ENS: A Naming Layer for AI Agent Identity

Technical Brief in Response to the NCCoE Concept Paper on
Software and AI Agent Identity and Authorization

Prepared by `estmcmxci.eth` on behalf of the ENS Ecosystem

`https://discuss.ens.domains/c/ens-dev/ai/79`

April 2, 2026

Core Thesis

Every identity stack needs a naming layer. In the traditional internet, DNS provides this. For AI agents operating across heterogeneous blockchain and web environments, the Ethereum Name Service (ENS) provides it.

ENS is open, permissionless naming infrastructure operational on Ethereum since 2017, with over two million registered names. It maps human-readable identifiers (e.g., `agent.org.eth`) to cryptographic owners, on-chain metadata, and service endpoints—the same function DNS performs for the web, with three properties DNS lacks.

What ENS Adds Beyond DNS

1. **Self-service registration with no registrar bottleneck.** Any agent with an Ethereum account can register or update an ENS name in a single transaction. There is no ICANN, no approval queue, and no human intermediary. An agent can provision its own identity, attach metadata, and begin operating—autonomously.
2. **Cryptographically provable ownership.** Every ENS name has an on-chain owner: an Ethereum account whose control is provable via ECDSA signatures (or smart-contract signatures under EIP-1271). Any verifier can confirm ownership by checking the on-chain registry and requesting a signature—no certificate authority required.
3. **Globally readable, auditable records.** ENS records live on a public blockchain. Every update is a timestamped, immutable transaction. Any party—auditor, regulator, peer agent—can read an agent's identity records and verify their history without special access, making policy enforcement and compliance logging straightforward by default.

How ENS Addresses NIST's Six Question Areas

1. General considerations. ENS is not a proposal; it is deployed infrastructure. AI agents in the blockchain ecosystem already use ENS names for governance, trading, and cross-chain coordination. ENS sits *below* the communication and authorization protocols NIST identifies (MCP, OAuth, SPIFFE)—it tells those protocols *who* the agent is, just as DNS tells HTTP *where* the server is.

2. Identification. An ENS name gives every agent a single, human-readable, cross-chain identifier. Subnames (`agent.org.eth`) model organizational hierarchy: the parent controls delegation, and any client can verify the relationship on-chain. Agents can attach structured metadata—capabilities, classification, endpoints—directly to their name as extensible text records.

3. Authentication. Because ENS ownership is an Ethereum account, authentication reduces to cryptographic signature verification. An agent proves it controls `myagent.eth` by signing a challenge with the owner key (ECDSA or EIP-1271 for smart-contract wallets). No certificate authority or federation trust is required—verification is deterministic and self-contained.

4. Authorization. ENS provides the identity that authorization protocols act on; it does not replace them. OAuth 2.0 and OIDC operate above the naming layer and can reference ENS identities the same way they reference domain names today. ENS subname hierarchies enforce delegation boundaries cryptographically: `agent.org.eth` can only be issued by the controller of `org.eth`.

5. Auditing and non-repudiation. Every ENS record update is an on-chain transaction with a block timestamp and a sender address. Ownership history is publicly queryable at any block height. This provides

a tamper-proof audit trail for all identity lifecycle events—registration, metadata changes, ownership transfers, and revocation—without requiring a dedicated logging service.

6. Prompt injection and metadata integrity. ENS does not operate at the LLM prompt layer, but it reduces the attack surface for identity-layer injection. Because records are signed by the owner and stored on-chain, an attacker cannot silently modify an agent's identity metadata. Ongoing standards work adds schema validation that rejects malformed metadata before it reaches agent runtime.[1]

Ongoing Standards Work

ENS protocol developers are extending the production naming layer with agent-specific capabilities through open, CC0-licensed standards processes. Four ENS Improvement Proposals (ENSIPs) and related specifications address cross-chain agent registry verification, protocol-specific endpoint discovery, typed metadata classification, and cryptographically signed identity manifests with version lineage.[2]

Four of seven stack layers are deployed in production on Ethereum mainnet today. The remaining three are under active development and build directly on the production foundation.

Next Steps

We invite NIST and NCCoE participants to explore the ENSIP drafts and the companion paper for deeper integration paths. ENS occupies a narrow, well-defined role—the naming layer—and is designed to compose with the protocols NIST has already identified, not compete with them.

All specifications are open-source and CC0 licensed. We welcome collaboration as technology contributors to the NCCoE demonstration effort.

Contact: ENS Standards Authors — AI Agent Identity Working Group
`https://discuss.ens.domains/c/ens-dev/ai/79`

Companion paper: *ENS as a Naming Layer for AI Agent Identity: Production Infrastructure for Agent Identification, Authentication, and Authorization* (12 pp., same submission)

[1] The Node Metadata Standard (NMS) and Agent Identity Profile (AIP) drafts define JSON Schema validation and signed manifests with a default-deny verification posture. See the companion paper for details.

[2] Key references: ENSIP-24 (arbitrary data resolution), ENSIP-25 (agent registry verification), ENSIP-26 (routing and discovery), NMS (node metadata classification), AIP (agent identity profiles), ERC-8004 (on-chain agent registry). Full technical treatment in the companion paper: *ENS as a Naming Layer for AI Agent Identity: Production Infrastructure for Agent Identification, Authentication, and Authorization.*