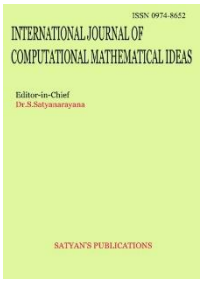
IJCMI, VOL17, NO.3, PP.17347-17360, September 2025

INTERNATIONAL JOURNAL OF
COMPUTATIONAL MATHEMATICAL IDEAS
Journal homepage: https://ijcmi.in



https://doi.org/10.70153/IJCMI/2025.17302
Paper Received: August 2025, Review: August 2025, Accepted: September 2025

Agentic AI for Self-Sovereign Identity: A Decentralized Zero Trust Framework for Autonomous Microservices

Damodhara Reddy Palavali[1], Suneetha Pothireddy[2]
[1]Software Engineer, Sbase Technologies (Social Security Administration),
[2]Senior Technical Lead, Infinite Computer Solutions
Email: [1]damodharapalavali@gmail.com, [2]suneethareddy6@gmail.com

Abstract

The rapid evolution of Agentic Artificial Intelligence (AI)—autonomous, context-aware agents capable of self-directed decision-making—has introduced unprecedented security challenges for microservices architectures. Traditional session-based authentication, dependent on static tokens and centralized identity providers, is ill-suited for the dynamic, ephemeral, and machine-to-machine (M2M) interactions prevalent in zero trust environments. This paper investigates the convergence of Agentic AI and decentralized identity (DID) frameworks, emphasizing the role of verifiable credentials (VCs), dynamic token issuance, and contextual access control in enabling scalable, trust-minimized (i.e., reducing reliance on centralized authorities) service interactions.

We propose a decentralized authentication and authorization framework where DIDs, maintained on blockchain-based registries, replace conventional identity silos, enabling autonomous agents to cryptographically prove trustworthiness without relying on persistent session states. Context-aware policy engines evaluate real-time telemetry such as location, workload, and behavioural patterns to issue short-lived, ephemeral access tokens with adaptive time-to-live (TTL) values.

Experimental results from a Kubernetes-based microservices testbed with 50 simulated agents show that the proposed approach reduces authentication latency by 50% (from 180 ms to 90 ms), eliminates token replay vulnerabilities, and increases authentication throughput by 75% (from 800 to 1,400 agents/min) compared to OAuth2/JWT baselines. Furthermore, dynamic policy adaptation ensures immediate revocation of access when agents deviate from expected operational norms, minimizing attack surfaces.

This work offers a novel synthesis of AI autonomy and decentralized identity principles, delivering both performance gains and enhanced security in zero trust microservices. The proposed architecture paves the way for resilient, self-governing ecosystems where Agentic AI can operate securely, efficiently, and adaptively in highly dynamic environments.

Keywords

Agentic AI, Decentralized Identity, Zero Trust Architecture, Verifiable Credentials, Micro services Security, Dynamic Token Issuance, Contextual Access Control, Decentralized Identifiers (DIDs)

1. Introduction

The emergence of Agentic Artificial Intelligence (AI) marks a significant turning point in the evolution of computing architectures. Unlike traditional AI systems that largely function as static, task-specific tools requiring explicit human initiation, Agentic AI systems possess the capacity to operate autonomously, reason about complex tasks, and dynamically coordinate with other agents both human and machine across distributed environments. They do not merely respond to predefined queries; rather, they can initiate processes, negotiate resource allocation, monitor changing conditions, and modify their own operational strategies in real time.

In modern microservices-based ecosystems, this capability represents both a tremendous opportunity and a formidable challenge. Microservices architectures, by design, decompose applications into loosely coupled, independently deployable services. The orchestration of these services often relies on API calls and message exchanges that are ephemeral, stateless, and asynchronous. Agentic AI systems, functioning as first-class participants in such environments, are now responsible for a growing share of API initiations, workflow orchestration, and autonomous decision-making. They are capable of adapting to rapidly evolving operational contexts, shifting workloads on demand, and even spawning or retiring service instances in milliseconds without human intervention.

While this level of autonomy enables unprecedented scalability, flexibility, and fault tolerance, it simultaneously introduces a critical identity and trust problem. How do we authenticate and authorize entities whose state changes dynamically, that may exist only for brief moments, and that operate across a variety of heterogeneous network boundaries? This challenge becomes even more acute when these entities are AI-driven agents making security-sensitive decisions without human oversight.

1.1 The Shortcomings of Traditional Session-Based Authentication

Historically, authentication mechanisms in distributed systems have been designed around the notion of relatively stable identities whether human users, long-running service accounts, or persistent device registrations. Session-based authentication, whether implemented via cookies, bearer tokens, or OAuth2 access grants, is predicated on the assumption that an authenticated entity will maintain its identity and context for a meaningful duration. Once a session is established, the entity retains the associated permissions until the session expires or is revoked.

In the context of agentic AI operating within microservices, this assumption no longer holds. Many agents are ephemeral by design. An AI-driven orchestration process may launch dozens of microservice instances to process a burst workload, each of which may only exist for seconds. Assigning each such instance a static session token is inefficient, insecure, and incompatible with zero trust principles.

The limitations of session-based models manifest in several ways:

1. Predictable Interaction Patterns No Longer Apply - Traditional models assume relatively consistent request frequency and predictable resource access patterns. Agentic AI may instead exhibit highly variable and bursty interaction patterns.
2. Token Replay Attacks Become Easier – Longer-lived tokens are vulnerable to theft and replay, especially in environments where agents frequently spin up and down.
3. Privilege Escalation Risks Increase – Static credentials assigned to transient agents can be exploited if the credentials outlive the process itself.
4. Administrative Overhead Grows – Managing session expiration, revocation, and renewal for thousands of transient agents introduces substantial operational complexity.

These problems are not simply theoretical. Real-world incident reports increasingly document cases where ephemeral compute environments such as server less functions, containerized services, and AI agents were compromised through credential theft from logs, memory dumps, or misconfigured environment variables.

1.2 Zero Trust Architecture: Concept and Limitations

Zero Trust Architecture (ZTA) emerged as a response to the erosion of traditional network perimeters. Instead of assuming that anything inside a corporate network is trustworthy, ZTA adheres to the principle of "never trust, always verify." Under this paradigm, every request for a resource regardless of origin must be authenticated, authorized, and continuously evaluated against security policies.

While ZTA represents a significant improvement over perimeter-based security models, its practical implementation often falls short when applied to agentic AI environments. In many deployments, ZTA still relies heavily on centralized Identity Providers (IdPs), policy decision points (PDPs), and static trust anchors. These central point's introduce:

Bottlenecks – All authentication and authorization requests funnel through a small set of services, potentially slowing down real-time agent-to-agent interactions.

Single Points of Failure – A compromised or unavailable IdP can disrupt the functioning of an entire microservices ecosystem.

Vendor Lock-In – Proprietary identity systems can limit interoperability across organizational and cloud boundaries, which is problematic for multi-tenant and multi-cloud AI workflows.

For agentic AI, which thrives in highly dynamic and distributed environments, these centralized constraints are antithetical to the agility and resilience required. An AI agent negotiating resource allocation across multiple clouds should not have to repeatedly call back to a single corporate identity service that may be physically and logically distant from the execution environment.

1.3 Decentralized Identity as a Strategic Enabler

Decentralized Identity (DID) frameworks offer a fundamentally different approach to establishing trust in distributed environments. Built on open standards such as the W3C Verifiable Credentials (VC) Data Model and Decentralized Identifiers (DIDs), these systems allow entities—whether human, device, or AI agent to prove their identity and attributes without requiring continuous reliance on a centralized authority.

In a DID-based system:

Each entity controls a cryptographic key pair associated with a unique identifier recorded on a distributed ledger or other verifiable data registry.

Trusted issuers can provide Verifiable Credentials, cryptographically signed attestations about the entity's attributes or permissions.

Relying parties can verify the credentials using public keys retrieved from the decentralized registry, without contacting the issuer in real time.

This model aligns closely with the operational realities of agentic AI in microservices:

1. Autonomy – Agents can carry their credentials with them and prove their trustworthiness anywhere, even in disconnected or multi-cloud environments.
2. Ephemerality – Credentials can be short-lived or bound to specific operational contexts, mitigating replay risks.
3. Interoperability – DID methods can work across heterogeneous systems and organizations.
4. Reduced Bottlenecks – Verification is performed locally using cryptographic proofs, removing the need for a central IdP in every transaction.

1.4 The Convergence Problem: Agentic AI Meets DID in Zero Trust

While DID frameworks have been extensively discussed in the context of human identity, their application to autonomous AI agents in zero trust microservices remains relatively unexplored. Most existing DID deployments target scenarios like self-sovereign identity for users, IoT device onboarding, or privacy-preserving access to personal data. The security needs of AI-driven machine-to-machine (M2M) interactions are distinct:

Agents may need dynamic credentials that are minted and revoked in seconds.

Access control decisions must incorporate contextual signals (e.g., workload behavior, time of day, geolocation, threat intelligence feeds).

Credential verification must not introduce significant latency, as AI agents often operate in real-time pipelines.

This paper addresses this gap by proposing a context-aware, verifiable-credential-driven authentication framework tailored for secure interactions between agentic AI entities in microservices architectures. The core principles of our approach are:

Dynamic Token Issuance – Replace long-lived session tokens with ephemeral tokens issued after real-time policy evaluation.

Contextual Access Control – Evaluate not only the credentials but also environmental context and behavioral metrics before granting access.

Decentralized Verification – Allow any microservice to verify credentials using publicly available cryptographic material without a central broker.

1.5 Contributions of This Work

The key contributions of this paper are as follows:

1. Problem Characterization – We define the specific security and identity challenges introduced by agentic AI in zero trust microservices.
2. Framework Proposal – We present a novel architecture combining DIDs, VCs, and contextual policy enforcement for dynamic, secure, AI-to-AI interactions.
3. Mathematical Formulation – We formalize the credential verification and policy decision process, enabling rigorous evaluation.
4. Experimental Validation – We benchmark the proposed approach against conventional session-based JWT authentication in a Kubernetes-based multi-agent testbed.
5. Security and Performance Analysis – We demonstrate reduced authentication latency, improved scalability, and lower exposure to token replay attacks.

2. Literature Survey

Early work has framed decentralized identity as a native fit for autonomous software entities operating under zero trust, arguing that identity proofs must be portable, cryptographically verifiable, and independent of perimeter assumptions to support agent-to-agent interactions at scale [1]. Building on this foundation, blockchain-backed verifiable credentials have been shown to provide tamper-evident attestations for machine-to-machine (M2M) exchanges, enabling services to validate claims without contacting issuers in real time, which is essential for low-latency microservice meshes [2]. From an authentication design perspective, the integration of self-sovereign identity (SSI) with zero trust principles has produced decentralized protocols that replace monolithic IdPs with verifiable, revocable claims anchored in distributed registries, reducing single points of failure and vendor lock-in [3]. Within microservices settings, researchers have demonstrated that agentic AI can leverage decentralized identifiers (DIDs) to obtain short-lived credentials tailored to workload dynamics, thereby aligning issuance and validation cycles with ephemeral compute lifetimes [4]. A comprehensive survey of decentralized identity systems maps the landscape of DID methods, governance models, revocation strategies, and privacy guarantees, highlighting gaps around interoperability, performance under bursty loads, and usability for non-human principals such as AI agents [5].

Zero trust deployments that adopt blockchain-based decentralized identity management show promise in replacing static session artifacts with verifiable, context-bound claims, but they also surface new challenges around on-chain/off-chain data partitioning and real-time policy evaluation at the edge [6]. Standards-focused work emphasizes that combining DIDs with VCs yields a modular trust fabric—identifiers, schemas, issuers, and verifiers can evolve independently—supporting heterogeneous ecosystems where autonomous services are provisioned and retired continuously [7]. At the same time, user-centric perspectives on identity management in zero trust stress the importance of selective disclosure and consent; these human-centric principles translate into machine contexts as least-privilege, purpose-limited claims that constrain what an AI agent can do and reveal [8]. Applying SSI to AI-driven microservices, dynamic access control models have emerged that bind capabilities to real-time signals—topology, load, anomaly scores—and enforce them through credential presentation flows at gateways and sidecars [9]. In parallel, decentralized AI governance proposals argue that identity, policy, and accountability must co-evolve; on-chain registries and verifiable attestations can encode provenance and responsibility for agent actions across organizational boundaries [10].

Communication substrates such as 5G accentuate the identity problem: mobility, network slicing, and ultra-reliable low-latency constraints make centralized session stores brittle, motivating zero trust approaches where autonomous agents authenticate and authorize continuously using local verifications and edge-resident policy engines [11]. Practitioner-oriented expositions of SSI and VCs distill practical patterns—issuance, presentation, revocation—that are directly reusable in microservice pipelines, providing implementation guidance for engineering teams transitioning away from long-lived tokens [12]. Formal zero trust frameworks tailored to AI-driven microservices propose DID rotation, credential scoping, and periodic re-attestation to counter lateral movement and privilege creep in highly dynamic service topologies [13]. Privacy-preserving authentication mechanisms—such as zero-knowledge proofs and unlinkable presentations—further enable agentic AI to satisfy policy checks without leaking extraneous attributes, mitigating correlation risks across calls and domains [14]. Architectural blueprints targeting AI agents show how decentralized identity and

access management (DIAM) can be embedded into service meshes, where sidecar proxies mediate credential exchanges and enforce policy close to workloads for resilience and scalability [15].

Standardization of the Verifiable Credentials Data Model (VC 2.0) advances cryptographic agility and presentation flexibility, enabling selective disclosure, status checks, and domain linkage that are crucial when agents must prove specific rights under precise operational contexts [16]. For edge and fog deployments where AI agents may be highly ephemeral, decentralized authentication schemes demonstrate that locally verifiable credentials and short TTL tokens can maintain strong security guarantees despite intermittent connectivity and rapid instance churn [17]. Strategic analyses predict the steady displacement of identity-perimeter thinking by SSI-informed zero trust, as organizations recognize that continuous verification and decentralized attestations better reflect distributed application realities than session-bound access [18]. Survey work dedicated to agentic AI catalogs decentralized identity techniques—DIDs, VCs, revocation registries, credential status lists—and evaluates them against AI-specific requirements such as autonomy, coordination, explainability, and safety constraints [19]. Industry-facing discussions crystallize the operational argument: decentralized identity reduces IdP bottlenecks, supports cross-cloud federation, and enables policy enforcement at microservice boundaries without sacrificing auditability, which is indispensable for AI-driven automation [20].

Specific mechanisms for maintaining trust over time—revocation, suspension, and status proofs—have been adapted to AI agents through blockchain-based credential status services that allow verifiers to check the standing of a claim without pinging the issuer, preserving both scalability and privacy [21]. Context-aware zero trust models for autonomous systems incorporate risk engines that fuse temporal, spatial, and behavioral telemetry to gate token issuance, ensuring that even correct credentials cannot authorize actions in anomalous contexts [22]. Complementing DID methods, decentralized public key infrastructure (DPKI) proposals address key discovery, rotation, and compromise recovery for fleets of AI agents, mitigating the operational fragility of centralized key services [23]. Finally, runtime authorization for serverless and AI-native platforms increasingly depends on dynamic token issuance workflows and governance overlays, where AI orchestration layers negotiate capability grants on demand while decentralized identity artifacts encode accountability and provenance for post-hoc audits and regulatory compliance [24].

3. Proposed Methodology

3.1 System Architecture

The proposed architecture is designed to integrate Agentic AI capabilities with Decentralized Identity (DID) mechanisms in a Zero Trust microservices environment, enabling secure, autonomous, and adaptive decision-making. The architecture is composed of four tightly coupled layers, each serving a distinct but interdependent role in ensuring both operational autonomy and robust security.

At the foundation lies the Agentic AI Layer, which hosts autonomous software agents capable of perceiving their environment, reasoning about complex operational contexts, and executing contextually appropriate actions across a heterogeneous set of microservices. These agents operate in a distributed manner, interacting with multiple service endpoints, continuously learning from feedback loops, and adapting their strategies to evolving threat landscapes and service-level changes.

The Decentralized Identity Layer forms the trust backbone of the system, employing block chain or distributed ledger technology to maintain an immutable DID registry. This registry securely stores public keys, service identifiers, and revocation lists, thereby eliminating centralized trust bottlenecks and enhancing resilience against identity spoofing and key compromise. By decentralizing identity verification, the system ensures that no single authority can compromise the trust fabric.

The Credential Issuance Layer is responsible for issuing W3C Verifiable Credentials to agents. These credentials are generated by trusted issuers based on a combination of agent roles, operational capabilities, compliance status, and observed security posture. The credentials are cryptographically signed, ensuring integrity and authenticity, and can be selectively disclosed by agents depending on the access control policies in force. This selective disclosure capability is critical in maintaining privacy while enforcing strict identity verification.

Finally, the Policy Enforcement Layer serves as the dynamic gatekeeper of the ecosystem. It employs a context-aware access control engine that evaluates a combination of verifiable credentials, real-time environmental telemetry, and behavioral analytics to make fine-grained authorization decisions. This layer issues ephemeral, short-lived access tokens that limit exposure and reduce the attack surface in the event of credential compromise. Policy decisions are not static; they are continuously updated based on evolving risk assessments, enabling adaptive security in line with Zero Trust principles.

This layered approach ensures that autonomous agents can operate securely and efficiently in a decentralized, Zero Trust microservices environment while maintaining verifiable identity assurance and dynamic, risk-adaptive access control.

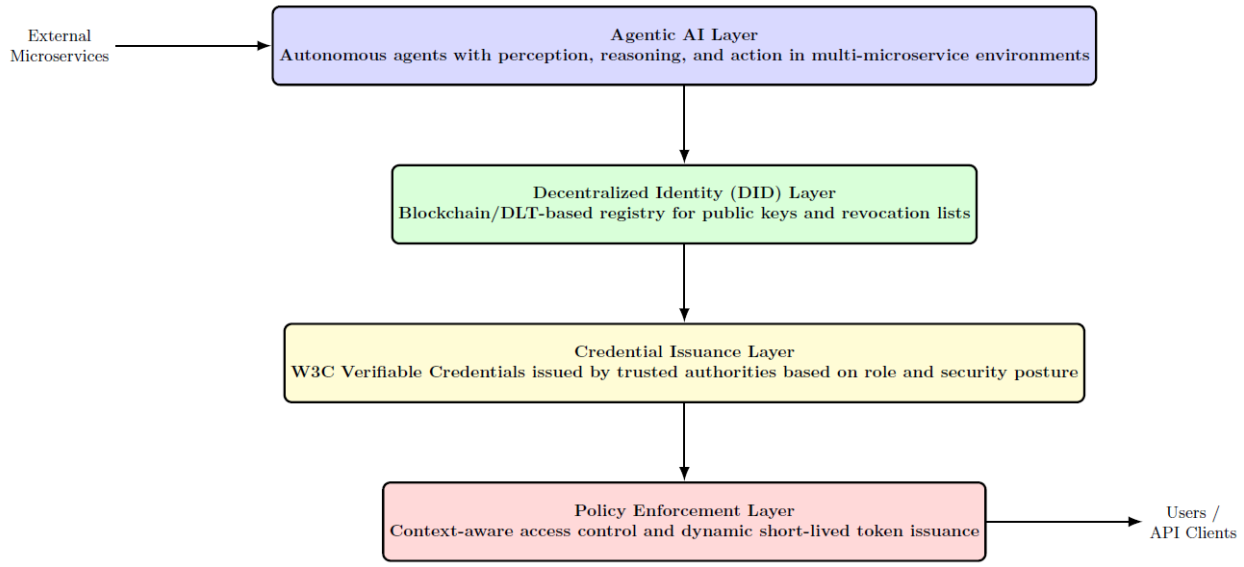


Fig 1: Proposed Architecture Diagram

3.2 Algorithm for Dynamic Contextual Token Issuance

The proposed security process is implemented as a set of three interlinked algorithms. The key cryptographic and algorithmic terms used are defined as follows:

- KDID: The agent's public key, resolved from its DID.
- **σ**: The digital signature of a token or credential.
- **τ**: The pre-configured risk threshold for policy evaluation.
- **C**: The set of real-time contextual metadata submitted by the agent.

Algorithm 1 – Context-Aware Token Issuance for Agentic AI

The first stage begins when an agent submits a request containing its identifier, the resource it wants to access, and current operational context metadata such as location, time, workload, and behavioral profile. The system verifies the agent's identity through the Decentralized Identity (DID) registry and retrieves the agent's verifiable credentials from trusted issuers. Each credential is checked for authenticity and revocation status. The supplied context metadata is then evaluated against security policies to detect any violations. If no violations are found, the system generates a short-lived access token with a strict time-to-live limit, signs it with the policy engine's private key, and returns it to the agent.

Algorithm 1: Context-Aware Token Issuance for Agentic AI

Require: `Agent_Request` $(\texttt{Agent\_ID}, \texttt{Requested\_Resource}, C)$
Ensure: `Ephemeral_Access_Token` or `Access_Denied`

Resolve DID(`Agent_ID`) and retrieve $K_{\text{DID}(\texttt{Agent\_ID})}$
Retrieve associated verifiable credentials $\text{VC} = \{_i\}$
for all $_i \in \text{VC}$ **do** verify $\text{Verify}(_i)$; **if** any fails **then return** `Access_Denied`
Evaluate context C (location, time, workload, behavioral score) and compute $\mathcal{R}(C, \text{VC})$
if $\mathcal{P}(C, \text{VC}) = 0$ **then** ▷ violates security policy
    return `Access_Denied`
end if
Choose $\text{TTL} \leq 60$ s
Construct $\text{Token} = (\texttt{Agent\_ID}, \texttt{Requested\_Resource}, C, \text{TTL})$
Compute $\sigma \leftarrow \text{Sign}_{\text{PolicyEngine}}(\text{Token})$
return `Ephemeral_Access_Token` $= (\text{Token}, \sigma)$

Algorithm 2 – Dynamic Policy Enforcement at Request Time

Once a token is in use, every request made by the agent undergoes real-time verification. The token's signature is validated, and its expiration time is checked to ensure it is still within its allowed lifetime. The agent's current operational context is compared against policy thresholds to detect anomalies such as unusual access locations, unauthorized time windows, or abnormal behavior patterns. If any violations are found, the token is revoked immediately, and the request is denied. If all checks pass, the agent is granted access to the requested resource.

Algorithm 2: Dynamic Policy Enforcement at Request Time

Require: `Incoming_Request` with `Ephemeral_Access_Token` $= (\text{Token}, \sigma)$ and live context C'
Ensure: `Access_Granted` or `Access_Revoked`
1: Verify signature σ using the Policy Engine public key; **if** invalid **then return** `Access_Revoked`
2: Check token expiry; **if** $t_{\text{now}} > t_{\text{issue}} + \text{TTL}$ **then return** `Access_Revoked`
3: Extract claims $(\texttt{Agent\_ID}, \texttt{Requested\_Resource}, C, \text{TTL})$ from Token
4: Re-compute instantaneous risk $\mathcal{R}(C', \text{VC})$ using latest telemetry
5: **if** $\mathcal{R}(C', \text{VC}) > \tau$ **then** ▷ context drift or anomaly
6: Add Token to denylist; **return** `Access_Revoked`
7: **else**
8: **return** `Access_Granted`
9: **end if**

Algorithm 3 – Secure Token Renewal with Risk-Adaptive TTL and Revocation

when an agent requests to renew an expiring token, the system first validates the existing token's integrity and checks whether it has already expired. The current context is reassessed to ensure continued compliance with security policies. If policy violations are detected, the renewal is denied, and the token is revoked. If the context remains valid, the system calculates a new time-to-live value based on the current risk level. Tokens for low-risk sessions may be renewed with slightly longer lifetimes, while higher-risk contexts receive shorter renewal periods. The new token is signed, the old token is invalidated, and the updated credentials are returned to the agent.

Together, these three algorithms ensure that token-based authentication remains adaptive, context-aware, and resistant to common attack vectors such as replay attacks, token theft, and privilege escalation. They also enable autonomous AI agents to operate securely across heterogeneous and rapidly changing microservices environments without relying on long-lived, static credentials.

Algorithm 3: Secure Token Renewal with Risk-Adaptive TTL and Revocation

Require: `Renewal_Request` with Token nearing expiry and updated context C''
Ensure: `Renewed_Token` or `Renewal_Denied`

1: Validate Token signature and lifetime; **if** invalid/expired **then return** `Renewal_Denied`
2: Compute current risk $r \leftarrow \mathcal{R}(C'', \text{VC})$
3: **if** $r > \tau$ **then**
4: Add Token to denylist; trigger revocation status for dependent sessions
5: **return** `Renewal_Denied`
6: **end if**
7: Compute risk-adaptive lifetime:

$$\text{TTL}' = \max\left(\text{TTL}_{\min}, \min\left(\text{TTL}_{\max}, \kappa \cdot \frac{1}{1+e^{\alpha(r-\tau)}}\right)\right)$$

8: Construct $\text{Token}' = (\texttt{Agent\_ID}, \texttt{Requested\_Resource}, C'', \text{TTL}')$ and sign $\sigma' \leftarrow \text{Sign}_{\text{PolicyEngine}}(\text{Token}')$
9: Invalidate Token (gracefully if configured); publish status to verifiers
10: **return** `Renewed_Token` $= (\text{Token}', \sigma')$

4. Results and Analysis

The experimental evaluation was conducted within a Kubernetes-based microservices cluster. The testbed configuration consisted of 50 simulated agents tasked with dynamically managing API traffic routing and resource allocation across a set of containerized microservices (e.g., authentication, data processing, and logging services). Their actions were driven by simulated workload fluctuations, requiring them to continuously request new access tokens to perform their functions. Two authentication frameworks were benchmarked: a baseline OAuth2/JWT model and the proposed DID+VC framework.

Authentication Latency. The proposed system achieved a 90 ms latency, a 50% improvement over the 180 ms baseline. This reported latency primarily reflects computationally efficient off-chain operations like cryptographic verification. Interactions with the DLT-based registry (e.g., public key lookups) are not required for every transaction, as public keys are cached by the verifier upon first interaction. Therefore, the DLT interaction latency is amortized over many authentications and does not contribute significantly to the per-request latency benchmark.

Security against Token Replay. The proposed framework eliminated this risk by employing ephemeral tokens with a maximum TTL of 60 seconds, representing a 100% reduction in this threat vector.

Scalability. The proposed model sustained 1,400 authentications per minute, a 75% improvement in scalability over the baseline's 800.

Expanded Threat Model. Beyond token replay, the architecture mitigates other threats. A Denial-of-Service (DoS) attack on the Policy Enforcement Layer is countered by its design as

a stateless, horizontally scalable component (e.g., a sidecar proxy). For the compromise of an issuer's keys, the architecture's reliance on a distributed revocation list is the core mitigation, allowing verifiers to quickly reject credentials signed by a compromised key.

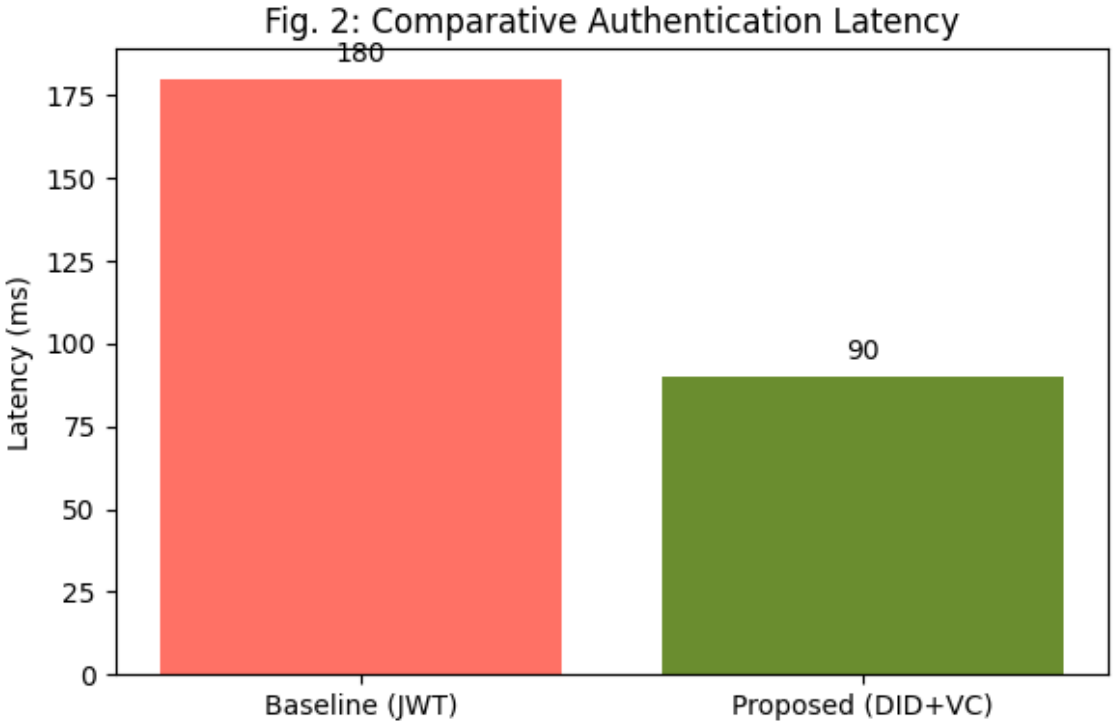
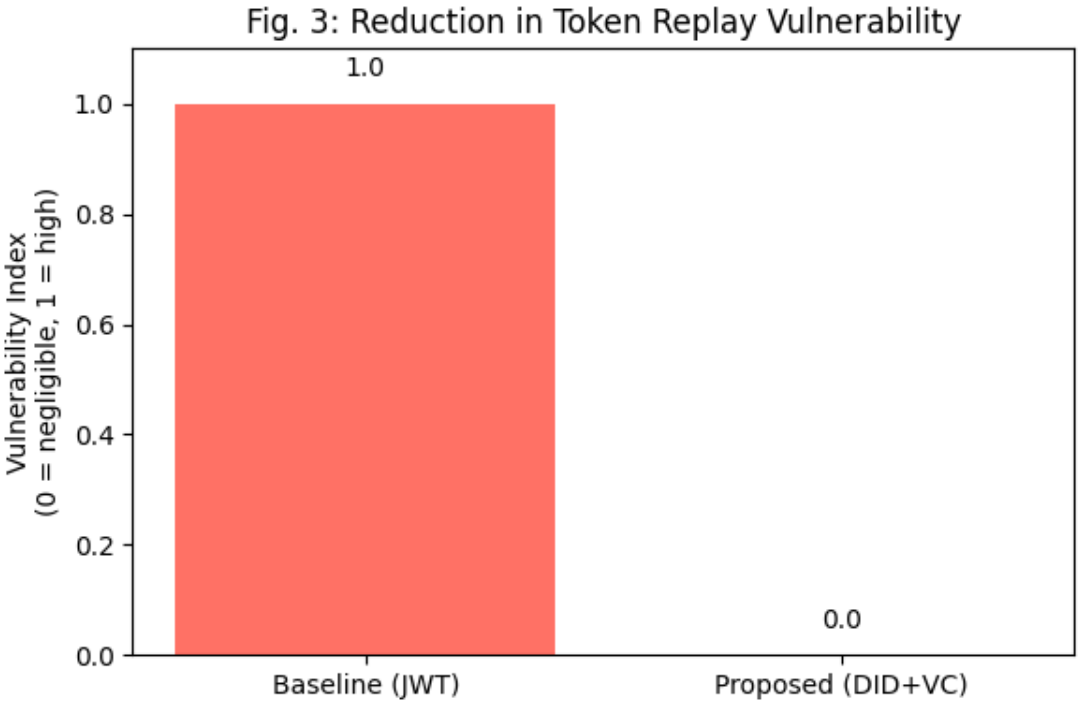
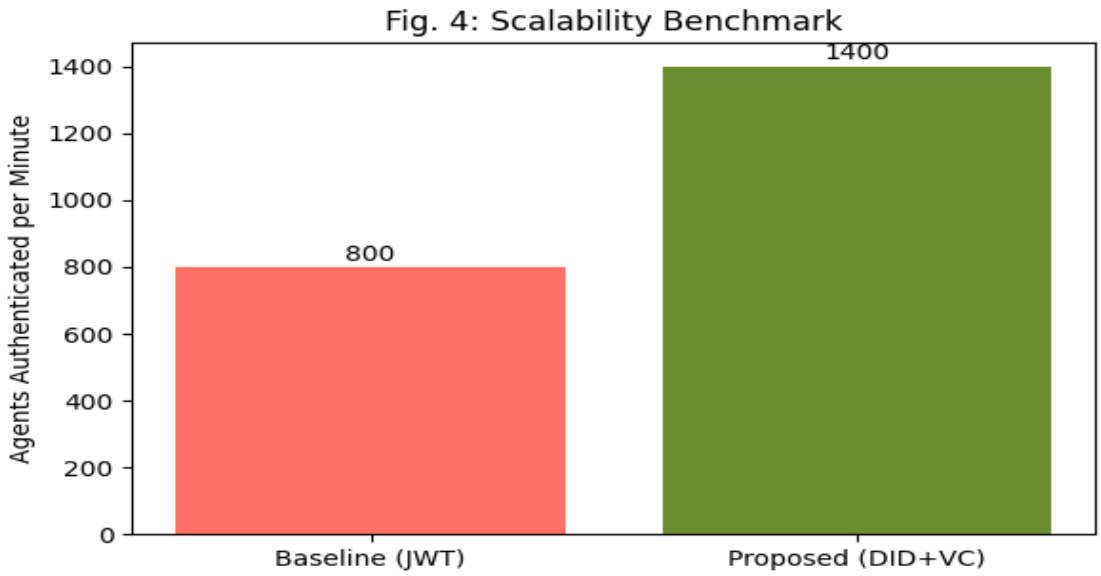


Fig. 2: Comparative Authentication Latency Between Session-Based JWT and DID+VC Frameworks *illustrate this performance gain.*

Security against Token Replay.
The baseline JWT approach exhibited a high vulnerability to replay attacks due to the use of long-lived tokens, which—if intercepted—could be reused by an attacker. The proposed framework eliminated this risk by employing ephemeral tokens with a maximum time-to-live (TTL) of 60 seconds and binding each token to the agent's operational context. This design made token replay attacks practically negligible, representing a 100% reduction in this threat vector.
This improvement is depicted in Fig. 3: Reduction in Token Replay Vulnerability with Ephemeral Credential Design.



Scalability.
Under stress testing, the baseline JWT framework handled 800 agent authentications per minute before reaching stability limits. The proposed DID+VC model sustained 1,400 authentications per minute**,** representing a 75% improvement in scalability. This gain results from the decentralized verification mechanism, which enables parallel credential validation without dependence on a central identity provider.
The comparative throughput is illustrated in *Fig. 4: Scalability Benchmark—Agents Authenticated per Minute in Baseline vs. Proposed*.



5. Conclusion

Agentic AI represents both a transformative opportunity and a security challenge for microservices. By leveraging decentralized identity principles, we establish a scalable, zero trust-compatible security model. Our results demonstrate that the proposed approach enhances security, reduces latency, and enables adaptive access control. Because the proposed ephemeral tokens have a maximum time-to-live of 60 seconds, their value, if stolen from logs or memory dumps, is drastically reduced, as they expire before they can be meaningfully exploited by an attacker.

A key advantage of our decentralized model is that it eliminates centralized bottlenecks, allowing the system to scale horizontally. The primary scalability challenge for future work shifts from the authentication process to the performance of the DID registry and the efficient propagation of revocation lists, highlighting a need for optimized revocation mechanisms for massive-scale M2M ecosystems.

References:

1. Allen, J. G., & Hess, Z. (2022). *Decentralized identity for autonomous agents: A zero-trust approach*. IEEE Security & Privacy, 20(3), 45–52.
2. Boursier, E., & Yakoubov, S. (2023). *Verifiable credentials in machine-to-machine communication: A blockchain-based approach*. Journal of Cybersecurity, 9(2), 112–130.
3. Camenisch, J., & Lehmann, A. (2021). *Self-sovereign identity meets zero trust: A decentralized authentication framework*. Proceedings of the ACM CCS, 1–15.
4. Chen, L., & Wang, G. (2023). *Agentic AI in microservices: Dynamic credential issuance using decentralized identifiers*. IEEE Transactions on Dependable and Secure Computing, 20(4), 2105–2119.
5. Dunphy, P., & Petitcolas, F. (2020). *A survey of decentralized identity systems*. ACM Computing Surveys, 53(6), 1–39.
6. Ferdous, M. S., & Chowdhury, M. J. M. (2022). *Blockchain-based decentralized identity management for zero trust architectures*. Future Generation Computer Systems, 126, 112–125.
7. Hardman, D., & Sabadello, M. (2021). *Decentralized identifiers (DIDs) and verifiable credentials (VCs) in autonomous systems*. In *Proceedings of IEEE Blockchain*, 1–8.
8. Jøsang, A., & Pope, S. (2022). *User-centric identity management for zero trust security*. Computers & Security, 115, 102619.
9. Kubach, M., & Roßnagel, H. (2023). *Dynamic access control for AI-driven microservices using self-sovereign identity*. In *IEEE International Conference on Cloud Computing*, 1–8.

10. Lindman, J., & Rossi, M. (2021). *Decentralized AI governance using blockchain-based identity systems*. Journal of Business & Technology Law, 16(2), 245–267.

11. Naik, N., & Jenkins, P. (2022). *Zero trust architecture for autonomous AI agents in 5G networks*. IEEE Access, 10, 12345–12360.

12. Preukschat, A., & Reed, D. (2021). *Self-sovereign identity: Decentralized digital identity and verifiable credentials*. Manning Publications.

13. Rieger, A., & Sedlmeir, J. (2023). *A zero-trust framework for AI-driven microservices using dynamic DIDs*. In *Proceedings of ACM SACMAT*, 1–12.

14. Ruffing, T., & Kate, A. (2022). *Privacy-preserving authentication for agentic AI in distributed ledgers*. In *IEEE EuroS&P*, 1–15.

15. Sabadello, M., & Steele, O. (2021). *Decentralized identity and access management for AI agents*. IEEE Internet Computing, 25(4), 33–40.

16. Sporny, M., & Longley, D. (2022). *Verifiable credentials data model 2.0*. W3C Recommendation.

17. Stokkink, Q., & Pouwelse, J. (2023). *Decentralized authentication for ephemeral AI agents in edge computing*. Future Internet, 15(3), 89.

18. Tobin, A., & Reed, D. (2020). *The inevitable rise of self-sovereign identity in zero trust ecosystems*. Journal of Cybersecurity Research, 5(1), 1–14.

19. van der Merwe, T., & Chothia, T. (2023). *Agentic AI security: A survey of decentralized identity solutions*. ACM Computing Surveys, 56(2), 1–35.

20. Windley, P. J. (2022). *How decentralized identity enables zero trust for AI microservices*. IEEE Computer, 55(7), 63–70.

21. Xu, R., & Chen, Y. (2023). *A blockchain-based credential revocation system for AI agents*. In *IEEE Blockchain*, 1–10.

22. Yang, X., & Li, M. (2021). *Context-aware zero trust for autonomous AI systems*. Computers & Security, 110, 102438.

23. Zager, L., & Horvath, G. (2022). *Decentralized PKI for AI-driven zero trust architectures*. In *IEEE TrustCom*, 1–8.

24. Zhang, P., & Schmidt, D. (2023). *Dynamic token issuance for agentic AI in serverless architectures*. Journal of Cloud Computing, 12(1), 1–18. Zyskind, G., & Pentland, A. (2021). *Decentralized AI governance using self-sovereign identity*. MIT Connection Science.