based access, federated identity management for single sign-on across different systems using standards like SAML and OAuth, and continuous monitoring to detect and respond to any suspicious activity in real time [4].

2. Problem Statement

This paper explores how identity-centric approaches enhance cloud security by addressing the difficulty of managing Identity and Access Management (IAM) across multiple cloud service providers (CSPs), especially in federated systems where cloud services need to establish mutual trust.

3. Objectives

This paper explores how identity-centric approaches enhance cloud security by addressing the difficulty of managing Identity and Access Management (IAM) across multiple cloud service providers (CSPs), especially in federated systems where cloud services need to establish mutual trust.

The objectives of the project are as listed below:

- To enable access to the system and data using the integration of Zendesk with AWS console.
- To implement Single Sign-On (SSO) for Zendesk using role-based access control in AWS console.

4. Related Works

The transition to cloud computing represents the next seismic shift in information technology. In his seminal work, *The Big Switch*, Nicholas Carr drew a powerful analogy, comparing this transformation to the early 20th-century electrification of industry. Back then, factories stopped generating their own expensive power and simply plugged into the emerging electric utility grid. Computing is undergoing an identical metamorphosis, with businesses increasingly embracing a utility-based model for their IT infrastructure. The National Institute of Standards and Technology (NIST) formally defines cloud computing as a model that offers broad, easy, and on-demand network access to a shared pool of configurable resources—things like networks, servers, storage, and applications. Crucially, these resources can be rapidly provisioned (set up) and quickly released with minimal fuss from an administrator.

While companies are enthusiastically adopting cloud-based services, security remains the single largest hurdle. A survey conducted by IDC highlighted this clearly, finding that a staggering 87.5 percent of major IT enterprises cited security as their top concern. Making cloud access more inherently secure would undoubtedly spur broader adoption and drive down operational expenses. A key report from the Cloud Security Alliance (CSA) in 2013, "The Notorious Nine: Cloud Computing Top Threats," outlined essential dangers to help organizations make smarter, risk-informed choices. Historically, user login information was guarded within a company's physical premises. However, the shift to the cloud means a company's security perimeter is now stretched far beyond its own controlled IT environment. This makes traditional security methods obsolete, necessitating a reliance on more sophisticated and advanced security controls.

Identity and Access Management (IAM) is absolutely critical for guaranteeing secure access and forging trust between an organization and its cloud service providers (CSPs). In the cloud context, IAM encompasses the people, processes, and technological systems required to confirm a user's identity and subsequently grant them appropriate access to services. When a user signs up for a new cloud service, they often receive yet another set of credentials, which is a major pain point for individuals already overwhelmed with passwords. For cloud developers, it's a time-consuming burden to design and maintain diverse authentication methods. For IT admins, managing and recovering forgotten credentials is a difficult chore. Ultimately, this proliferation of login information makes credentials a prime target for cyber attackers.

Despite the compelling advantages of the cloud, migration has been slow, primarily due to persistent security worries concerning data privacy, transparency, risk management, and regulatory compliance. Consequently, cloud security has become a high-priority subject for both academic and industry research. A specific area of intense focus is how to securely handle and manage sensitive identity credentials. The act of storing and processing critical identity data with a third party—the Cloud Service Provider—outside of a company's

direct control raises fundamental questions about oversight and visibility, causing organizations to hesitate before committing their most sensitive identity information to the cloud.

One earlier research paper demonstrated a user successfully logging into a cloud storage service using pre-existing credentials from a trusted identity provider (IdP). This solution enhanced security by utilizing Authentication, Authorization, and Delegation APIs alongside a proxy-based IdP. The project leveraged the Trusted Architecture for Securely Shared Services (TAS3) framework to migrate services to the cloud. The implementation allowed for user federation and secure cloud access with delegation options, meaning users could temporarily grant others permission to access their data.

It also incorporated a Level of Assurance (LoA) system for making access decisions and partially implemented "break the glass" policies, which permit overriding access control during an emergency. The Authentication API used the SAML protocol to communicate with various configured IdPs, allowing the user to choose their preferred method. The Authorization API governed resource access, with access rules saved in a Policy Decision Point (PDP) database. The Delegation API operated via a Delegation Issuing Web Service (DIS) with an LDAP backend storing user delegation specifics. The main weaknesses of this previous work were in its trust and auditing components.

Access control systems are fundamentally designed to regulate who can access data on a system or physical assets. Using cryptography, a verifiable, secure, time-domain attribute-based access control (TAAC) scheme, combined with session keys, can be used to encrypt sensitive content like video. The Attribute-Based Cryptosystem (ABC) is an evolution of identity-based cryptosystems (IBC), offering more adaptability and flexibility. Instead of relying on a single identity, various user attributes are utilized to distinguish and classify a user. However, managing these attributes in a massive, interconnected system like the cloud presents significant challenges. Privacy and security researchers are actively working on how to maintain control over the disclosure and location of data privileges.

In the cloud, the objective of access control is to block unauthorized clients from accessing objects, thereby enhancing overall security. Companies are also developing sophisticated cloud auditing tools. For example, Microsoft offers SecGuru to examine Azure data centers using a set of policies integrated with the Z3 SMT solver. IBM provides a suite of monitoring tools integrated with QRadar. The MAC framework provides a high level of operating system security. Though earlier access control mechanisms existed for secure data access, they depended on system-level security and granted object access; even with an administrator present, defining the exact policy can be complex. With the emergence of highly advanced malware, such as Stuxnet, attackers began targeting exposed program entry points. Today, many major organizations rely on cutting-edge identity management systems like IdM4Cloud, Novell Identity Manager, and Microsoft Identity and Access.

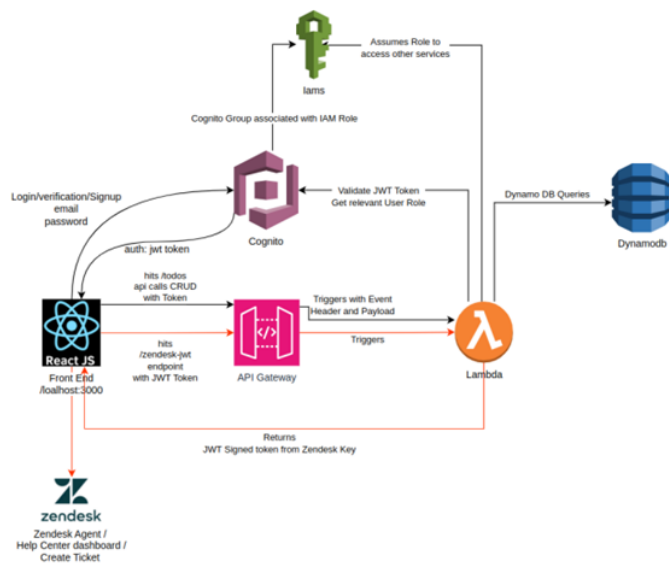


Figure 1. Overall System Architecture

5. Methodology

In this section, we will discuss the methodology that is used for the completion of Identity centric cloud security system. Its development consists of various phases ranging starting from user authentication to testing and validation of the authentication token.

5.1 Data Collection and Evaluation

Data for this study was collected from the system logs of an organization, providing real world insights into the operational environment. The logs were carefully analyzed to extract relevant security events and behavior patterns that could inform the evaluation of the proposed framework. The framework was then tested and evaluated in the Amazon Cognito Console, a cloud-based identity and access management service, to assess its performance and effectiveness in a controlled cloud environment. This provided a practical context to validate the framework's functionality within the security infrastructure of Amazon Web Services (AWS) [5], allowing for the assessment of key metrics such as security event handling, access control, and system response to simulated security incidents.

Identity and access management service, to assess its performance and effectiveness in a controlled cloud environment. This provided a practical context to validate the framework's functionality within the security infrastructure of Amazon Web Services (AWS) [6], allowing for the assessment of key metrics such as security event handling, access control, and system response to simulated security.

The methodology is explained step by step below:

It is managed through npm (Node Package Manager), which handles the installation of required packages.

- **User Authentication (AWS Cognito):** The process starts with user authentication managed by AWS Cognito. Users can sign up, log in, and verify their email ad-

resses. Upon successful login, Cognito issues a JSON Web Token (JWT) that serves as proof of the user's identity.

- **API Gateway Setup (Backend Interface):** An API Gateway is set up to handle incoming requests from the frontend (http requests). This gateway acts as a mediator between the React app and the backend services (Lambda functions), ensuring that requests are properly routed to the appropriate Lambda functions.
- **Lambda Function Implementation:** Lambda functions are created to handle the core logic of the application. These functions validate the JWT tokens received from the frontend and perform necessary operations, such as generating a Zendesk JWT token for user authentication with Zendesk.
- **Data Management with DynamoDB:** The Lambda functions interact with DynamoDB to store and retrieve user-specific data, such as to-do items. This ensures that user data is securely managed and easily accessible.

GET: Handles different GET requests based on the presence of id or user-id in the query parameters. Calls appropriate functions (get-todo-item-by-id, get-todo items-by-user-id, get-all-todo-items).

POST: Creates a new todo item using create-todo-item.

PUT: Updates an existing todo item using put-todo-item.

DELETE: Deletes a todo item by ID using delete-todo-item.

- **Zendesk Integration:** A new endpoint is added to the API Gateway specifically for generating a Zendesk JWT token and is also responsible for Single Sign On. The Lambda function signs the user information and returns the Zendesk JWT token to the frontend.

A new "/zendesk-jwt" endpoint is added to the API Gateway to handle requests from the frontend for generating a JWT token to authenticate with Zendesk.

JWT Token: The Lambda function generates a Zendesk JWT token by signing the user information (like email and ID) using the Zendesk shared secret.

- **Zendesk-Shared-Secret:** This is the secret key used to sign the JWT for Zendesk. It ensures that the token is signed securely using the "HS256" algorithm.
- **generateZendeskJwt(user):**

user: This object contains the user's information, typically including their name, email and ID.

Payload Construction:

name: The user's name.

email : The user's email address.

external id: This field represents the user's ID.

It is used as a unique identifier for the user in the Zendesk system.

ReactJS Flow: When users on the frontend (ReactJS) want to access Zendesk features.

- **Frontend Interaction:** The React frontend makes API calls to the newly created end point to obtain the Zendesk

JWT token. Once received, the frontend redirects users to the Zendesk help center, where they can access support features without needing to log in again.

- **Testing and Validation:** The entire setup is tested to ensure that the authentication flow works correctly, and users can access both the application and Zendesk services without issues. This includes checking the validity of JWT tokens and ensuring secure data handling.

The major challenge presented by current cloud access methods is the constant necessity for users to re-enter login details when accessing multiple services, even across different cloud providers, which significantly heightens the risk to the user's online identity. To mitigate this, there is a pressing need for a centralized Identity Management (IDM) service that securely connects users across various providers. This system proposes to analyze the digital trail left by user transactions to build a quantifiable "trust" level, offering cloud service providers the added value of improved identity security, the ability to establish trust with providers not directly integrated into the IDM, and reinforced assurance for highly sensitive transactions.

1) IAM Policies

The paper sets up various IAM (Identity and Access Management) policies set up in the AWS console [7]. Here are the key points regarding IAM policies:

- **Admin Role Policy:** This policy grants broad permissions for administrative tasks, allowing interactions with Cognito identity pools and user groups, full access to DynamoDB, and permissions for Lambda functions, S3 (Simple Storage Service), and API Gateway [8]. It is intended for administrators who need extensive access across AWS services.
- **Admin Role Trust Policy:** This specifies who can assume the Admin role, allowing only authenticated users from a specific Cognito identity pool and a designated IAM role to assume this role. It includes conditions that ensure only authorized entities can access sensitive operations.
- **Basic Role Policy:** This provides limited permissions for regular users, allowing them to get credentials for identity, invoke specific Lambda functions, and have read-only access to the DynamoDB table. This policy is designed for users with restricted access compared to administrators.
- **Basic Role Trust Policy:** Similar to the Admin Role Trust Policy, this defines who can assume the Basic role, ensuring that only authenticated users from a specific Cognito identity pool can access it.

In cloud computing, IAM policies are essential for managing and securing user access to resources. These policies define roles and permissions, assigning specific access levels to users and groups, and applying rules directly to resources. They ensure that users have appropriate access based on their roles and needs, control actions they can perform, and enforce conditions like multifactor authentication (MFA). By implementing well-defined IAM policies, organizations can efficiently manage access, enhance security, and maintain

compliance across their cloud environments.

2) Identity Management Infrastructure-IDaaS

The core idea behind our new Identity Management (IDM) system is to make moving around a federated cloud environment easy and safe for users. Our main goal is to get rid of the annoying requirement for users to create a brand new set of login details for every single cloud service they want to use. To achieve this, we have implemented an Identity-as-a-Service (IDaaS) model powered by AWS Cognito. This robust platform gives us a complete identity management solution, expertly handling everything from proving a user's identity (authentication) and granting them permissions (authorization) to general user account maintenance. By taking this approach, we can guarantee strong security and compliance standards while completely freeing users from the headache of managing countless usernames and passwords. Ultimately, our entire framework is built with security, privacy, and user trust at its foundation, forming what we call a "trusted identity ecosystem."

6. Results and Discussion

The paper successfully implemented secure user authentication using AWS Cognito, ensuring that only authorized users can access resources. The authentication process involves the following steps:

- **User registration:** Users register with the application using their email address and password.
- **User verification:** Users verify their email address by clicking on a verification link sent by AWS Cognito.
- **User authentication:** Users authenticate with the application using their email address and password.

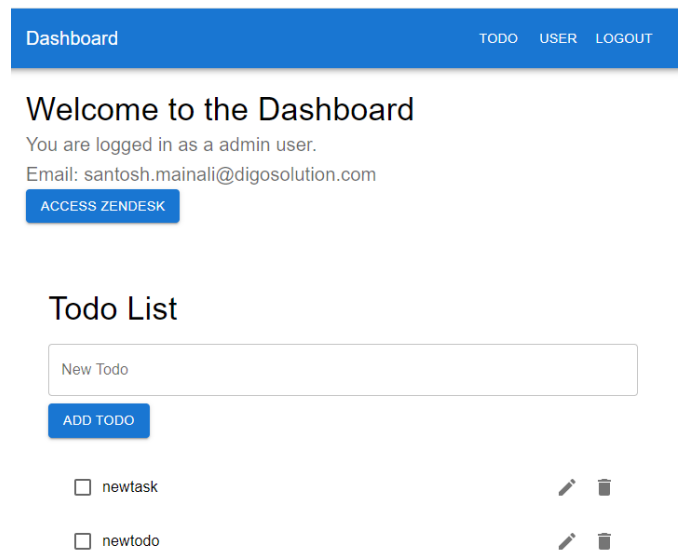


Figure 2. Snapshot of Token Representation After Logging to Zendesk

- **Role-Based Access Control:** The paper demonstrated the effectiveness of using role based access control to manage access to resources, ensuring that users can only

perform actions that are authorized by their role. The paper defines two roles:

Admin: This role has full access to all resources and can perform all actions.

Basic: This role has limited access to resources and can only perform specific actions.

Scalability: The paper showed that the use of AWS services, such as AWS Lambda and AWS API Gateway, can provide a scalable and flexible architecture for identity centric cloud security.

Timestamp	Message
	No older events at this moment. Retry
2024-09-24T12:13:08.011Z	INIT_START Runtime Version: nodejs120.v35 Runtime Version ARN: arn:aws:lambda:us-east-1:123456789012:runtime-version:nodejs120.v35
2024-09-24T12:13:08.043Z	START RequestId: 4443f264-3c0f-4a62-8330-3016350d9ef2 Version: SLATEST
2024-09-24T12:13:08.045Z	2024-09-24T12:13:08.045Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Lambda function invoked
2024-09-24T12:13:08.058Z	2024-09-24T12:13:08.058Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Event: {"resource":"/zendesk-ss..."} 2024-09-24T12:13:08.058Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Token Authentication Started
2024-09-24T12:13:08.058Z	2024-09-24T12:13:08.058Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Checkpoint 1
2024-09-24T12:13:08.058Z	2024-09-24T12:13:08.058Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Checkpoint 2
2024-09-24T12:13:08.058Z	2024-09-24T12:13:08.058Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Checkpoint 2.1
2024-09-24T12:13:08.059Z	2024-09-24T12:13:08.059Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Decoded Header: {"header":{"id..."/>
2024-09-24T12:13:08.059Z	2024-09-24T12:13:08.059Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Checkpoint 2.2
2024-09-24T12:13:08.059Z	2024-09-24T12:13:08.059Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO 3WS URL: https://cognito-idp.e.../...
2024-09-24T12:13:08.059Z	2024-09-24T12:13:08.059Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Checkpoint 2.3
2024-09-24T12:13:08.097Z	2024-09-24T12:13:08.097Z 4443f264-3c0f-4a62-8330-3016350d9ef2 IIFO Checkpoint 2.4

Figure 3. Live Tails of Users Accessing the System at Particular Time

6.1 Summary of tools

To further detail the technical implementation that maps the system's functional requirements to the specific AWS services used, defining their role in the overall Identity-Centric Cloud Security Framework, summary of tools are listed below:

- **AWS Cognito:** Centralized user directory for sign-up, sign-in, and issuing secure JSON Web Tokens (JWTs).
- **AWS Lambda :** Executes the serverless code to implement the Role-Based Access Control (RBAC) policies and generate the specific Zendesk SSO Token.
- **AWS API Gateway :** Provides a secure, scalable RESTful entry point for the front-end application to access the Lambda logic.
- **AWS DynamoDB :** Highly available NoSQL database used for storing application metadata, user-role mapping, and configuration.
- **Zendesk :** The service provider integrated via Single Sign-On (SSO) using the token generated by the system.

Acknowledgements

Sincere gratitude is extended to the supervisor for his unwavering guidance and support throughout the entire research process. His feedback has played a crucial role in shaping the direction and success of this research. The continuous encouragement and patience shown at every stage have been indispensable in overcoming various challenges and in maintaining focus throughout the study.

The valuable discussions and thoughtful advice provided have greatly enhanced both the quality of the research and the intellectual growth of the author. It is through the mentorship that this work has reached its completion. The contribution of his time and effort is deeply appreciated.

References

- [1] S. A. Almulla and C. Y. Yeun, "Cloud computing security management," in *2010 Second International Conference on Engineering System Management and Applications*, 2010, pp. 1–7.
- [2] E. Samlinson and M. Usha, "User-centric trust based identity as a service for federated cloud environment," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–5.
- [3] *A Security Framework for Cloud Manufacturing*, ser. International Manufacturing Science and Engineering Conference, vol. Volume 1: Materials; Micro and Nano Technologies; Properties, Applications and Systems; Sustainable Manufacturing, 06 2014. [Online]. Available: <https://doi.org/10.1115/MSEC2014-4082>
- [4] 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, pp. 163–174.
- [5] M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, "Serverless execution of scientific workflows: Experiments with hyperflow, aws lambda and google cloud functions," *Future Generation Computer Systems*, vol. 110, pp. 502–514, 2020.
- [6] F. Turan, S. S. Roy, and I. Verbauwhede, "Heaws: An accelerator for homomorphic encryption on the amazon aws fpga," *IEEE Transactions on Computers*, vol. 69, no. 8, pp. 1185–1196, 2020.
- [7] S. O. Olabanji, O. O. Olaniyi, C. S. Adigwe, O. J. Okunleye, and T. O. Oladoyinbo, "Ai for identity and access management (iam) in the cloud: Exploring the potential of artificial intelligence to improve user authentication, authorization, and access control within cloud-based systems," *Authorization, and Access Control within Cloud-Based Systems (January 25, 2024)*, 2024.
- [8] N. Papageorgopoulos, D. Vergeti, E. Politi, D. Ntalaperas, E. Tsimoni, and X. S. Papageorgiou, "Securing an agri-food marketplace: An implementation of a robust security layer with api gateway integration," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2023, pp. 2145–2150.
- [9] L. Lamport, *LaTeX: A document preparation system, User's guide and reference manual*, 2nd ed. Addison-Wesley Professional, 1994.
- [10] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl, "The not so short introduction to latex 2_ε," 2001.
- [11] H. Kopka and P. Daly, "A guide to {LaTeX}—document," 1995.
- [12] *LaTeX Tutorials – A Primer*, Indian TeXUsers Group.
- [13] P. Donegan, "Building security into csp cloud transformation strategies," *International Journal of Artificial Intelligence & Applications*, pp. 02–06, 2017.
- [14] G. Batra, V. Atluri, J. Vaidya, and S. Sural, "Deploying abac policies using rbac systems," *Journal of computer security*, vol. 27, no. 4, pp. 483–506, 2019.
- [15] T. S. Sobh, "Identity management using saml for mobile clients and internet of things," *Journal of High Speed Networks*, vol. 25, no. 1, pp. 101–126, 2019.
- [16] J. Basney, P. Cao, and T. Fleury, "Investigating root causes of authentication failures using a saml and oidc observatory," in *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*. IEEE, 2020, pp. 119–126.
- [17] L. Kumar, "Risk mitigation in cloud-based identity management systems: Best practices."