ENIGMA · ACADEMIC PUBLISHING

Finance & AI

Volume 1 · Number 2

2026 · Regular Issue

Published by IAIC AI RESEARCH & TRADING — FZCO, Dubai, United Arab Emirates

enigma.ist/j/finance-ai

Every article in this issue is free to read, free to download and free to publish.

CONTENTS

Article 1033

Parakletos: On-Chain Identity and Accountability Architecture for Autonomous AI Agents in Trust-Critical Systems

Maxwell, A.B.M.

DOI 10.66693/finai.1033

This file is a print-ready copy of the complete issue, assembled from the published articles. It is provided free of charge and may be printed and distributed without permission.

Generated 2026-11-03. The authoritative version of every article is the one at its registered DOI.

Finance & AI, Vol. 1, No. 2, 2026 DOI: 10.66693/finai.1033
Received: May 8, 2026 Accepted: August 6, 2026 Published: August 6, 2026

PARAKLETOS: ON-CHAIN IDENTITY AND ACCOUNTABILITY ARCHITECTURE FOR AUTONOMOUS AI AGENTS IN TRUST-CRITICAL SYSTEMS

Antoine B.M. Maxwell[1]*

[1] *o2O Substation Framework, Memphis, TN, USA*

maxdev901@gmail.com

ABSTRACT

As autonomous AI agents increasingly operate in enterprise and multi-agent environments, the absence of verifiable identity, immutable action provenance, and enforceable authorization constitutes a critical infrastructure gap. This paper introduces Parakletos, a cryptographic accountability framework deployed on the Solana blockchain that provides Agent Passports: on-chain credentials binding agent identity, delegation permissions, and action logs with permanent provenance. Leveraging Solana's Address Lookup Tables (ALTs) for scalable node management and a hardware-anchored two-node physical quorum for deterministic validation, Parakletos successfully coordinated 120 autonomous AI agents in a single session on Solana Devnet. The architecture is designed around the principle of Enabling by Design, treating compliance and sovereignty as structural features rather than post-hoc additions. We present the system architecture, implementation details, experimental results, and implications for compliance-native multi-agent AI deployment across trust-critical operational domains.

Keywords: autonomous AI agents, on-chain identity, action provenance, multi-agent systems, Solana blockchain, cryptographic accountability, distributed ledger, AI governance, quorum validation, agent passports, decentralized identity, self-sovereign infrastructure

1 INTRODUCTION

The emergence of autonomous AI agents as operational actors in enterprise workflows, multi-agent networks, and critical infrastructure represents a fundamental shift in how computational systems engage with the world. Unlike traditional software, autonomous agents make decisions, initiate transactions, and coordinate with other agents, often without direct human oversight at each step. This autonomy creates an accountability vacuum: when an agent acts, there is currently no standardized mechanism to cryptographically prove what acted, under whose authority, and with what scope of permission.

Existing AI orchestration frameworks such as LangChain, AutoGPT, and multi-agent platforms treat identity and authorization as application-layer concerns, implemented ad hoc and stored off-chain. This architecture creates several failure modes in critical contexts: (1) *impersonation*: agents can act on behalf of others without verifiable credential binding; (2) *repudiation*: action logs are mutable or absent, enabling post-hoc denial of agent behavior; and (3) *authorization drift*: permission scopes expand beyond intended boundaries without enforcement checkpoints. These gaps present operational risks in trust-critical domains such as financial systems, supply chains, healthcare, or municipal infrastructure.

This paper introduces Parakletos, a sovereign accountability layer built on the Solana blockchain that addresses these failure modes by design. The name derives from the Greek *paraklētos* (advocate, intercessor), reflecting the system's role as the trusted intermediary that validates agent identity and at-

*Corresponding author.

 DOI: 10.66693/finai.1033
Received: May 8, 2026 Accepted: August 6, 2026 Published: August 6, 2026

tests to agent action. Parakletos provides three primitive capabilities: cryptographic Agent Passports for verifiable identity, immutable on-chain action logs for permanent provenance, and registry-based authorization for enforceable permission scoping. These primitives are implemented as Anchor programs on Solana Devnet, with a hardware-anchored two-node quorum providing physical-layer attestation.

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 presents the Parakletos architecture. Section 4 describes the implementation. Section 5 presents experimental results from Devnet deployment. Section 6 discusses implications and limitations. Section 7 concludes.

2 Related Work

2.1 Decentralized Identity and Self-Sovereign Infrastructure

The W3C Decentralized Identifiers (DID) specification [1] provides a foundational standard for self-sovereign identity, enabling entities to create verifiable identifiers without reliance on centralized registries. Verifiable Credentials (VCs) [2] extend this to attestable claims. However, existing DID implementations target human identity and are not optimized for the high-throughput, low-latency requirements of autonomous AI agent coordination. The Trust over IP framework [3] provides governance models but lacks implementation-level primitives for programmatic agent authorization.

2.2 Blockchain-Based Accountability Systems

Blockchain ledgers have been proposed as accountability substrates for IoT devices [4], supply chain provenance [5], and financial audit trails [6]. Digital Twin architectures applied to investment cash flows [7], such as those examined in this journal's Vol. 1, No. 2, demonstrate the viability of distributed ledger environments for modeling complex financial state. Parakletos extends this paradigm to AI agent identity, treating each agent's on-chain record as a living digital passport rather than a static data entry.

2.3 AI Agent Orchestration

Multi-agent frameworks including CrewAI, AutoGen, and LangGraph have advanced agent coordination capabilities but treat authorization as a configuration-layer concern. None provide cryptographically binding identity at the agent level, nor do they anchor action logs to an immutable ledger. Parakletos is positioned as an accountability substrate that existing orchestration frameworks can integrate with, rather than a replacement for orchestration logic.

2.4 Decentralized Identity and Governance for Autonomous AI Agents

A rapidly developing body of work extends decentralized identity credentials to autonomous software agents. Rodriguez Garzon et al. [11] propose issuing W3C DIDs and Verifiable Credentials directly to AI agents, establishing cryptographic bindings between agents, their capabilities, and their human or organizational principals. Saavedra [12] addresses verifiable delegation across centralized, federated, and decentralized identity systems, focusing on revocable, auditable authority chains for agents acting on behalf of principals. Karim et al. [13] survey the intersection of AI agents and blockchain, identifying secure and scalable multi-agent collaboration, and the absence of enforceable on-chain authorization, as open challenges. Parakletos is complementary to this line of work but differs in enforcement locus and on-chain action provenance. Where prior work anchors claims *about* agents, Parakletos enforces the authorization itself at the smart contract layer and commits action-log roots on-chain, addressing the accountability gap identified in [13].

Finance & AI, Vol. 1, No. 2, 2026 DOI: 10.66693/finai.1033
Received: May 8, 2026 Accepted: August 6, 2026 Published: August 6, 2026

3 PARAKLETOS ARCHITECTURE

3.1 DESIGN PHILOSOPHY: ENABLING BY DESIGN

Parakletos is built on the principle of Enabling by Design, the thesis that compliance, accountability, and sovereignty must be architectural features, not post-hoc constraints. This inverts the conventional approach where monitoring mechanisms are layered onto existing systems retroactively. In the Enabling by Design paradigm, every agent that joins the network does so through a verifiable registration act; every action is logged at the point of execution; and every authorization decision is enforced by program logic rather than application policy. The result is a system where trust is not assumed but proven.

3.2 CORE COMPONENTS

Program Derived Address (PDA): a Solana account deterministically derived from a program ID and seed value, used to store persistent state without requiring a private key.

Component	Function	Implementation
ShardRegistry	On-chain node registration and state management	Anchor program (Rust)
Agent Passport	Cryptographic identity binding per agent	Account PDA per node
Address Lookup Table (ALT)	Scalable batch authorization	Solana native ALT API

Table 1: Core components of the Parakletos architecture.

3.3 AGENT PASSPORT MODEL

Each agent registered in Parakletos receives an Agent Passport: a Program Derived Address (PDA) account anchored to the ShardRegistry. The Passport contains: (1) a unique node identifier derived from the agent's public key; (2) an authorization timestamp and authorizing entity; (3) a permission scope field defining the agent's operational boundaries; and (4) a hash-linked log root enabling action provenance verification without storing full logs on-chain. This structure balances on-chain verifiability with storage efficiency. Full action logs are maintained off-chain (Section 6.2); the Passport's log root is re-committed on-chain at state synchronization checkpoints, with the commit interval configurable per deployment (in the experimental session, at session finalization). Between commitments, actions are attested only by the local hash-linked chain: an adversary who fully controls the off-chain store could suppress uncommitted entries, but cannot alter any action preceding the last committed root without breaking the hash chain against the on-chain commitment.

3.4 QUORUM ARCHITECTURE & HARDWARE BOUNDARIES

To establish physical-layer roots of trust, Parakletos employs a two-node physical quorum architecture. The primary authority node (Mothership) manages upgrade authority and executes state transitions, while a secondary validation node connects over an isolated physical network interface (TCP/IP over Thunderbolt). State transitions require bi-directional signature attestation from both nodes before committing to the ShardRegistry, mitigating single-node compromise risks. While a two-node physical topology introduces explicit physical locality dependencies, it provides verifiable physical isolation during key initialization and registration sequences. Deployment parameters for the proof-of-concept configuration are given in Appendix A.

3.5 THREAT MODEL

Parakletos assumes an adversary capable of compromising off-chain agent runtimes, manipulating application-layer requests, or eavesdropping on unencrypted RPC communication. The architecture explicitly protects against: (1) *Unauthorized Action Execution*: prevented via PDA program-derived signers enforcing registry constraints; (2) *Historical Log Tampering*: prevented by committing cryptographic log roots to Solana state; and (3) *Sybil Agent Generation*: mitigated via ShardRegistry registration deposits and authorized node keys.

Finance & AI, Vol. 1, No. 2, 2026 DOI: 10.66693/finai.1033
Received: May 8, 2026 Accepted: August 6, 2026 Published: August 6, 2026

4 IMPLEMENTATION

The system is implemented as a set of Anchor smart contracts deployed to Solana Devnet under Program ID `FsRpXPvwNbCaCU3CwC9UW9eFrwJYLCaq4hFVFhXyNd3w`. The client library is authored in TypeScript utilizing `@solana/web3.js` and `@coral-xyz/anchor`. Address Lookup Tables (ALTs) are created programmatically to load up to 256 account addresses into a single transaction instruction set, bypassing Solana's legacy transaction size constraints during batch node authorization.

5 EXPERIMENTAL RESULTS

Testing was conducted on Solana Devnet evaluating a workforce fleet of 120 autonomous AI agent accounts initialized and authorized in a single execution session. Using Address Lookup Tables (ALT), 120 agent Passports were loaded and batch-authorized within an ALT structure finalized on-chain at address `5zGKvvDZS1tM3JTqKqds1yhwSrMLxARJ4wczTQecrPiH`. Batch authorization execution completed in approximately 125,352 ms across Devnet RPC constraints with a 100% transaction success rate across all 120 nodes. Storage capacity for the root ShardRegistry was initialized with a declared operational capacity envelope of 1.8 TB. Key measurements are summarized in Table 2.

Metric	**Result**
Nodes coordinated (single session)	120
Authorization method	ALT-based batch
End-to-end session time	125,352 ms
Transactions required (vs. legacy 32-account limit)	1 (vs. 4; 75% reduction)
Total session cost	0.0284 SOL
Transaction success rate	100% (all 120 nodes)
Quorum nodes	2 (hardware-anchored)
Network	Solana Devnet

Table 2: Parakletos Devnet deployment summary.

6 IMPLICATIONS AND LIMITATIONS

6.1 SYSTEM IMPLICATIONS

By embedding authorization enforcement directly into on-chain program logic, Parakletos provides a structural foundation for verifiable AI governance. Enterprise operators can audit agent permissions deterministically without inspecting internal model weights or relying on self-reported logs. The architecture establishes a practical model for compliant agent operations across shared decentralized infrastructure.

6.2 LOG STORAGE & PROVENANCE ARCHITECTURE

To reconcile blockchain storage costs with high-frequency agent actions, Parakletos employs a hybrid storage strategy. Full action payloads, execution transcripts, and telemetry are stored off-chain in distributed storage layers (e.g., Arweave/IPFS) or localized encrypted log vaults. Only Merkle tree roots or hash-linked chain commitments are posted on-chain during state synchronization checkpoints. This maintains complete auditability while minimizing ledger footprint.

7 CONCLUSION

Parakletos addresses the fundamental accountability gap in autonomous AI agent systems by establishing an on-chain identity, authorization, and action log provenance layer on Solana. By combining Agent Passports, Address Lookup Tables, and hardware-anchored quorum validation, the system

Received: May 8, 2026 Accepted: August 6, 2026 Published: August 6, 2026

demonstrates scalable, verifiable agent coordination. Built on the Enabling by Design paradigm, Parakletos provides an operational framework for autonomous agents as they integrate into complex, multi-agent, and decentralized environments such as Web3 and DeAI.

Code and Data Availability

The complete Parakletos implementation (Anchor program in Rust, TypeScript client, and Mocha test suite) is open-source under the Apache-2.0 license at `https://github.com/O2oIO/parakletos` (release v1.0.0). The deployed program is independently verifiable on Solana Devnet under Program ID `FsRpXPvwNbCaCU3CwC9UW9eFrwJYLCaq4hFVFhXyNd3w`; the finalized Address Lookup Table from the 120-node session is at address `5zGKvvDZS1tM3JTqKqds1yhwSrMLxARJ4wczTQecrPiH`.

A Deployment Configuration

The experimental quorum described in Section 3.4 was deployed as a closed-loop two-node physical network: a primary node (MacBook Pro, Apple M3 Pro) and a secondary validation node (MacBook Air), connected via TCP/IP over Thunderbolt on an isolated private subnet (primary 192.168.100.1, secondary 192.168.100.2). Registry REST services operate on port 8080; the o2O Hallway Protocol for node discovery and inter-node messaging operates on port 9090. These parameters are specific to the proof-of-concept deployment and are configurable in the released implementation.

The canonical test suite invocation against Devnet is:

```
ANCHOR_PROVIDER_URL=https://api.devnet.solana.com \
ANCHOR_WALLET=<wallet.json> \
./node_modules/.bin/tsx ./node_modules/mocha/bin/mocha \
  -t 1000000 tests/parakletos_program.mts
```

References

[1] W3C. (2022). Decentralized Identifiers (DIDs) v1.0. W3C Recommendation. `https://www.w3.org/TR/did-core/`

[2] W3C. (2022). Verifiable Credentials Data Model v1.1. W3C Recommendation. `https://www.w3.org/TR/vc-data-model/`

[3] Trust over IP Foundation. (2021). Introduction to Trust over IP. Linux Foundation. `https://trustoverip.org/introduction-to-trust-over-ip/`

[4] Zhang, Y., et al. (2020). Blockchain for IoT security and privacy: A survey. *IEEE Internet of Things Journal*, 7(6), 4603–4618.

[5] Saberi, S., et al. (2019). Blockchain technology and its relationships to sustainable supply chain management. *International Journal of Production Research*, 57(7), 2117–2135.

[6] Dai, J., & Vasarhelyi, M. A. (2016). Toward blockchain-based accounting and assurance. *Journal of Information Systems*, 30(3), 5–21.

[7] Glushankov, K., Barykin, S. E., & Dinets, D. (2026). Digital Twin models for investment cash flow provenance on distributed ledgers. *Enigma Journal of Finance & AI*, 1(2), 45–58.

[8] Coral Crypto. (2024). Anchor Framework Documentation. `https://www.anchor-lang.com/`

[9] Solana Labs. (2024). Address Lookup Tables (ALTs). Solana Documentation. `https://docs.solana.com/developing/lookup-tables`

[10] Antonopoulos, A. M., & Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media.

DOI: 10.66693/finai.1033
Received: May 8, 2026 Accepted: August 6, 2026 Published: August 6, 2026

[11] Rodriguez Garzon, S., et al. (2025). AI Agents with Decentralized Identifiers and Verifiable Credentials. arXiv:2511.02841.

[12] Saavedra, D. R. (2026). Interoperable Architecture for Digital Identity Delegation for AI Agents with Blockchain Integration. arXiv:2601.14982.

[13] Karim, M. M., Hoang Van, D., Khan, S., Qu, Q., & Kholodov, Y. (2025). AI Agents Meet Blockchain: A Survey on Secure and Scalable Collaboration for Multi-Agents. *Future Internet*, 17(2), 57. https://doi.org/10.3390/fi17020057