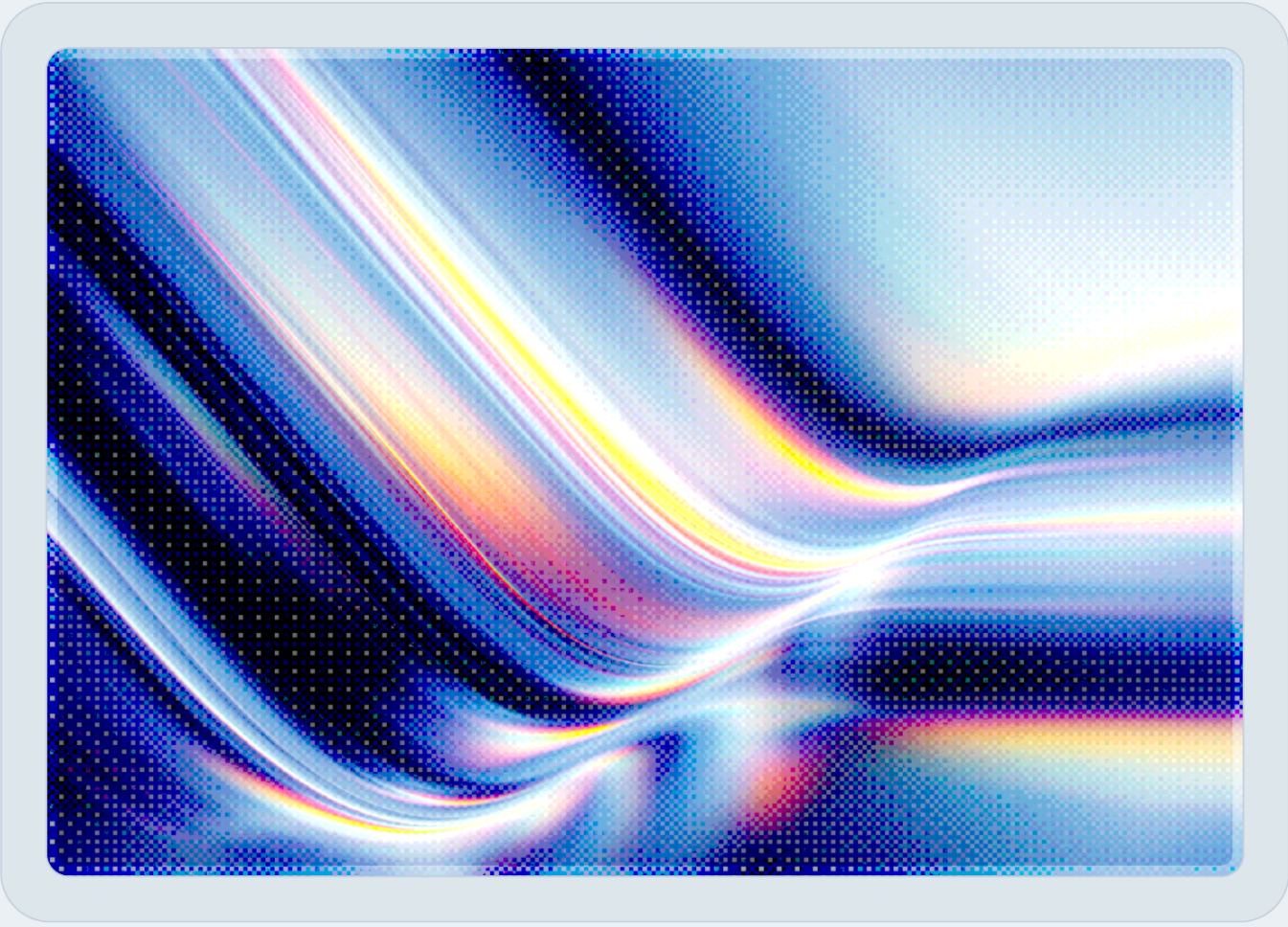
CSAI Foundation | Cloud Security Alliance

Anchoring Agent Identity in DNS: A Security Analysis of the Agent Name Service (ANS)

What the Linux Foundation's ANS Means for AARM, the Agentic Trust Framework, MAESTRO, and the AI Controls Matrix

2026-06-25

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Table of Contents

Executive Summary

On June 23, 2026, the Linux Foundation announced its intent to launch the Agent Name Service (ANS), an open standard that anchors the identity and discovery of autonomous AI agents in the Domain Name System and a supporting public-key infrastructure [1][2]. The announcement frames ANS as the missing trust layer for an agentic economy in which agents must locate one another, verify who they represent, and confirm that an agent's code and operational history are authentic before they transact at machine speed. The standard did not appear from nowhere: ANS began as a 2025 research paper and an effort within the OWASP GenAI Security Project's Agentic Security Initiative, and its co-authors include figures already central to Cloud Security Alliance research — most notably Ken Huang, author of CSA's MAESTRO threat-modeling framework [3][6]. CSA is therefore not an outside observer of ANS; its own contributors helped create it, and the ANS paper itself uses MAESTRO to model its threats.

This paper analyzes ANS from a security perspective and assesses its impact on four bodies of CSA work that together define the agentic control plane: Autonomous Action Runtime Management (AARM), the Agentic Trust Framework (ATF), MAESTRO, and the AI Controls Matrix (AICM). Our central finding is that ANS addresses a real and urgent gap — agents today routinely borrow human or shared identities rather than holding distinct, verifiable ones [36] — and that a capability-aware, cryptographically verifiable discovery fabric is a genuine advance over ad-hoc agent discovery. ANS supplies precisely the verifiable identity substrate that AARM presupposes but does not define, that the Agentic Trust Framework requires for its "Who are you?" element, and that AICM's identity controls assume exists.

At the same time, ANS inherits the accumulated security debt of the two centrally rooted hierarchies it builds upon. The standard markets itself as a federated alternative to "centralized or proprietary registries," yet its own specification roots trust in the DNS hierarchy and in the X.509 certificate-authority system [4]. The dependency it leans on most heavily to make name-to-key bindings tamper-evident — DNSSEC — is validated on roughly a third of resolution paths but cryptographically signed on only about seven percent of zones, two decades after its introduction [15]. Classic DNS attacks (cache poisoning, registrar compromise, route hijacking, denial of service, and homograph spoofing) do not merely persist in an agent-identity system built on DNS; they become more consequential, because automated agents make trust decisions programmatically and at scale, without the human cues that sometimes catch these attacks today.

For CSA, ANS is best understood not as a competitor to its frameworks but as a new, security-critical layer that those frameworks must now account for. This paper recommends that CSA treat ANS as one candidate identity fabric among several — alongside W3C Decentralized Identifiers, SPIFFE workload identity, and OAuth-based delegation — and that it preserve framework neutrality by positioning AICM as the control catalog that maps across all of them. We close with a concrete research agenda: a fail-closed trust profile for agent-identity resolution, revocation-propagation guidance that reconciles certificate revocation with

the Agentic Trust Framework's demotion semantics, candidate AICM control objectives for discovery-fabric integrity and discovery-metadata privacy, an extension of the Agentic AI Red Teaming Guide to registry and certificate-authority compromise, and Observatory measurement of agent-identity adoption.

Introduction: What Was Announced

The Linux Foundation's announcement positions ANS as foundational infrastructure for an agentic ecosystem that enterprises are adopting faster than they can govern. The Foundation cites World Economic Forum data indicating that a large majority of executives intend to deploy AI agents within one to three years despite widespread uncertainty about how to securely evaluate and manage them [1]. ANS proposes to resolve part of that uncertainty by giving every agent a verifiable identity and a discoverable, capability-aware name, anchored in DNS infrastructure that already processes on the order of a hundred million queries per second [2].

It is important to separate what ANS is today from how it is being marketed. The technical substance of ANS lives in a May 2025 academic paper, "Agent Name Service (ANS): A Universal Directory for Secure AI Agent Discovery and Interoperability," authored by Ken Huang, Vineeth Sai Narajala, Idan Habler, and Akram Sheriff, and in the corresponding IETF Internet-Draft [3][4]. That work describes a system that is "DNS-inspired" — it borrows DNS's hierarchical naming and delegation concepts and uses a dedicated Agent Registry as its resolution endpoint — rather than one that encodes agent identity directly into live DNS resource records. The Linux Foundation's 2026 framing leans harder into literal DNS integration and adds the vision of incorporating Decentralized Identifiers (DIDs) and Legal Entity Identifiers (LEIs) into a unified verification model [1][2]. The concrete DNS record encoding, the DNSSEC deployment profile, and the DID and LEI grammars implied by the launch messaging are not yet specified in the primary sources; they should be read as roadmap, not as defined wire formats. The project is at an "intent to launch" stage, pre-1.0, with development to continue in a Linux Foundation GitHub organization and a reference implementation that currently uses mock cryptography for demonstration rather than production-grade PKI [7].

The roster of backing organizations is itself analytically relevant. Named supporters include GoDaddy, Cloudflare, Infoblox, Cisco, Salesforce, and Hashgraph Online, alongside DistributedApps.ai and OWASP [8]. Several of the most prominent backers are commercial DNS incumbents — a registrar, a DNS-and-network-security vendor, and a DNS appliance vendor — whose businesses stand to benefit from extending DNS into agent identity. That alignment is not disqualifying, but it is a lens through which independent observers have already viewed the proposal, and CSA's analysis should acknowledge it openly [9].

This paper proceeds in four movements. It first describes how ANS works in enough technical detail to support a rigorous security assessment. It then analyzes ANS on its own security merits, with particular attention to the consequences of rooting agent identity in DNS and X.509. It then assesses what ANS

means for each of AARM, the Agentic Trust Framework, MAESTRO, and AICM. Finally, it sets out the research CSA should perform as a result.

How ANS Works

Naming and Capability-Aware Discovery

ANS gives each agent a structured, human-readable name that encodes not just who the agent is but what it does and how to speak to it. The grammar defined in the specification composes a protocol, an agent identifier, a capability descriptor, a provider, and a semantic version, with an optional extension field [3][4]:

```
ANSName = Protocol "://" AgentID "." agentCapability "." Provider
".v" Version ["." Extension]
```

A concrete example – `a2a://textProcessor.DocumentTranslation.AcmeCorp.v2.1.hipaa` – reads as a document-translation agent operated by AcmeCorp, reachable over the A2A protocol at version 2.1, with a deployment extension signaling a HIPAA-relevant configuration. The protocol field is drawn from an enumerated set that today includes A2A, MCP, and ACP, so that a single naming scheme can span heterogeneous agent ecosystems. Versions must follow semantic-versioning rules, which lets a requesting agent ask for a compatible range rather than a single exact build. This capability-aware, version-aware naming is the feature that most distinguishes ANS from generic DNS service discovery, which resolves names to addresses but carries no semantic understanding of agent capability or compatibility [3].

Resolution Flow

Resolution in ANS is a directory lookup against an Agent Registry rather than a chain of ordinary DNS queries, and it is deliberately structured so that every answer is cryptographically verifiable before it is used. A requesting agent submits a capability request specifying the protocol, agent identifier, capability, provider, and a requested version range. The registry matches candidate records, negotiates the highest compatible version using a range model borrowed from semantic versioning, and returns an endpoint record consisting of the endpoint data, a signature, and a certificate. The client then verifies the registry's signature over the endpoint data, validates the certificate chain back to a trusted certificate authority – including a revocation check via certificate revocation lists or the Online Certificate Status Protocol – and only then treats the

endpoint as authentic [4]. The registry returns a time-to-live with each answer, with a recommended default of three hundred seconds, after which resolvers must revalidate. This caching behavior, ordinary and sensible for a directory service, becomes security-relevant when we consider revocation latency later in this paper.

Identity, Cryptography, and Protocol Bridging

The trust model of ANS is conventional public-key infrastructure applied to agents. Each registered agent receives a key pair and an X.509 certificate issued by a certificate authority through a registration authority, binding the agent's public key to its verified identity and organizational affiliation. The trust anchor for the entire system is the certificate authority that issues the registry's own certificate; agents prove identity by signing messages with their private keys, and verifiers check those signatures against the public keys in the agents' certificates [3][4]. Data structures are expressed as JSON Schema. A modular protocol-adapter layer bridges ANS to A2A, MCP, and ACP, parsing each protocol's native metadata — A2A Agent Cards, MCP tool and resource descriptions, and ACP role definitions — and storing protocol-specific data in an extensions object [3].

The following table summarizes the architecture at a glance.

Element	ANS approach	Security-relevant property
Name	Protocol/AgentID/capability/provider/version grammar	Human-readable; squatting and homograph exposure
Resolution	Directory lookup against an Agent Registry	Registry is a high-value, systemic dependency
Identity proof	Per-agent X.509 certificate from a CA via an RA	Inherits CA-system trust assumptions
Trust anchor	CA that issues the registry's certificate	Centrally rooted, not decentralized in practice
Integrity of bindings	Signed registry responses; "DNSSEC-like" mechanisms	DNSSEC is rarely deployed in the real world
Revocation	CRL / OCSP, checked during resolution	Propagation latency interacts with TTL caching
Interoperability	Protocol adapters for A2A, MCP, ACP	Broad reach; adapter correctness is critical

Element	ANS approach	Security-relevant property
Roadmap claims	DID and LEI integration, literal DNS anchoring	Not yet specified in primary sources

Security Analysis

The Centralization Paradox

The most important security observation about ANS is structural rather than incidental. ANS is presented as a way to establish a "federated identity layer at internet scale without relying on centralized or proprietary registries" [2], yet its own specification reproduces not one but two centrally governed trust hierarchies. The naming and delegation model is rooted, as DNS is, in a single authoritative hierarchy administered through IANA and ICANN. The identity model is rooted in the X.509 certificate-authority system, in which any trusted certificate authority can, in principle, issue a valid certificate for any name — a structural weakness demonstrated repeatedly in the web PKI, from the 2011 DigiNotar compromise that enabled the interception of hundreds of thousands of users' traffic to the large-scale Symantec mis-issuance that led browsers to progressively distrust that authority [21][22][23]. Mechanisms such as Certificate Transparency and Certification Authority Authorization records mitigate these failures but do not remove the central roots; Certificate Transparency makes mis-issuance detectable after the fact, and CAA records are themselves DNS records, folding the constraint back into the DNS hierarchy.

This matters because the rest of the agentic field is moving in the opposite direction. W3C Decentralized Identifiers are defined explicitly to be "decoupled from centralized registries, identity providers, and certificate authorities," allowing a controller to prove control of an identifier without permission from any other party [24]. CSA's own agentic-IAM research recommends DIDs and Verifiable Credentials as the basis for agent identity [35]. ANS, by contrast, reintroduces precisely the central roots that decentralized-identity work is trying to escape, and binds agent identity to domain ownership — a relationship that is mutable in ways agent identity should not be, since domains expire, transfer, and change hands while the agent they name persists. This is not a fatal objection, but it is a paradox that CSA should name plainly: the "decentralized" framing of ANS describes its federation of operators, not the rooting of its trust.

DNS Threats, Transferred and Amplified

The classic catalog of DNS attacks does not stay in the network layer when DNS becomes the substrate for agent identity; each attack acquires a new and more serious meaning.

Cache poisoning and response spoofing have been practical since the Kaminsky disclosure of 2008, and were revived by the SAD DNS side-channel technique in 2020, which defeated source-port randomization and left a large fraction of open resolvers — including major public ones — exploitable for a period [13][14]. In an ANS context, a successfully poisoned lookup does not merely send a user to the wrong website; it causes an autonomous agent to discover and authenticate an impostor, and then to exchange credentials, instructions, or sensitive data with it, with no human in the loop to notice that something is amiss. The standard's answer to this class of attack is signed responses and "DNSSEC-like" integrity, but here the empirical record is sobering. Two decades after its introduction, DNSSEC is validated on roughly a third of resolution paths yet cryptographically signed on only about seven percent of zones, and in the largest top-level domain the proportion of signed names is in the low single digits [15]. The decisive question for ANS is therefore not whether it can use DNSSEC but whether it mandates signing and validation and fails closed when a name resolves through an unsigned zone or when revocation status cannot be confirmed. If ANS fails open in those conditions — as most of the deployed DNS ecosystem effectively does — the integrity guarantee becomes the exception rather than the rule.

Several other DNS attack classes bypass zone-level integrity entirely. Registrar and registry compromise, exemplified by the Sea Turtle campaign that altered name-server records for dozens of organizations by attacking the control plane rather than the protocol, would let an adversary legitimately re-point or re-issue an agent's identity records — a valid takeover that signature checks alone will not catch [16]. Route hijacking operates a layer lower still: the 2018 BGP hijack of Amazon Route 53 redirected DNS resolution to a malicious server for roughly two hours regardless of any cryptographic protections, because it attacked reachability rather than authenticity [17]. Denial-of-service attacks against the resolution fabric, as in the 2016 Dyn outage that took dozens of major platforms offline, would translate in an ANS world into a fleet-wide inability for agents to discover or authenticate peers — a systemic single point of failure that could halt agent-to-agent interaction even while the agents themselves are perfectly healthy, and that creates pressure to fail open under load [19].

Two further properties of DNS deserve specific mention for an identity use case. DNS offers no confidentiality: queries travel in plaintext, and DNSSEC authenticates without hiding, as the DNS privacy RFCs make explicit [18]. The names and patterns of agent-discovery queries would therefore leak to on-path observers and to resolver and registry operators, revealing which agents communicate with which services and exposing the topology of an organization's agent fleet even when message payloads are encrypted. And because an agent's identity is, in part, its human-readable name, homograph and typosquatting attacks gain new force: an adversary can register a name that is visually identical to a legitimate one, and an automated agent making programmatic trust decisions lacks even the imperfect protections a human browser user enjoys [20]. The ANS authors themselves acknowledge name collisions and name squatting as open problems and suggest an ICANN-like governance model in response [3].

Conflict of Interest and the "Yet Another Standard" Problem

Independent commentary has raised two non-technical concerns that CSA's analysis should not omit. The first is a potential conflict of interest: several of the loudest backers of a DNS-based identity standard are companies whose revenue is tied to DNS, and at least one competing proposal originates from a top-level-domain operator [9]. The second is the familiar dynamic in which a new standard intended to unify a fragmented space becomes merely one more entrant; ANS launches into a field that already includes Google's A2A discovery, MCP's authorization model, W3C DIDs, SPIFFE, OAuth-based delegation work, Okta's Cross App Access, and a NIST conceptual effort, several of which are summarized in the table below [9][10][12]. Neither concern is a reason to dismiss ANS, but both are reasons for CSA to engage as a neutral party rather than as an advocate.

Approach	Identity root	Anchored in DNS?	Best understood as
ANS	DNS hierarchy + CA root	Yes (by design)	Discovery + verifiable naming
Google A2A "AgentCard"	HTTPS origin + optional JWS/DID	Yes by default	Capability advertisement
MCP server identity	HTTPS URI + OAuth resource server	Yes to locate; OAuth to trust	Tool/server access
W3C DID / VC	Self-owned cryptographic key	No (except did:web)	Decentralized identity
SPIFFE / SPIRE	Self-defined trust domain + attestation	No	Workload identity for mTLS
OAuth 2 / OIDC + agent drafts	Authorization server / issuer	No	Delegated authorization
Okta Cross App Access	Enterprise identity provider	No	Enterprise-managed agent auth
NIST NCCoE concept	Technology-agnostic; human delegation chain	N/A	Identity/authorization principles

Impact on CSA Frameworks

ANS does not replace any CSA framework; it occupies a layer those frameworks have so far assumed rather than specified. The agentic control plane that CSA and the CSAI Foundation have articulated separates the question of who an agent is and what it is allowed to do (the Agentic Trust Framework), how to enforce decisions at the moment of action (AARM), how to reason about threats (MAESTRO), and which controls to implement (AICM) [37][38][39]. ANS proposes a concrete answer to a sub-question all four depend on: how an agent's identity is established, named, and discovered across organizational boundaries. The sections below assess the consequences for each framework in turn.

Autonomous Action Runtime Management (AARM)

AARM specifies what a conformant runtime must do at the moment an agent attempts an action: intercept it before execution, evaluate it against intent-aware policy, render one of five authorization decisions, and produce a tamper-evident receipt that is cryptographically bound to an agent identity [37]. Two of AARM's nine requirements depend directly on an identity substrate that AARM itself does not define. Requirement R6, Identity Binding, mandates that action receipts be cryptographically bound to an agent identity to enable non-repudiation and unique attribution; requirement R5 requires those receipts to be tamper-evident. ANS is a natural candidate to supply that identity: an agent that holds an ANS-issued X.509 certificate can sign its actions with a key that resolves, through the ANS registry, to a verified organizational affiliation. In this sense ANS and AARM are genuinely complementary — ANS answers "which agent is this," and AARM records "what this agent did and what the runtime decided."

The complementarity comes with a dependency that CSA should make explicit. AARM's non-repudiation guarantee is only as strong as the identity to which receipts are bound. If an agent's ANS identity can be spoofed through DNS poisoning or assumed through registrar compromise, then AARM will faithfully produce authentic-looking receipts attributed to the wrong agent, and the forensic value of those receipts collapses at its root. The weaknesses in the identity fabric therefore propagate directly into AARM's strongest claims. Several of AARM's enumerated threat classes intersect this concern precisely: the confused-deputy threat (T2) and cross-agent propagation (T7) both depend on the trust an agent places in the identity of a peer, and a poisoned discovery result is exactly the mechanism by which a confused-deputy or propagation attack would be seeded at scale. There is also a temporal gap: AARM presumes that a revoked identity stops being trusted, but ANS revocation propagates through CRL/OCSP and is subject to registry TTL caching, so an agent that should no longer be trusted may remain resolvable and verifiable for the duration of a cached answer. CSA should produce guidance on how AARM receipts ought to reference ANS identities and on how identity revocation must propagate to in-flight AARM-governed sessions.

Agentic Trust Framework (ATF)

The Agentic Trust Framework maps even more directly onto ANS. ATF organizes agent governance around five Zero Trust questions, the first of which — "Who are you?" — is its Identity element, which requires that every agent have a verifiable identity bound to an accountable human or owning entity [38]. ANS is, in effect, a candidate implementation of that element: it provides the verifiable cryptographic identity and, through its roadmap incorporation of Legal Entity Identifiers, a path to binding that identity to an owning organization. What ANS does not provide is the binding to an accountable human and the delegation chain that NIST's agent-identity work emphasizes; ANS can tell you which organization stands behind an agent, but not which person authorized it or on whose behalf it currently acts [30]. ATF's Identity element and ANS are thus partial answers to the same question that should be composed rather than conflated.

ATF's defining contribution — a four-level maturity model in which agents earn autonomy through demonstrated trustworthiness and can be demoted to the lowest level immediately upon a critical incident — interacts with ANS in a way that surfaces a concrete research need. Immediate demotion is only meaningful if the rest of the ecosystem stops trusting the demoted agent quickly, and in an ANS deployment that means certificate revocation and registry invalidation must propagate faster than the agent can cause further harm. The default three-hundred-second TTL and the well-known latencies of CRL and OCSP distribution are in tension with the "demote now" semantics ATF requires. ATF already publishes crosswalks to AICM, NIST SP 800-207, ISO/IEC 42001, the AWS Agentic Scoping Matrix, the OWASP Agentic Top 10, and MAESTRO; ANS belongs in that crosswalk as a candidate Identity-element implementation, with explicit notes on its revocation-latency limitations relative to ATF's demotion model.

MAESTRO

MAESTRO models agentic systems across seven layers — foundation models, data operations, agent frameworks, deployment and infrastructure, evaluation and observability, a cross-cutting security-and-compliance layer, and the agent ecosystem — and treats attacks that exploit interactions between layers as a distinct and important category [33]. Agent identity, discovery, and agent-to-agent communication live primarily in Layer 7, the Agent Ecosystem, whose threat catalog already names agent-identity attacks, malicious agent discovery and registry manipulation, and communication-channel attacks. ANS is therefore squarely a Layer 7 construct, with the vertical Security and Compliance layer (Layer 6) as the integrating overlay that governs it.

There is a notable reflexive relationship worth highlighting: the ANS foundational paper conducts its own threat analysis using MAESTRO, and MAESTRO's author is an ANS co-author [3][33]. CSA's framework is thus already the lingua franca for reasoning about ANS, which positions CSA well to lead the analysis. The most valuable MAESTRO-specific contribution CSA can make is to take ANS seriously as a source of new cross-layer threats rather than only a Layer 7 component. A compromised ANS registry is a Layer 7 failure that cascades downward: by returning malicious endpoints, it can redirect agents into attacker-controlled

infrastructure (Layer 4) and steer them toward poisoned data stores and retrieval pipelines (Layer 2), turning a single discovery-fabric compromise into a systemic, multi-layer incident. CSA should extend MAESTRO's Layer 7 threat catalog to include the DNS-anchoring-specific threats analyzed in this paper — poisoning, registrar and certificate-authority compromise, revocation latency, and discovery-metadata leakage — and publish a MAESTRO threat-modeling template for identity-and-discovery fabrics of the ANS type.

AI Controls Matrix (AICM)

AICM provides 243 control objectives across 18 domains and is the natural place to express, in auditable terms, what good looks like for an organization deploying agents under ANS [34]. Several AICM domains map directly. The Identity and Access Management domain is the primary anchor, and it already contains agent-aware controls — notably IAM-19, Agent Access Restriction — alongside foundational controls for unique identification, strong authentication, and authorization mechanisms; an ANS-issued identity is a means of satisfying these for agents. The Cryptography, Encryption and Key Management domain governs the certificate issuance, key protection, and revocation on which ANS depends. The Logging and Monitoring domain covers the discovery-query and resolution logging an organization would need for detection and forensics. The Supply Chain Management, Transparency and Accountability domain captures the trust an organization places in external registries and certificate authorities — a dependency ANS makes load-bearing. The Application and Interface Security domain's agent-boundary control (AIS-11) is also relevant where ANS bridges protocols.

The mapping also exposes a gap. AICM does not yet contain a control objective specific to the integrity of an agent discovery and identity fabric, nor one addressing the confidentiality of discovery metadata — the topology leakage discussed earlier maps conceptually to the Data Security and Privacy domain but is not addressed for agent discovery specifically. The table below summarizes the principal mappings and the gap.

AICM domain	Relevance to ANS	Status
Identity & Access Management (IAM)	Agent identity, unique ID, strong auth, authorization, IAM-19 agent access restriction	Covered
Cryptography, Encryption & Key Mgmt (CEK)	Certificate issuance, key protection, revocation	Covered
Logging & Monitoring (LOG)	Discovery-query and resolution logging for detection/forensics	Covered
Supply Chain, Transparency & Accountability (STA)	Trust in external registries and certificate authorities	Covered

AICM domain	Relevance to ANS	Status
Application & Interface Security (AIS)	Agent security boundaries where ANS bridges protocols (AIS-11)	Partial
Data Security & Privacy (DSP)	Confidentiality of discovery metadata / fleet topology	Gap
(new) Discovery-fabric integrity	Resolution integrity, fail-closed behavior, revocation propagation	Gap

Because AICM is the control basis for the STAR for AI assurance program and the AI-CAIQ self-assessment, closing these gaps would also let CSA express ANS-related expectations as assessable assurance questions over time.

Research Gaps CSA Should Address

The arrival of ANS surfaces a coherent research agenda that spans several CSA programs and working groups. We present it as a set of recommended efforts rather than a prioritized backlog, since prioritization depends on the pace of ANS's own development.

The most urgent gap is the absence of a fail-closed trust profile for agent-identity resolution. Given the empirical state of DNSSEC deployment, CSA should define the conditions under which an ANS resolution must be treated as untrustworthy — resolution through an unsigned zone, an unreachable revocation responder, an expired or unverifiable certificate chain — and should specify fail-closed rather than fail-open behavior as the secure default, with explicit guidance on how to manage the availability trade-off. This work sits naturally with the Best Practices and Zero Trust working groups and builds on CSA's existing Zero Trust guidance.

A second gap concerns revocation propagation and its relationship to the Agentic Trust Framework's demotion semantics. CSA should establish a recommended latency budget for identity revocation and registry invalidation, reconcile it with default TTL behavior, and specify how a demotion event in ATF translates into revocation actions in an ANS deployment so that "immediate demotion to Intern" is meaningful in practice. This is joint work for the stewards of AARM and ATF.

A third gap is in AICM itself. CSA should evaluate candidate control objectives for discovery-fabric integrity and for discovery-metadata confidentiality, and assess whether existing IAM, CEK, and STA controls require agent-discovery-specific guidance. This belongs with the Security Controls Catalog working group and would flow downstream into STAR for AI and the AI-CAIQ.

A fourth gap is comparative and architectural. ANS is one of several candidate identity fabrics, and CSA is well placed — given its existing agentic-IAM research recommending DIDs and Verifiable Credentials — to publish a neutral comparison and a hybrid-architecture pattern that shows how ANS, W3C DIDs, SPIFFE workload identity, and OAuth-based delegation can interoperate rather than compete, and how an organization should choose among them [35][36]. This work would also articulate how the human-accountability and delegation-chain concerns that ANS does not address can be supplied by composing it with other mechanisms.

A fifth gap is adversarial. CSA's Agentic AI Red Teaming Guide should be extended with a registry-and-certificate-authority threat model and a red-team playbook for ANS-style fabrics, covering poisoning, registrar compromise, route hijacking, revocation evasion, and homograph and squatting attacks, so that organizations can test their deployments rather than assume them secure.

Beyond these, three further efforts merit attention. CSA should contribute name-governance recommendations — drawing on its members' collective experience — to the question of squatting, collision, and homograph governance that the ANS authors have flagged as open. CSA should investigate discovery-privacy mitigations such as private information retrieval and query anonymization for agent discovery, an area the ANS threat model gestures at but does not resolve. And CSA's Observatory should measure agent-identity adoption empirically — the degree to which agents hold distinct verifiable identities, the uptake of ANS and competing approaches, and the continued growth of non-human identity sprawl, which already runs at a ratio of machine to human identities on the order of dozens to one and is rising with AI adoption [31][32]. The Observatory's existing surveys on autonomous-agent deployment and identity gaps provide a baseline to extend [36].

Recommendations and Strategic Posture

CSA should engage with ANS actively, early, and neutrally. The case for engagement is strong: ANS addresses a real gap, its intellectual lineage runs directly through CSA contributors and the MAESTRO framework, and CSA's frameworks are already the vocabulary in which ANS reasons about its own threats. The case for neutrality is equally strong: ANS is a pre-1.0 proposal entering a crowded field, its backers include parties with commercial interests in the DNS approach, and the broader trajectory of agentic identity may favor decentralized or hybrid models that ANS does not fully embrace. CSA's most durable contribution is not to endorse a single identity fabric but to position AICM as the neutral control catalog and MAESTRO as the neutral threat-modeling language that map across whichever fabrics the market adopts, while AARM and the Agentic Trust Framework specify, respectively, how decisions are enforced and how trust is governed regardless of the underlying identity mechanism.

Concretely, CSA should open lines of engagement with the Linux Foundation's ANS effort, the OWASP GenAI Security Project from which it sprang, the relevant IETF activity, the NIST NCCoE agent-identity project, and the OpenID Foundation and vendor efforts pursuing OAuth-based agent delegation, so that CSA's control and threat-modeling work is informed by, and informs, the standards as they mature. Internally, CSA should treat the research agenda above as a cross-program effort spanning Best Practices, Assurance, Future Forward, the Observatory, and CxO Trust, with the AARM and ATF stewardship bodies as the natural homes for the runtime-and-governance pieces.

Conclusion

The Agent Name Service is a serious and timely attempt to give autonomous agents what they conspicuously lack today: a verifiable identity and a discoverable, capability-aware name that works across organizational and protocol boundaries. It deserves to be taken seriously, and CSA — through MAESTRO, AICM, AARM, and the Agentic Trust Framework, and through the contributors who helped create ANS — is unusually well placed to shape it. But the decision to root agent identity in DNS and the certificate-authority system is consequential. It buys global reach and familiarity at the cost of inheriting two centrally governed trust hierarchies and the long catalog of attacks against them, and it leans most heavily on the one DNS security mechanism, DNSSEC, that the world has conspicuously failed to deploy at scale. Those properties do not make ANS unsuitable; they make ANS a layer that must be deployed with eyes open, governed by fail-closed defaults, monitored as a systemic dependency, and complemented by the human-accountability, runtime-enforcement, and decentralized-identity mechanisms it does not itself provide. The research agenda this paper sets out — a fail-closed trust profile, revocation-propagation guidance, AICM control extensions, a neutral comparison of identity fabrics, an adversarial playbook, and empirical measurement — is how CSA can ensure that as agent identity moves into DNS, it does so on a foundation the security community has actually examined.

References

[1] Biometric Update. "Linux Foundation unveils DNS-based identity standard for AI agents." June 2026.

[2] Linux Foundation. "Linux Foundation Announces Intent to Launch Agent Name Service to Establish Trusted Identity Infrastructure for AI Agents." June 23, 2026.

[3] Huang, K., Narajala, V. S., Habler, I., Sheriff, A. "Agent Name Service (ANS): A Universal Directory for Secure AI Agent Discovery and Interoperability." arXiv:2505.10609, May 2025.

[4] Narajala, V. S., et al. "Agent Name Service (ANS) Protocol — Internet-Draft draft-narajala-ans-00." IETF, 2025.

[5] Narajala, V. S., Courtney, et al. "Agent Name Service v2 (domain-anchored) — draft-narajala-courtney-ansv2-01." IETF.

[6] OWASP GenAI Security Project. "Agent Name Service (ANS) for Secure AI Agent Discovery v1.0."

[7] Huang, K. "ANS reference implementation (GitHub)." (MIT license; uses mock cryptography for demonstration.)

[8] PRNewswire. "Linux Foundation Announces Intent to Launch Agent Name Service…." June 23, 2026.

[9] Lardinois, F. "Can DNS become the basis for AI agent identity?" The New Stack, June 23, 2026.

[10] The Register. "Agent Name Service proposal." May 20, 2025.

[11] "Upgrade or Switch: Do We Need a Next-Gen Trusted Architecture for the Internet of AI Agents?" arXiv:2506.12003.

[12] Help Net Security. "DNS-AID: a minimal approach to AI agent discovery over DNS." June 1, 2026.

[13] Internet Systems Consortium. "BIND 9 and CVE-2008-1447 (Kaminsky DNS cache poisoning)."

[14] Man, K., et al. "DNS Cache Poisoning Attack Reloaded (SAD DNS), ACM CCS 2020"; Cloudflare, "SAD DNS Explained."

[15] APNIC Labs. "DNSSEC measurement statistics"; APNIC Blog, "Towards an industry best practice for DNSSEC automation," Feb 25, 2026.

[16] Cisco Talos. "Sea Turtle: DNS hijacking abuses trust in core internet service." 2019.

[17] Internet Society. "Amazon's Route 53 BGP Hijack." April 2018.

[18] Bortzmeyer, S. "RFC 7626: DNS Privacy Considerations." IETF, 2015.

[19] ThousandEyes. "Dyn DNS DDoS Attack analysis." 2016.

[20] zvelo. "What Is an IDN Homograph Attack?"

[21] Princeton University. "Mitigating the Threat of a Malicious CA: Functional PKI (F-PKI), NDSS 2022."

[22] Mozilla Security Blog. "Fraudulent *.google.com Certificate (DigiNotar)." August 2011.

[23] Google Security Blog. "Chrome's Plan to Distrust Symantec Certificates." September 2017.

[24] W3C. "Decentralized Identifiers (DIDs) v1.0." Recommendation, July 2022.

[25] A2A Project. "Agent Card concept."

[26] Model Context Protocol. "Authorization specification."

[27] SPIFFE. "SPIFFE overview."

[28] IETF. "OAuth 2.0 for AI Agents acting On Behalf Of Users – draft-oauth-ai-agents-on-behalf-of-user-02."

[29] Okta. "Okta Announces Cross App Access Partners." June 2026.

[30] NIST NCCoE. "Accelerating the Adoption of Software and AI Agent Identity and Authorization (Initial Public Draft)." February 5, 2026.

[31] CyberArk. "Machine Identities Outnumber Humans by More Than 80 to 1." 2025.

[32] GitGuardian. "The State of Secrets Sprawl 2026."

[33] Huang, K. "Agentic AI Threat Modeling Framework: MAESTRO." Cloud Security Alliance, February 6, 2025.

[34] Cloud Security Alliance. "AI Controls Matrix (AICM)." 2025.

[35] Cloud Security Alliance. "Agentic AI Identity and Access Management: A New Approach." August 2025.

[36] Cloud Security Alliance. "Identity and Access Gaps in the Age of Autonomous AI." March 2026.

[37] AARM. "Autonomous Action Runtime Management specification." CSAI Foundation.

[38] Agentic Trust Framework. "Agentic Trust Framework." CSAI Foundation (CC BY 4.0; founding author Josh Woodruff / MassiveScale.AI).

[39] Cloud Security Alliance. "CSAI Foundation Announces Key Milestones to Secure the Agentic Control Plane." April 29, 2026.