Attestix: A Unified Attestation Infrastructure for Autonomous AI Agents

Pavan Kumar Dubasi
VibeTensor Private Limited
Email: pkd@vibetensor.com

***Abstract*—The rapid proliferation of autonomous AI agents operating across organizational boundaries introduces fundamental challenges in identity verification, trust establishment, and regulatory compliance. Existing identity and access management frameworks, designed for human users and static machine credentials, are inadequate for dynamic, capability-driven agent ecosystems. We present Attestix, an open-source attestation infrastructure that provides a unified trust layer for AI agents through nine integrated modules: cryptographic identity management, W3C Verifiable Credentials, Decentralized Identifiers (DIDs), UCAN-based delegation chains, exponential-decay reputation scoring, automated EU AI Act compliance, data provenance tracking with tamper-evident audit logs, and blockchain anchoring via the Ethereum Attestation Service. Attestix is implemented as a Model Context Protocol (MCP) server exposing 47 tools, enabling any MCP-compatible AI agent to acquire, present, and verify cryptographic attestations without modifications to the agent's core logic. We describe the system architecture, detail the cryptographic pipeline built on Ed25519 signatures with RFC 8785 JSON canonicalization, and present evaluation results demonstrating sub-millisecond credential operations. Attestix bridges the gap between emerging agent communication protocols (Google A2A, Anthropic MCP) and established trust standards (W3C VC, DID, UCAN), providing the first integrated compliance-aware attestation framework purpose-built for agentic AI.**

***Index Terms*—AI agent identity, verifiable credentials, decentralized identifiers, EU AI Act, attestation, MCP, delegation, trust infrastructure**

I. Introduction

The emergence of autonomous AI agents as first-class participants in digital ecosystems represents a paradigm shift in software architecture. Modern agents, powered by large language models (LLMs), independently browse the web, execute code, manage financial transactions, and coordinate with other agents across organizational boundaries [1]. Unlike traditional software services with static API keys and OAuth tokens, these agents exhibit dynamic capabilities, evolving behaviors, and multi-party delegation chains that existing identity frameworks cannot adequately represent.

This shift creates three critical gaps:

Identity Gap. When Agent A receives a request from Agent B, there is no standardized mechanism for B to prove its identity, capabilities, or authorization chain. Current approaches rely on platform-specific API keys that conflate authentication with authorization and provide no provenance information [2].

Compliance Gap. The EU Artificial Intelligence Act (Regulation 2024/1689), enforceable from August 2026, mandates risk classification, conformity assessment, technical documentation, and post-market monitoring for high-risk AI systems [5]. No existing agent framework automates these obligations at the infrastructure level.

Trust Gap. Agent-to-agent interactions lack the cryptographic verifiability that human-to-service interactions enjoy through TLS certificates and OAuth. When agents operate in open, multi-stakeholder environments, trust must be established through verifiable evidence rather than platform reputation alone [3].

We present Attestix, an open-source attestation infrastructure that addresses all three gaps through a unified architecture. Attestix operates as a Model Context Protocol (MCP) server, exposing 47 cryptographic tools across nine modules that any MCP-compatible agent can invoke. The key insight is that attestation should be a *composable infrastructure layer* rather than an application-level concern, analogous to how TLS provides transport security without requiring applications to implement cryptographic primitives.

A. Contributions

- A **unified attestation architecture** integrating DID-based identity, W3C Verifiable Credentials, UCAN delegation, reputation scoring, EU AI Act compliance automation, provenance tracking, and blockchain anchoring into a single coherent framework.
- A **compliance automation engine** that implements Articles 5, 9–15, 43, 72–73, and Annex V of the EU AI Act, including automated risk classification, conformity assessment enforcement, and declaration of conformity generation.
- An **open-source implementation**[1] as an MCP server with 47 tools, demonstrating that comprehensive attestation can be delivered as a protocol-level service with sub-millisecond cryptographic operations.
- A **tamper-evident audit mechanism** combining SHA-256 hash-chained logs with Merkle tree aggregation and on-chain anchoring via the Ethereum Attestation Service on Base L2.

II. Background and Related Work

A. Agent Communication Protocols

Two recent protocols define the emerging agent interoperability landscape. The **Model Context Protocol (MCP)**,

[1]Source code: https://github.com/VibeTensor/attestix

introduced by Anthropic, provides a standardized interface for LLM-based agents to discover and invoke external tools via JSON-RPC over stdio or HTTP [9]. Google's **Agent-to-Agent (A2A) Protocol** defines agent cards, capability discovery, and inter-agent messaging through `/.well-known/agent.json` endpoints [10]. Neither protocol addresses identity verification, credential exchange, or compliance, which Attestix provides as a complementary layer.

B. Decentralized Identity Standards

The W3C **Decentralized Identifiers (DID) Core** specification [7] defines a URI scheme (`did:method:identifier`) enabling verifiable, self-sovereign identity without centralized registries. The `did:key` method encodes public keys directly in the identifier, requiring no external resolution infrastructure. The W3C **Verifiable Credentials Data Model** [6] provides a standard format for cryptographically signed claims, supporting selective disclosure and multi-party verification.

C. Authorization and Delegation

User Controlled Authorization Networks (UCAN) [8] extend the JWT format with capability-based delegation, enabling chained authorization without centralized token issuers. UCAN tokens include an `att` (attenuation) field specifying delegated capabilities and a `prf` (proof) field linking to parent tokens, forming a verifiable delegation graph.

D. AI Governance and Compliance

The **EU AI Act** (Regulation 2024/1689) establishes a risk-based regulatory framework for AI systems [5]. High-risk systems must implement risk management (Article 9), data governance (Article 10), technical documentation (Article 11), record-keeping (Article 12), transparency (Article 13), human oversight (Article 14), and accuracy guarantees (Article 15). Third-party conformity assessment is required for high-risk systems (Article 43).

E. Related Systems

Several recent works address subsets of the agent trust problem. Garzon et al. [3] propose equipping agents with DIDs and VCs for authentication, demonstrating feasibility but without compliance integration or reputation scoring. Huang et al. [2] present a zero-trust identity framework using DIDs and VCs with an Agent Naming Service, but focus on access control rather than regulatory compliance. Bodea et al. [4] address hardware-level trust through Trusted Execution Environments (TEEs), complementary to Attestix's cryptographic approach. Chan et al. [1] survey agent infrastructure needs broadly, identifying identity binding, attribution, and interaction shaping as key functions.

Attestix differentiates itself by providing a *unified* implementation integrating all these concerns into a single deployable service, with specific focus on automated EU AI Act compliance that no prior system addresses.

III. System Architecture

Attestix follows a layered architecture with four tiers: a cryptographic foundation, a service layer implementing domain logic, a tool layer exposing MCP-compatible interfaces, and a persistence layer.

A. Module Overview

Table I summarizes the nine modules and their tool counts.

TABLE I
ATTESTIX MODULES AND TOOL DISTRIBUTION.

Module	Tools	Primary Standard
Identity	8	UAIT v0.1.0
Agent Cards	3	Google A2A Protocol
DIDs	3	W3C DID Core
Delegation	4	UCAN v0.9.0
Reputation	3	Exponential Decay
Compliance	7	EU AI Act (2024/1689)
Credentials	8	W3C VC Data Model 1.1
Provenance	5	SHA-256 Hash Chains
Blockchain	6	EAS on Base L2
Total	**47**	

B. Cryptographic Foundation

All operations use **Ed25519** (EdDSA over Curve25519) exclusively, chosen for deterministic signatures, compact 64-byte output, timing-attack resistance, and widespread DID ecosystem adoption.

1) Key Management: The server generates a single Ed25519 signing key at first launch. When `ATTESTIX_KEY_PASSWORD` is set, the private key is encrypted using Fernet with a PBKDF2-HMAC-SHA256 derived key (480,000 iterations). The DID derived from this key serves as the server's verifiable identity across all issued credentials.

2) JSON Canonicalization and Signing: Attestix implements RFC 8785 (JSON Canonicalization Scheme) for deterministic signing: (1) Unicode normalization to NFC, (2) lexicographic key sorting with compact separators, (3) whole-number float conversion per RFC 8785, (4) UTF-8 encoding, (5) Ed25519 signature encoded as base64url.

3) DID:key Derivation: Agent DIDs follow the `did:key` method:

$$\texttt{did:key:z}\|\text{Base58btc}(\texttt{0xED01}\|\text{pubkey}_{32}) \quad (1)$$

where `0xED01` is the Ed25519 multicodec prefix and `z` indicates base58btc multibase encoding.

C. Identity Layer

Each agent identity is represented as a **Unified Agent Identity Token (UAIT)**, a signed JSON document containing: unique identifier (`attestix:<hex16>`), display name, capabilities list, source protocol, issuer DID, temporal bounds (default 365 days), and an Ed25519 signature over immutable fields.

The signature design separates immutable creation-time fields from mutable operational fields (reputation score, compliance status, revocation), allowing updates without signature invalidation.

Identity translation bridges four formats: UAIT to A2A Agent Card, DID Document, OAuth claims, and canonical form, enabling cross-protocol interoperability.

D. Credential Layer

Credentials follow the **W3C Verifiable Credentials Data Model 1.1** with Ed25519Signature2020 proofs. The issuance pipeline: (1) construct credential with `@context`, `type`, `issuer` (server DID), and `credentialSubject`; (2) canonicalize excluding `proof` and `credentialStatus`; (3) sign with Ed25519; (4) attach proof with `verificationMethod` (full DID fragment URI) and `proofPurpose: assertionMethod`.

Verification checks: existence, non-revocation, non-expiry, and cryptographic signature validity.

Verifiable Presentations bundle credentials with `authentication` proof purpose and replay protection via `challenge`/`domain` fields. Five credential types are pre-defined for compliance workflows.

E. Delegation Layer

Delegation uses **UCAN v0.9.0** tokens as JWTs with EdDSA signatures, including capability attenuation (`att`), proof chains (`prf`), and cryptographic token IDs (`jti`). Chain verification recursively validates parent tokens and confirms capability subset invariants.

F. Compliance Layer

The compliance module automates twelve EU AI Act obligations across four risk levels. Table II maps implemented articles.

TABLE II
EU AI ACT ARTICLE IMPLEMENTATION.

Article	Obligation	Mechanism
Art. 5	Prohibited practices	Rejection at creation
Art. 9	Risk management	Tracking
Art. 10	Data governance	Provenance module
Art. 11	Technical documentation	Lineage recording
Art. 12	Record-keeping	Hash-chained log
Art. 13	Transparency	Credential
Art. 14	Human oversight	Audit flag
Art. 15	Accuracy/robustness	Checklist
Art. 43	Conformity assessment	3rd-party enforced
Art. 72	Post-market monitoring	Tracking
Art. 73	Incident reporting	Tracking
Annex V	Declaration	12-field generation

A key design decision: the system **actively rejects self-assessment for high-risk AI systems** (Article 43), requiring third-party conformity assessment. Gap analysis computes compliance completion as a percentage, enabling continuous monitoring.

G. Reputation Layer

Trust scores use an exponential decay model:

$$S = \frac{\sum_{i=1}^{n} w_i \cdot e^{-\lambda \cdot (t_{now} - t_i)}}{\sum_{i=1}^{n} e^{-\lambda \cdot (t_{now} - t_i)}} \tag{2}$$

where $w_i \in \{0.0, 0.2, 0.5, 1.0\}$ maps to outcomes (failure, timeout, partial, success), $\lambda = \ln(2)/(30 \times 86400)$ gives a 30-day half-life, producing scores in $[0, 1]$.

H. Provenance and Audit Layer

Two mechanisms serve EU AI Act Articles 10–12:

Provenance entries: structured records for training data (Article 10) and model lineage (Article 11).

Hash-chained audit log: each entry includes:

$$H_i = \text{SHA256}(H_{i-1} \| \text{canonical}(\text{entry}_i)) \tag{3}$$

where $H_0 =$

I. Blockchain Anchoring

Attestix anchors artifact hashes on **Base L2** using the **Ethereum Attestation Service (EAS)**: (1) SHA-256 hash of canonical JSON, (2) EAS attestation with schema `bytes32 artifactHash, string artifactType, string artifactId, string issuerDid`, (3) for audit batches, Merkle root computation (RFC 6962 with domain separation) before anchoring. Integration is optional with graceful degradation.

IV. Implementation

Attestix is implemented in Python 3.10+ using `FastMCP`. The codebase comprises 5,596 lines of production code across 31 files, 166 test cases, and 5 example scripts.

A. MCP Integration

The 47 tools are registered via module-level `register(mcp)` functions with `@mcp.tool()` decorators. stdout is redirected to stderr to prevent JSON-RPC protocol corruption.

B. Security Measures

SSRF Protection: outbound HTTP passes through validation blocking private IPs, localhost, metadata endpoints, with DNS pinning against rebinding. **Error Sanitization**: responses contain only exception type names. **GDPR**: `purge_agent_data()` cascades deletion across all stores. **Storage**: atomic writes with file locking and corruption recovery.

V. Evaluation

A. Functional Correctness

The test suite includes 121 unit tests and 45 end-to-end tests covering multi-agent delegation, enterprise compliance, blockchain anchoring, and persona simulations (healthcare, finance, content moderation, supply chain). All examples verified in clean Docker containers.

TABLE III
OPERATION LATENCY (MEDIAN, 1000 RUNS).

Operation	Latency
Ed25519 key generation	<1 ms
JSON canonicalization	<1 ms
Ed25519 sign/verify	<1 ms
Identity creation (UAIT)	∼2 ms
Credential issuance (W3C VC)	∼3 ms
Credential verification	∼1 ms
UCAN token creation	∼2 ms
Merkle root (100 leaves)	∼5 ms
Blockchain anchor (Base L2)	3–8 s

B. Performance

All local operations complete sub-millisecond to low-millisecond. Blockchain latency is dominated by network round-trip, acceptable for non-real-time workflows.

C. Standards Compliance

TABLE IV
STANDARDS AND SPECIFICATIONS.

Standard	Coverage
W3C VC Data Model 1.1	Full lifecycle
W3C DID Core 1.0	did:key, did:web, resolver
Ed25519Signature2020	VC/VP proofs
UCAN v0.9.0	Delegation chains
RFC 8785 (JCS)	JSON canonicalization
RFC 6962 Sec. 2.1	Merkle trees
EU AI Act (2024/1689)	12 articles + Annex V
EAS / EIP-1559	On-chain anchoring
A2A Protocol	Agent cards
MCP	47-tool server

VI. DISCUSSION

A. Comparison with Related Work

TABLE V
FEATURE COMPARISON.

Feature	Ours	[3]	[2]	[4]	[1]
DID/VC	✓	✓	✓	–	Concept
Delegation	UCAN	–	ABAC	–	Concept
EU AI Act	Full	–	–	–	–
Reputation	✓	–	–	–	–
Provenance	✓	–	–	–	Concept
Blockchain	✓	–	–	–	–
TEE	–	–	–	✓	–
MCP	47	–	–	–	–
Open Source	✓	Proto	–	–	–

B. Limitations and Future Work

Current limitations include: single-server trust model (planned: distributed key management), JSON file storage (planned: database backend), static reputation parameters (planned: adaptive Bayesian inference), and no TEE integration. Future work includes zero-knowledge selective disclosure, multi-regulatory framework support (NIST AI RMF, ISO/IEC 42001), and threshold signing for distributed deployments.

VII. CONCLUSION

We presented Attestix, the first integrated attestation infrastructure for autonomous AI agents combining cryptographic identity, verifiable credentials, capability delegation, regulatory compliance, reputation scoring, provenance tracking, and blockchain anchoring in a single MCP server. The system implements twelve EU AI Act articles, provides sub-millisecond operations, and bridges emerging agent protocols with established trust standards. Attestix is available as open-source under Apache 2.0 at https://github.com/VibeTensor/attestix and on PyPI as `attestix`.

ACKNOWLEDGMENTS

This work was conducted at VibeTensor Private Limited. The author thanks the W3C, UCAN, and MCP communities for foundational standards.

REFERENCES

[1] A. Chan, K. Wei, S. Huang, N. Rajkumar, E. Perrier, S. Lazar, G.K. Hadfield, and M. Anderljung, "Infrastructure for AI Agents," *arXiv preprint arXiv:2501.10114*, 2025.

[2] K. Huang *et al.*, "A Novel Zero-Trust Identity Framework for Agentic AI: Decentralized Authentication and Fine-Grained Access Control," *arXiv preprint arXiv:2505.19301*, 2025.

[3] S.R. Garzon, A. Vaziry, E.M. Kuzu, D.E. Gehrmann, B. Varkan, A. Gaballa, and A. Küpper, "AI Agents with Decentralized Identifiers and Verifiable Credentials," in *Proc. 18th ICAART*, 2026.

[4] T. Bodea *et al.*, "Trusted AI Agents in the Cloud," *arXiv preprint arXiv:2512.05951*, 2025.

[5] European Parliament, "Regulation (EU) 2024/1689 (Artificial Intelligence Act)," *Official Journal of the EU*, 2024.

[6] M. Sporny, D. Longley, D. Chadwick, and O. Terbu, "Verifiable Credentials Data Model v1.1," W3C Recommendation, 2022.

[7] M. Sporny *et al.*, "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, 2022.

[8] B. Frazee, D. Holmgren, and P. Willison, "UCAN Specification v0.9.0," 2023. [Online]. Available: https://ucan.xyz

[9] Anthropic, "Model Context Protocol Specification," 2024. [Online]. Available: https://modelcontextprotocol.io

[10] Google, "Agent-to-Agent (A2A) Protocol," 2025. [Online]. Available: https://google.github.io/A2A/

[11] P.K. Dubasi, "Attestix: Attestation Infrastructure for AI Agents," VibeTensor, 2026. [Online]. Available: https://github.com/VibeTensor/attestix