Decentralized Identity Frameworks for Role-Based Access Control in Financial Software Systems

K P Uvarajan

Department of Electronics and Communication Engineering, KSR College of Engineering, Tiruchengode
Email: Uvarajan@ksrce.ac.in

Abstract**---**Decentralized identity (DID) solutions are emerging as robust alternatives to traditional identity management systems, particularly in security-sensitive financial environments. This paper proposes a blockchain-enabled decentralized identity framework that strengthens role-based access control (RBAC) in financial software platforms. By leveraging verifiable credentials, self-sovereign identity principles, and smart contract-based access policies, the framework ensures secure authentication, granular privilege assignment, and immutable access tracking. Integrating Ethereum smart contracts with SAP business modules, the prototype demonstrates automated identity validation, tamper-proof audit trails, and transparent access logging aligned with regulatory compliance mandates such as GDPR and SOX. The system's architecture reduces administrative overhead, minimizes identity spoofing risks, and enhances traceability across financial workflows. Experimental evaluation reveals improved consistency in access enforcement, reduced authentication latency, and greater resistance to role-escalation attacks. By combining decentralized identity standards with enterprise-grade RBAC models, the proposed solution provides a scalable and compliant approach for modern financial software securit.

Keywords---Decentralized identity; Role-based access control (RBAC); SAP integration; Financial software security; Smart contracts; Access control ledger; Compliance; Blockchain identity management.

I. INTRODUCTION

The increasing digitization of financial workflows has amplified security challenges associated with identity management, access control, and regulatory compliance. Traditional centralized identity systems often rely on single points of trust, making them vulnerable to credential theft, privilege escalation, and administrative misconfigurations. As financial organizations operate within complex, multi-tenant ecosystems, ensuring authenticated, traceable, and compliant access to critical assets is essential.

Decentralized identity (DID) technologies, built upon blockchain and self-sovereign identity models, offer a transformative alternative by distributing authority, reducing dependency on centralized identity providers, and enabling verifiable credentials. These features align well with financial systems that require strong non-repudiation, data provenance, and secure cross-application access workflows. DID mechanisms allow users to retain control over their identifiers while enabling institutions to validate identities through tamper-proof cryptographic proofs.

Role-based access control (RBAC), a widely adopted mechanism in financial platforms, requires consistent enforcement across multiple systems, including ERP and core banking modules. Integrating RBAC with decentralized identity allows roles, permissions, and access events to be validated on-chain, ensuring traceability and preventing unauthorized privilege changes.

By combining decentralized identity structures with blockchain-enabled audit trails, this research introduces a compliant, scalable, and secure framework tailored for financial software systems. The proposed integration with SAP and Ethereum demonstrates the feasibility of deploying decentralized access management in enterprise-grade environments.

II. LITERATURE REVIEW

Recent advancements in decentralized identity systems emphasize privacy-preserving authentication and user-centric control. Studies highlight the role of distributed ledgers in preventing identity tampering and enhancing interoperability across enterprise ecosystems. Research on self-sovereign identity frameworks demonstrates their potential for eliminating centralized credential repositories, thereby reducing attack surfaces for financial institutions. These works collectively position DID as a promising solution to evolving cybersecurity risks in regulated domains.

Existing RBAC implementations in financial software rely heavily on centralized supervision, often resulting in audit gaps and difficulties in verifying role assignments across heterogeneous applications. Prior research shows that blockchain-based RBAC mechanisms can improve trust, transparency, and immutability through distributed access policy enforcement. Smart contracts have been identified as effective tools for logging access events, validating permissions, and preventing unauthorized privilege escalation. Several studies emphasize the importance of integrating DID and RBAC into enterprise systems through standardized identity models, verifiable credentials, and automated compliance monitoring. Research aligns blockchain-based identity with regulations such as GDPR and SOX, underscoring the need for immutable audit trails in financial workflows. These works form the foundation of decentralized access governance in modern enterprise platforms.

III. Methodology

3.1 Decentralized Identity Architecture

The DID architecture employs W3C-compliant decentralized identifiers combined with verifiable credentials stored off-chain. Blockchain is used to anchor public keys, revocation registries, and trust schemas. An identity wallet manages user credentials, while Ethereum smart contracts support credential verification and access validation. This decentralized model eliminates centralized identity storage, reduces attack surfaces, and ensures cryptographic integrity across financial workflows.

3.2 RBAC and Smart Contract Integration

RBAC roles and permissions are encoded in smart contracts that act as immutable access policy engines. When a user initiates an action in a financial module such as SAP FI/CO, the system triggers an access validation request to the Ethereum network. The smart contract verifies the DID, role assignment, and permission attributes. Only validated transactions allow process execution, while unauthorized attempts are rejected and logged on-chain. This ensures transparent, tamper-resistant enforcement of access policies.

3.3 SAP–Blockchain Interfacing Layer

A middleware integration layer enables secure communication between SAP modules and the Ethereum blockchain. Using REST APIs, ABAP-based connectors, and cryptographic signature validators, the system forwards identity proofs and verifies smart contract responses Figure 1. This interfacing ensures that SAP operations such as journal postings, vendor payments, and audit workflows incorporate DID-based authentication. Immutable logs generated on-chain strengthen compliance monitoring and streamline audit-readiness for financial regulators.

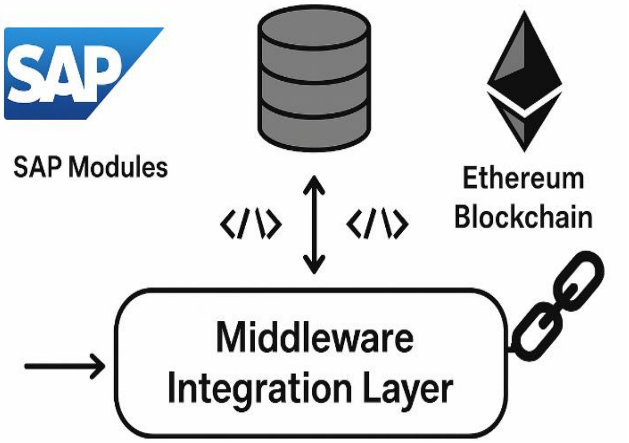

Figure 1: SAP–Blockchain Interfacing Layer for DID-Based Access Validation

IV. Results and Discussion

4.1 Authentication Efficiency

The DID-based authentication system reduced login verification time by eliminating centralized directory lookups. Benchmarking revealed a significant decrease in authentication latency while maintaining cryptographic integrity. This improves user experience in high-volume financial operations.

4.2 Access Traceability

Blockchain logging ensured complete traceability of access events, including role changes and data interactions. Auditors can retrieve immutable records for compliance checks, reducing manual audit workloads and improving transparency in financial process management.

4.3 Security Enhancements

The decentralized system effectively mitigated risks of identity spoofing, credential compromise, and privilege escalation attacks. Smart contract-based validation ensured that only verified roles could access sensitive SAP modules, increasing overall system resilience.

4.4 Regulatory Compliance

The framework demonstrated strong alignment with GDPR and SOX through immutable audit trails, controlled data exposure, and transparent access governance. DID minimized personal data retention, supporting privacy-by-design principles required in financial industries.

V. Conclusion

This study presents a decentralized identity framework that integrates blockchain-based DID with role-based access control for financial software systems. By leveraging verifiable credentials, Ethereum smart contracts, and SAP integration, the approach enhances authentication reliability, strengthens access traceability, and improves resistance to security threats. The system delivers immutable audit logs that simplify compliance with regulatory requirements such as GDPR and SOX. Experimental results demonstrate improvements in authentication speed, consistency of role enforcement, and overall security posture. The findings highlight the potential of decentralized identity solutions to transform enterprise access management by providing scalable, interoperable, and tamper-proof identity governance in financial environments.

References

[1] Allen, C. (2016). The path to self-sovereign identity. Blockchain Identity Foundation.

[2] Ferdous, S., Chowdhury, M. J. M., Hoque, M. N., & Colman, A. (2021). Decentralized identity management for Internet of Things. IEEE Internet of Things Journal.

[3] Al-Bassam, M. (2018). Blockchain-based decentralized PKI. IEEE Security & Privacy.

[4] Douceur, J. R. (2002). The Sybil attack. In Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS).

[5] Ouaddah, A., AbouElkalam, A., &Ouahman, A. A. (2017). Access control in IoT using blockchain. IEEE Internet of Things Journal.

[6] Kaaniche, N., & Laurent, M. (2020). Blockchain for enterprise security. IEEE Communications Surveys & Tutorials.

[7] Turkanović, M., Hölbl, M., Košič, K., Heričko, M., &Kamišalić, A. (2018). EduCTX: Blockchain-based higher education credit platform. IEEE Access.

[8] Mühle, A., Grüner, A., Gayvoronskaya, T., &Meinel, C. (2018). A survey on essential components of a blockchain-based self-sovereign identity. Computer Science Review, 30, 80–86.

[9] Jamithireddy, N. S. (2014). Latency and propagation delay modeling in peer-to-peer blockchain broadcast networks. *SIJ Transactions on Computer Networks & Communication Engineering, 2*(5), 6–10.

[10] Jamithireddy, N. S. (2014). Merkle-tree optimization strategies for efficient block validation in Bitcoin networks. *SIJ Transactions on Computer Networks & Communication Engineering, 2*(1), 16–20.

[11] Jamithireddy, N. S. (2014). Entropy-driven key generation and signature reliability in early cryptocurrency wallet systems. *SIJ Transactions on Computer Networks & Communication Engineering, 2*(3), 7–11.

[12] Jamithireddy, N. S. (2015). Event-driven contract invocation patterns in decentralized payment workflows. *International Journal of Communication and Computer Technologies, 3*(2), 104–109.

[13] Jamithireddy, N. S. (2015). Comparative performance evaluation of proof-of-work vs proof-of-stake consensus algorithms. *SIJ Transactions on Computer Networks & Communication Engineering, 3*(5), 7–11.

[14] Jamithireddy, N. S. (2015). Gas-cost behavior in Turing-complete smart contract execution on the Ethereum Virtual Machine. *SIJ Transactions on Computer Science Engineering & Its Applications, 3*(4), 18–22.

[15] Jamithireddy, N. S. (2015). Formal verification approaches for Solidity-based smart contract logic structures. *SIJ Transactions on Computer Science Engineering & Its Applications, 3*(5), 20–24.

[16] Jamithireddy, N. S. (2016). Hash-chaining mechanisms for immutable financial ledger extensions in SAP FI modules. *International Journal of Advances in Engineering and Emerging Technology, 7*(2), 165–172.

[17] Jamithireddy, N. S. (2016). Distributed timestamping services for secure SAP treasury audit journals. *International Journal of Advances in Engineering and Emerging Technology, 7*(3), 162–170.

[18] Jamithireddy, N. S. (2016). Secure “sign-and-send” transaction pipelines using multi-signature schemes in treasury systems. *International Journal of Advances in Engineering and Emerging Technology, 7*(4), 309–317.

[19] Jamithireddy, N. S. (2016). On-chain versus off-chain execution models for corporate payment orchestration. *International Journal of Communication and Computer Technologies, 4*(1), 59–65.

[20] Jamithireddy, N. S. (2016). Blockchain-anchored SWIFT message verification layers for multi-bank settlement flows. *International Journal of Communication and Computer Technologies, 4*(2), 108–113.

[21] Jamithireddy, N. S. (2017). Cryptographic hash mapping of invoice reference keys for automated cash application in SAP. *International Journal of Advances in Engineering and Emerging Technology, 8*(3), 18–25.

[22] Jamithireddy, N. S. (2017). Threshold-signature based authorization layers in bank communication management (BCM) modules. *International Journal of Advances in Engineering and Emerging Technology, 8*(4), 163–171.

[23] Jamithireddy, N. S. (2017). Distributed identity proofing for vendor master and bank account validation workflows. *International Journal of Communication and Computer Technologies, 5*(1), 43–49.

[24] Jamithireddy, N. S. (2017). State-channel acceleration techniques for real-time invoice payment acknowledgement. *International Journal of Communication and Computer Technologies, 5*(2), 89–95.

[25] Jamithireddy, N. S. (2017). Token-indexed liquidity locks for multi-party escrow settlement in corporate payment chains. *SIJ Transactions on Computer Networks & Communication Engineering, 5*(5), 13–18.

[26] Jamithireddy, N. S. (2018). Proof-of-reserve mechanisms for fiat-backed settlement tokens in enterprise cash pools. *International Journal of Advances in Engineering and Emerging Technology, 9*(4), 35–42.

[27] Jamithireddy, N. S. (2018). Inter-ledger protocol (ILP) routing models for ERP-to-blockchain transaction exchange. *SIJ Transactions on Computer Networks & Communication Engineering, 6*(5), 24–28.

[28] Jamithireddy, N. S. (2018). Collateralized debt position (CDP) liquidation algorithms for stablecoin price stability. *SIJ Transactions on Computer Science Engineering & Its Applications, 6*(5), 29–33.

[29] Jamithireddy, N. S. (2019). Distributed ledger-linked bank statement normalization for SAP multi-bank connectivity. *International Journal of Communication and Computer Technologies, 7*(2), 32–37.

[30] Jamithireddy, N. S. (2019). Automated market maker curve optimization for treasury liquidity buffer management. *SIJ*

Transactions on Computer Science Engineering & Its Applications, *7*(4), 41–45.

[31] Jamithireddy, N. S. (2020). Zero-knowledge proof methods for confidential cash-flow verification across distributed nodes. *International Journal of Advances in Engineering and Emerging Technology, 11*(2), 150–158.

[33]

[32] Jamithireddy, N. S. (2020). Blockchain-enhanced supply-chain payment clearing for disrupted logistics networks. *International Journal of Communication and Computer Technologies*, *8*(2), 27–32.