Secure Access Management in Serverless Computing Through Blockchain Integration

Swarup Panda

Email: *swaruppanda1331[at]gmail.com*

Abstract: ***Serverless architecture is increasingly used by companies and organizations to reduce costs and enhance scalability. However, the difficulty of performing instantaneous and accurate incident analysis remains a major challenge, as the number of incidents and threats continues to rise. Currently, most serverless platforms rely on centralized Identity and Access Management (IAM) to control system access. This reliance raises concerns regarding the inability to audit access events and the potential impact of IAM failures on other services.***

Keywords: Serverless Security, Blockchain, Access Control, Decentralized Identity, Smart Contracts

1. Introduction

To address these issues, an architectural design proposed that leverages blockchain distributed ledger technology to enable machine-to-machine communication and support decentralized access control in serverless computing. The design integrates blockchain and smart contracts to provide an auditability capability that overcomes the limitations of existing centralized access control systems [1],[2].

2. Background and Motivation

Serverless architecture enables developers to build and run applications without managing the underlying infrastructure. These event-driven and fully managed systems are adopted by all major cloud providers, including Amazon Web Services, Google Cloud Platform, Microsoft Azure, IBM Cloud, Oracle Cloud, Cloudflare Workers, and Alibaba Cloud. Data science, finance, IoT, and gaming are some of the key application domains.

Access control is an essential security mechanism in cloud environments [3]. A centralized identity and access management system (IAM) is commonly used to authorize users and serverless functions. It consolidates the entire access control and is the sole source of truth. Many risks and challenges arise from centralization. The IAM becomes a single point of failure. The centralized authority may suffer from a wide range of system disruptions and cyberattacks, including running out of storage capacity, misconfiguration, correlation of access records, and denial of service [4]. Centralized internal and external identity providers can be bottlenecks for global serverless architecture. IAM's lack of rapid scalability results in high costs and poor performance.
Auditability is another critical security challenge. Serverless architecture alters the threat landscape of cloud environments. The shift from application security to third-party dependencies creates complex and unreproducible vulnerabilities. The statelessness and ephemeral lifespan of serverless functions complicate forensics. Tamper-evident audit logs are desirable to detect anomalous access patterns and identify malicious actors. No effective solutions exist to satisfy these requirements when access control is centralized and proprietary.

3. Serverless Architectures Overview

Serverless architectures rely on third-party cloud service providers to execute applications event-driven, on-demand, and on a pay-as-you-go basis [5]. The provider manages scalability and fault-tolerance, contributing to the software and operations productivity gains that are the main motivations pushing organizations to adopt serverless architectures. The traditional security perimeter model changes significantly in serverless environments. Some characteristics, such as the usage of ephemeral computing units triggering additional events on other parts of architecture, make existing security solutions inefficient in serverless architectures. Hence, new security adjustments are necessary to address these architectural differences.

4. Current Access Control Mechanisms

The term serverless computing denotes highly dynamic execution environments where programmers deploy business functionalities and delegate the management of server resources to a third party [6]. The significant security improvements in these architectures stem from the reduction of the attack surface of server management and the adoption of fine-grained access control mechanisms that are not natively available in serverful architectures. Among them, the problem of minimizing the lateral movement of attackers (the sequential compromise of additional infrastructure components using the already compromised ones) is tackled in the Azure SEAL architecture by the adoption of ephemeral credentials [7]. Currently, access control policies are exclusively expressed via a centralized identity and access management (IAM) system owned and controlled by the cloud vendor, which poses the risk of a single point of failure and the related dependence on the single vendor capabilities [8]. Such extreme centralization also hinders auditability of authorization decisions.

5. Limitations of Centralized IAM

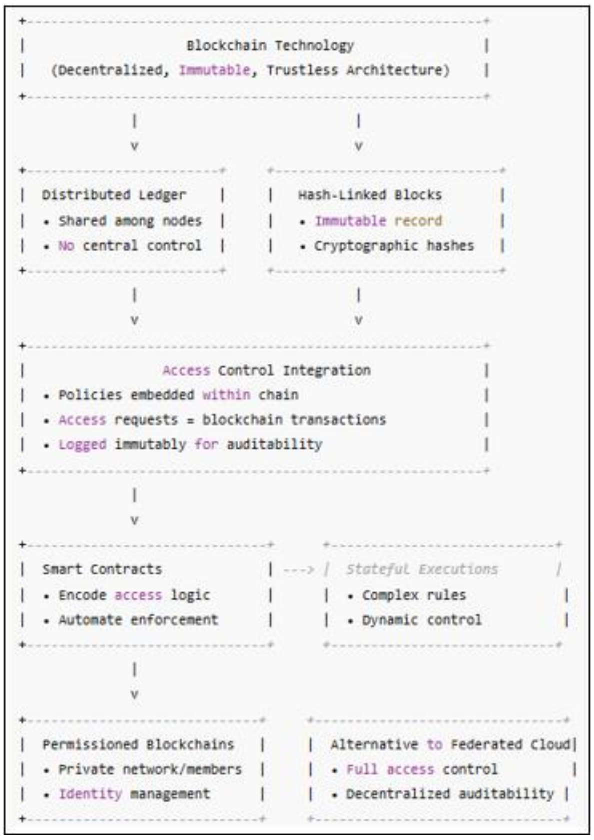
Centralized IAM systems are typically not designed to provide a transparent and irrefutable record of authorization operations. Almost all access control proposals for serverless rely on centralized IAM providers. Yet a centralized approach cannot generally satisfy the serverless mandate for minimal

DOI: https://dx.doi.org/10.21275/SR25804083648

management and operational overhead. Centralization creates a single point of failure and a scalability bottleneck for the entire system. Furthermore, serverless functions by design are stateless—any granted access rights must be checked separately. As a result, clients first obtain a JSON Web Token (JWT) from the identity provider and then attach the token with each serverless request [9] [10].

6. Blockchain Technology Fundamentals

Blockchain is a distributed ledger technology enabling the trustless exchange of digital assets without intermediaries [11]. Decentralized systems record data in immutable blocks connected by hash functions. Active research focuses on access control in multiuser blockchain environments. In cloud contexts, access control policies can be encapsulated within the chain, with access requests authorized through blockchain transactions which are then logged for auditability [12]. The blockchain network architecture permits stateful contract executions, enabling complex logic that extends beyond traditional access schemes. Permissioned blockchains with private membership manage identity and access, serving as an alternative to federated cloud architectures where organizations retain full control over access control and auditing without relying on central authorities [13].



```
+--------------------------------------------------------+
|                   Blockchain Technology                |
|  (Decentralized, Immutable, Trustless Architecture)    |
+--------------------------------------------------------+

              |                       |
              v                       v

+-------------------------+    +--------------------------+
| Distributed Ledger      |    |  Hash-Linked Blocks      |
| • Shared among nodes    |    |  • Immutable record      |
| • No central control    |    |  • Cryptographic hashes  |
+-------------------------+    +--------------------------+

              |                       |
              v                       v

+--------------------------------------------------------+
|                Access Control Integration              |
| • Policies embedded within chain                       |
| • Access requests = blockchain transactions            |
| • Logged immutably for auditability                    |
+--------------------------------------------------------+

              |
              v

+-----------------------------+    +---------------------------+
| Smart Contracts             | ---> |  Stateful Executions    |
| • Encode access logic       |    |  • Complex rules          |
| • Automate enforcement      |    |  • Dynamic control        |
+-----------------------------+    +---------------------------+

              |
              v

+----------------------------+     +---------------------------+
| Permissioned Blockchains   |     | Alternative to Federated Cloud|
| • Private network/members  |     | • Full access control     |
| • Identity management      |     | • Decentralized auditability |
+----------------------------+     +---------------------------+
```

Figure 1: Blockchain Technology Fundamentals – Overview Diagram

7. Decentralized Identity Concepts

The development of decentralized identity concepts has been driven by the need to reduce the reliance on centralized cloud identity providers [14]. The federated access control paradigm is used to evaluate the attributes of a requestor within the authentication context and verify the requestors' credentials. However, with attributes available from multiple information sources, each with different levels of verification and trust, generating trustable claims requires careful assessment. Verification of verification claims is necessary to ensure the validity of authentication attributes. Blockchains enable an intuitive construction of identity solutions, as illustrated in previous developments to replace centralized identity providers and to facilitate the linkage of human and device identities with blockchain [15]. The decentralized nature encompassed in blockchain-enabled identity solutions allows users to provide proof of their credentials to service providers without the need for an intermediary or the necessity to disclose the underlying data.

In addition to decentralization, identity solutions must provide discoverability so that anyone can confirm the association between an identity and blockchain accounts. Once association is established, verification of the signature on a given message is straightforward, enabling an easy proof-of-ownership mechanism that facilitates authorization decisions and plays an important role in developing decentralized identity and access control solutions. The concepts of decentralized identity then underpin the introduction of architectures where smart contracts take a central role in the access-control process, bridging serverless applications and blockchain.

8. Smart Contracts for Access Control

Access control in serverless computing could be greatly improved by leveraging blockchain-based smart contracts. Smart contracts are executable logic deployed on a blockchain. They can be used to specify sophisticated access-control policies, to verify that resource requests satisfy these policies, and to grant or deny access accordingly [16].

A blockchain-based architecture for access control reduces dependency on centralized Identity and Access Management (IAM) services, which are single points of failure, scalability bottlenecks, and single trust anchors. Decentralized identity provides unforgeable proof that a digital identity belongs to a given entity and enables minimally disclosed proofs of identity to verify that the entity satisfies one or more access control capabilities. Smart contracts implement access control logic. With decentralized identities and smart contracts, access control can be enforced without relying on centralized IAM components.

9. Proposed Architecture

Contemporary serverless architectures are based on Function-as-a-Service (FaaS), a paradigm where developers author applications using cloud provider-supplied functions that are triggered by events. Despite the growth in adoption, security and access control remain poorly researched in the community. Contemporary access control approaches rely on a centralized identity and access management (IAM) system, introducing a single point-of-failure, bottleneck, and management overhead. This work leverages decentralized and immutable blockchains to enhance the security posture of serverless architecture by proposing a novel architecture for blockchain-based access control. Serverless functions communicate with an authorization smart contract deployed to a blockchain to determine whether a requester has sufficient privileges to invoke a specific function. The architecture enables a highly scalable, fault-tolerant, and strongly audited mechanism for access control that substantially reduces reliance on centralized IAM. It is typified by very low request latency, remains practically scalable, and facilitates cross-platform open access adoption without requiring organizations to migrate all applications to a single cloud provider [17] [18].

An identity provider (IdP) is responsible for off-chain identity provisioning and management. When a user requests to invoke a serverless function, the request is forwarded to the function, which then queries an authorization smart contract. After receiving a valid authorization taken from the smart contract, the function verifies the requester's identity, and the signature associated with the authorization taken through a verification smart contract before executing the function.

9.1 System Components

The architecture enables fine-grained access control of serverless functions and facilitates auditing. The system comprises three components: an identity provider, a blockchain access control smart contract, and service providers that host the serverless functions supporting access control enforcement [19]. The identity provider issues cryptographically signed identities for access-validation purposes. The smart contract operates as the core access-control layer, used to encode access-control policies and carry out the access-control evaluation process. The service providers manage serverless functions [20].

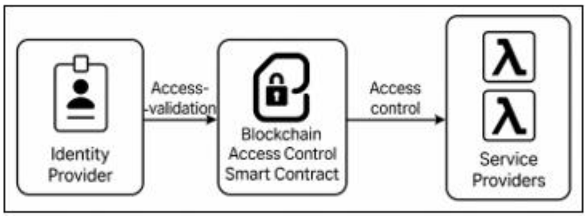


Figure 2: System components

9.2 Data Flow

Figure 3. depicts the interactions involved when a user requests access to a cloud resource. As indicated in earlier sections, the user initially authenticates with an identity management service to obtain an access token. This token is then attached to the request, which is directed at a serverless function hosted on a cloud platform. Upon receipt, the serverless function verifies the access token with the identity service and retrieves the associated public key to validate the request signature. Assuming the signature is confirmed, the function queries a smart contract on a permissioned blockchain to determine the user's permissions. If the smart contract grants access, the serverless function proceeds to invoke the requested cloud resource. This process ensures that access verification is decentralized, reducing reliance on centralized identity providers, and provides an immutable audit trail through the blockchain [21] [22].

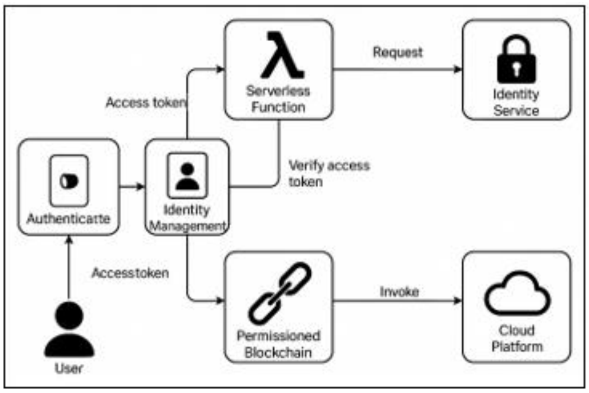


Figure 3: Interactions involved when a user requests access to a cloud resource

10. Implementation Considerations

The realizability of the proposed blockchain-based access control architecture depends on the deployment of several components that are widely adopted in the industry, and which are supported by standard development tools across the cloud provider landscape [23]. The identity provider component can be implemented with any service offering cryptographically verifiable credentials, including decentralized identity platforms [24]. Multiple blockchains enable smart contracts with arbitrary on-chain programmed logic [25]. Trusted platforms capable of verifying that the code executing off chain is authentic allow an off-chain service trusted by the smart contract to intervene in the authorization process.

10.1 Technology Stack

Table 10.1 summarizes the technology stack. Blockchain platforms for decentralized access control comprise Ethereum and Hyperledger Fabric. Ethereum serves as an open-source, public blockchain enabling smart contract implementation in Turing-complete programming languages. Smart contracts define relationships between blockchain users and manage ledger entries according to business rules. Hyperledger Fabric, an open-source private permissioned blockchain framework, facilitates developing modular applications with configurable consensus and membership services. Table 10.2 compares these two platforms.

Serverless components include platforms and frameworks such as Amazon Web Services (AWS), OpenWhisk, and Chalice. AWS offers cloud infrastructure for deploying autonomous functions. OpenWhisk, an open source serverless platform, executes actions in response to events within a distributed cloud environment. Chalice, a Python serverless microframework, enables rapid creation and deployment of applications integrated with AWS Lambda.
The development environment incorporates Visual Studio Code, the AWS Command Line Interface (CLI), the Solidity compiler (Solc), the Node Package Manager (NPM), and Python. Visual Studio Code provides a lightweight code editor and debugger accessible via various web browsers. The AWS CLI facilitates management of AWS services. Solc compiles Solidity source code into low-level machine-readable opcode targeting the Ethereum Virtual Machine.

NPM serves as a package manager for JavaScript. Python executes automation-related tasks.

Additional tools encompass Ganache, Captcha, and Metamask. Ganache is used to simulate a local blockchain network for testing and debugging smart contracts. Captcha distinguishes humans from robots to protect applications from bots. Metamask enables sending and receiving transaction requests and executing Ethereum smart contracts.

10.2 Integration Challenges

The decentralized nature of the blockchain paradigm still clashes with the centralized behavior of serverless architectures. The two systems rely on different mechanisms to control the interaction and the exchange of information between entities. Organizations must develop complex software bridges to enable cooperation between function spaces and blockchain environments. The accountability provided by the immutable features of blockchain comes at a non-negligible cost. Serverless functions orchestrating communication between blockchain contracts often require multiple reiterations when performing access control tasks, increasing the average latency of an operation.

11. Auditability in Decentralized Systems

In serverless architectures, auditability refers to the capacity to track and review logged requests and actions for accountability and forensic analysis. In centralized architectures, auditability is achievable through well-established cybersecurity mechanisms. Decentralized architectures lack a globally accepted trustworthy repository of logs, hindering these mechanisms. In blockchain-based systems, the adherence to public-key cryptography and the global replication of an append-only tamper-proof ledger facilitates transparent and tamper-proof logging of information aligned with the blockchain’s consistency assumptions [26]. Implementing all access control-related operations as transactions on a blockchain can enable the provision of a widely transparent and globally distributed tamper-proof log of requests.

12. Performance Evaluation

Efficient and transparent authorization is critical for access control in serverless applications. Replacing centralized data stores with a public permissionless blockchain can help achieve secure, difficult to tamper with authorization. To verify the practical feasibility of the proposed blockchain-based access control architecture, its performance characteristics were evaluated.

The evaluation examines scalability and latency aspects. Scalability considers how the system handles increasing numbers of users and access requests, while latency assesses the delay incurred during the authorization process. These metrics are key to ensuring that a blockchain-based access control solution complements serverless architectures without introducing significant overhead.

The following subsections present details for each aspect.

12.1 Scalability Analysis

Public blockchains impose significant constraints on the scalability of access control architecture. The ability to support many nodes and a high rate of transactions is a key factor for the success of blockchain-based access control. The architecture proposed above is intended to support a network of users and services of unlimited size and operating on access-control workloads of arbitrary complexity, while retaining the security and decentralization properties of the underlying blockchain. State-of-the-art blockchains face a fundamental scalability challenge due to the interplay of conflicting requirements. Vitalik Buterin summarized the problem with the blockchain trilemma: the more a blockchain improves scalability, security, or decentralization, the more it tends to compromise at least one of the other two. Most public blockchains address transaction scalability by broadcasting all transactions and blocks to all nodes in the network, thus causing a bottleneck related to the maximum processing power, storage, and bandwidth of individual nodes. Approaches that partition the network to keep a subset of nodes responsible for each transaction can improve throughput at the expense of reduced security and/or decentralization. The architecture above draws inspiration from a blockchain that removes the prevailing single-node bottleneck by distributing transaction validation and storage across many nodes organized in randomly selected committees, without compromising security or decentralization. The maximum workload that can be processed scales linearly with the number of nodes in the network, but no node processes data whose volume is proportional to the overall system workload. The scalability results of the blockchain were presented in the dynamic-adversary setting typical of permissionless environments, indicating suitability to the applications described above [27]. The cost of reading data records on the blockchain is not considered a system bottleneck. Instead, challenge production and result verification impose a far greater workload on clients verifying challenge parameters and access tokens, potentially slowing down such operations [1].

12.2 Latency Considerations

The proposed architecture integrates multiple components to establish a secure and scalable access control system for serverless environments. Identity provider (IdP) issues verifiable credentials that a client uses to obtain an access token from an authorization server. This token is employed to access a serverless function, which, in turn, verifies authorization by querying a permission smart contract deployed on a blockchain platform. The permission smart contract encodes access policies and conducts token verification processes, leveraging information about the client's identity encapsulated in the verifiable credentials. Incorporating an access-control system that combines serverless functions with blockchain technology significantly simplifies deployment and ensures scalability to handle millions of simultaneous access requests. Due to the complex integration of serverless and blockchain components, the overall system latency—measured by the client requested the serverless function’s response—serves as the primary performance metric for evaluation. To mitigate latency that could degrade user experience, such measurements guide optimizations aimed at preserving responsiveness and usability in the access control infrastructure [28] [29].

13. Security Implications

Threat models for decentralized access control must address attacks on direct data channels as well as on cloud functions and the blockchain platform. Direct data channels are vulnerable to interception, message injection, and denial of service. Compromised cloud functions may corrupt, leak, or misuse data or attempt to escalate privileges. Blockchain interactions are subject to Sybil and DoS attacks.

Mitigation depends on the attacker's capabilities and the resources targeted. Data-channel attacks can be remedied with traditional end-to-end encryption, while stronger adversaries require secure computation, secret sharing, trusted execution environments, or more robust logic obfuscation. Attacks on cloud functions are mitigated by minimizing their privileges without compromising performance and by leveraging the inherent non-repudiation, authentication, auditability, and provenance of decentralized ledgers. The blockchain's code and data are replicated by mutually distrustful parties, protecting smart contracts from corruption or deletion. Enforcing signatures on all transactions effectively prevents spoofing and Sybil attacks, whereas denial of service remains a fundamental risk.

The resulting mechanism constitutes a flexible, scalable, blockchain-based authorization system for serverless architectures. It enables fine-grained control without reliance on a centralized trusted authority or conventional heavyweight authentication and authorization protocols. Mechanisms for universal identity, verifiable claims, and reputation can be incorporated to mitigate or eliminate the need for impersonation and long-term secrets. All access-request decisions become publicly auditable, creating a valuable and cost-effective security and compliance tool with broad applicability beyond the cloud itself. [30] [31]

13.1 Threat Models

Although blockchain and other decentralized technologies have been proposed to support fine-grained access control in serverless architectures, the security implications of these approaches have yet to be studied in detail. At the same time, serverless computing continues to gain widespread adoption, increasing the urgency of examining these concerns. Serverless architectures host workloads or services on cloud infrastructure and execute functions on demand. These functions require access to the environment and any hosted services—raising the prospect of granting a potentially very large attack surface to untrusted or guest code and thus creating new opportunities for attackers to access or modify information that is otherwise off limits. Access control within serverless environments is therefore an enduring challenge, and one that is unlikely to be significantly diminished through increased adoption or usage [9]. The complexity arises from the need to maintain consistent, auditable, and secure control over data in a distributed and dynamic development landscape. The problem, therefore, is how to best ensure access control in serverless environments. This section presents the SMACS architecture, which combines serverless computing, distributed ledger technology (DLT), and decentralized identifiers (DIDs): a suite of complementary, emerging distributed technologies. SMACS reduces reliance on a centralized Identity and Access Management system, supports long-term auditability, leverages consumer credential wallets for privacy, and facilitates fine-grained Zero Trust control based on principals such as location or device integrity [1].

13.2 Mitigation Strategies

Centralized identity management (IdM) systems constitute a single point of failure and a scalability bottleneck [9]. There is a critical need to limit the reliance on third-party IdPs without sacrificing auditability. Distributed ledger technology (DLT) provides a reusable and extensible way of building cryptographically verifiable claims: both user claims and system-issued access-control claims can be stored in a decentralized ledger. Identity providers, administrators, and service providers can issue such claims to steer the system.

Blockchain technology and smart contracts are distributed systems that jointly provide a secure communication and computation substrate beyond the scope of traditional cloud infrastructure [1]. In a serverless setting, use of an Ethereum blockchain and a set of smart contracts enables an IdP-agnostic access-control mechanism. Decentralized architecture provides strong guarantees of transparency, traceability, and non-repudiation of access events.

14. Case Studies

To evaluate the generality and effectiveness of the aIaaS)h, the architecture was implemented in various use cases, including a Software as a Service (SaaS) platform providing Digital Ocean services for virtual machines, a racing game developed with the Phaser JavaScript framework in an OpenShift environment (also with the Digital Ocean provider operating Digital Ocean Infrastructure as a Service (IaaS)), and the DVB (Digital Video Broadcasting) sports live streaming platform that is already running in a traditional environment with centralized Identity and Access Management (IAM). These deployments confirm the architecture's broad applicability. Based on these case studies and carried-out experimental evaluations, the architecture can offer secure access control to high-frequency invocation serverless systems while mitigating the dependence on centralized IAM [6].

14.1 Real-World Applications

Successful real-world implementations verify the flexibility and security of the architecture. Several companies have adopted this approach for specific use cases. Furthermore, systematic evaluations of Amazon Web Services (AWS) IAM environments reveal general benefits compared to current practices [1].

14.2 Comparative Analysis

Centralized IAM services suffer from single points of failure and scalability bottlenecks. Blockchain technology underpins decentralized access control models by providing an

Paper ID: SR25804083648 DOI: https://dx.doi.org/10.21275/SR25804083648

immutable distributed ledger [1]. Decentralized identities represent frameworks that enable self-sovereign control over identity-related data, thus reducing reliance on central authorities [3]. Deploying smart contracts to represent access control policies offers programmable, automated enforcement mechanisms [6]. Categories: security and privacy; access control; serverless computing; blockchain.

15. Future Work

Decentralized access control based on blockchain can provide a promising way to enforce security-critical policies in serverless clouds and constitutes a building block for human-machine interaction and continuous delegations throughout chained executions. Future work will focus on extending the architecture to support more scalable blockchain implementation and a wider range of identities such as devices and services [2].

A growing number of cases show that many organizations worldwide, including major cloud providers, are embracing the serverless paradigm. Serverless-based applications are frequently developed through function composition, involving numerous mutually untrusted entities. Existing systems rely on centralized IAM with a single point of failure and scalability limitations, failing to provide reliable auditing and preventing continuous delegation. This paper introduces a blockchain-based access control architecture able to address these challenges [1]. The architecture comprises several components, including identity providers, smart contracts, and serverless functions. Data flow involves request authorization and verification, with smart contracts enforcing policies and maintaining auditability. Relevant implementation considerations encompass the choice of blockchain platforms and serverless frameworks, along with integration aspects. Performance evaluation addresses scalability analysis and latency considerations, assessing system efficiency. Security implications cover threat models and mitigation strategies, ensuring robustness. Case studies highlight real-world applications and provide a comparative analysis with centralized systems.

16. Conclusion

Serverless architecture allows developers to build and run applications without managing infrastructure, using frameworks like AWS Lambda, Google Cloud Functions, and Azure Functions. However, shortcomings of conventional access control mechanisms in serverless environments, particularly the dependence on centralized identity and access management (IAM) services and limited auditability—raise security and privacy concerns. The centralized IAM service acts as a single point of failure and must support numerous clients, while the default access-control policy passes the client's credentials through the service, limiting auditability.

This issue is addressed by proposing a novel architecture that leverages blockchain and smart contracts to achieve decentralized access control with serverless functions. The key idea integrates decentralized identity, multi-authority attribute-based access control, and policy delegation in a blockchain system. The architecture is implemented and evaluated on Hyperledger Fabric and AWS Lambda, showing superior auditability and performance in terms of scalability and latency compared to centralized IAM alternatives [29] [30].

References

[1] O. C. Ri, Y. J. Kim, and Y. J. Jong, "Blockchain-based RBAC Model with Separation of Duties constraint in Cloud Environment," 2022.

[2] H. Song, Z. Tu, and Y. Qin, "Blockchain-Based Access Control and Behavior Regulation System for IoT," 2022. ncbi.nlm.nih.gov

[3] Panda, Sibaram Prasad. "Securing 5G Critical Interfaces: A Zero Trust Approach for Next-Generation Network Resilience." 2025 12th International Conference on Information Technology (ICIT). IEEE, 2025.

[4] Panda, Sibaram Prasad. "Blockchain Technologies for Secure and Transparent Artificial Intelligence Systems." (2025).

[5] S. Rouhani, R. Belchior, R. S. Cruz, and R. Deters, "Distributed Attribute-Based Access Control System Using a Permissioned Blockchain," 2020.

[6] M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security Services Using Blockchains: A State of the Art Survey," 2018.

[7] M. Amine Bouras, B. Xia, A. Omer Abuassba, H. Ning et al., "IoT-CCAC: a blockchain-based consortium capability access control approach for IoT," 2021. ncbi.nlm.nih.gov

[8] S. Pancari, A. Rashid, J. Zheng, S. Patel et al., "A Systematic Comparison between the Ethereum and Hyperledger Fabric Blockchain Platforms for Attribute-Based Access Control in Smart Home IoT Environments," 2023. ncbi.nlm.nih.gov

[9] S. Gilda, T. Jain, and A. Dhalla, "None Shall Pass: A blockchain-based federated identity management system," 2022. [PDF]

[10] Priyambada Swain. “Cost-efficient AI Models for Real-time Big Data Analytics”. International Journal of Computer Application, vol. 15, no. 4, RSPUBLICATION, July 2025, pp. 118–44, doi:10.5281/zenodo.16436367.

[11] Priyambada Swain. “Comparative Study of Spark Mllib Vs. Tensorflow on Distributed Big Data Sets”. International Journal of Computer Application, vol. 15, no. 4, RSPUBLICATION, July 2025, pp. 145–67, doi:10.5281/zenodo.16437469.

[12] D. Basile, C. Di Ciccio, V. Goretti, and S. Kirrane, "A Blockchain-driven Architecture for Usage Control in Solid," 2023. [PDF]

[13] B. Liu, S. Sun, and P. Szalachowski, "SMACS: Smart Contract Access Control Service," 2020.

[14] S. P. Panda, Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing, 2025.

[15] S. P. Panda, Artificial Intelligence Across Borders: Transforming Industries Through Intelligent Innovation. Deep Science Publishing, 2025.

[16] U. Ugobame Uchibeke, S. Hosseinzadeh Kassani, K. A. Schneider, and R. Deters, "Blockchain access control Ecosystem for Big Data security," 2018.

Paper ID: SR25804083648 DOI: https://dx.doi.org/10.21275/SR25804083648

[17] Panda, Swarup. "'Kubernetes in AWS (EKS): Enhancing DevOps Workflow Efficiency.'" International Journal of Computer Application, 2025.

[18] Panda, Swarup. "Optimizing Electric Vehicle Routing: A Statistical Analysis of ACO, GA, and SA Algorithms." International Journal of Science and Research, 2025.

[19] Muppala, Mohanraju. "Digital Oceans - Artificial Intelligence, IoT, and Sensor Technologies for Marine Monitoring and Climate Resilience." Deep Science, 2025.

[20] Muppala, Mohanraju. "Artificial Intelligence and Machine Learning in Maritime Shipping."

[21] Koneti, Subramanya Bharathvamsi, et al. "Explainable AI in Healthcare: Bridging the Gap Between Accuracy and Interpretability." International Journal of Emerging Trends in Engineering and Development, 2025.

[22] T. Sylla, L. Mendiboure, M. Aymen Chalouf, and F. Krief, "Blockchain-Based Context-Aware Authorization Management as a Service in IoT," 2021.

[23] G. Del Monte, D. Pennino, and M. Pizzonia, "Scaling Blockchains Without Giving up Decentralization and Security," 2020.

[24] SHIVADEKAR, SAMIT. "Secure Multi-Tenant Architectures in Microsoft Fabric: A Zero-Trust Perspective." (2025).

[25] Shivadekar, Samit. "AI for Climate Change and Sustainability."

[26] Z. Luo, R. Murukutla, and A. Kate, "Last Mile of Blockchains: RPC and Node-as-a-service," 2022.

[27] K. Alpernas, C. Flanagan, S. Fouladi, L. Ryzhyk et al., "Secure Serverless Computing Using Dynamic Information Flow Control," 2018. [PDF]

[28] S. P. Panda, Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing, 2025.

[29] S. P. Panda, Artificial Intelligence Across Borders: Transforming Industries Through Intelligent Innovation. Deep Science Publishing, 2025.

[30] Panda, S. P., Koneti, S. B., & Muppala, M. (2025). Benefits of Site Reliability Engineering (SRE) in Modern Technology Environments. *Available at SSRN 5285768*.

[31] Partha Sarathi Mohapatra, "Integrating AWS with SQL Databases", International Journal of Research in Engineering & Science ISSN:(P) 2572-4274 (O) 2572-4304, vol. 9, no. 2, pp. 35–59, May 2025, doi: 10.5281/zenodo.15488890

[32] Mohapatra, P. S. (2025). Integrating AWS with SQL Databases. RSPUBLICATION.

[33] U. Ugobame Uchibeke, S. Hosseinzadeh Kassani, K. A. Schneider, and R. Deters, "Blockchain access control Ecosystem for Big Data security," 2018.