

# Why Naming Matters: Identifier Design for Decentralized Digital Infrastructure in the Age of AGI

---

LIXIA ZHANG, TIANYUAN YU,  
XINJIAO LI, DIRK KUTSCHER

IETF 125, MARCH 16, 2026

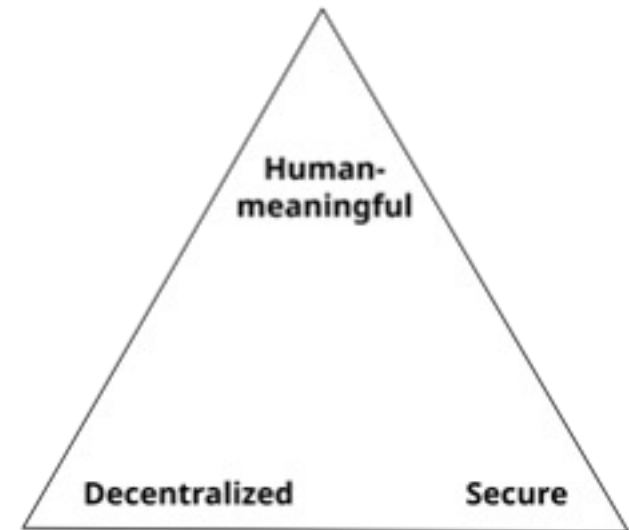
### What does decentralization mean?

- **Decentralization:** decentralizing the control power
  - No single entity/organization can unilaterally dictate decisions
- **Decentralizing the control power** → *decentralizing trust management*
  - Multiple autonomous entities, each chooses their own trust anchor
- **Decentralizing trust** *requires* establishing trust relations between semantically named entities
- This in turn requires **globally unique, semantically meaningful names**

## ZOOKO'S TRIANGLE

### Can we have globally unique, semantically meaningful, and decentralized names?

- **Zooko's triangle** identified 3 desirable properties for naming network entities
  - **Human-meaningful:** Meaningful and memorable names are provided to the users.
  - **Secure:** Cryptographic verifiability of name ownership
  - **Decentralized:** anyone can create and use names *without* needing permission from a central registry or authority
- **Claim:** human-meaningful, decentralized, and secure are mutually exclusive: pick two.



## REBUTTLE ZOOKO'S TRIANGLE

### Zooko's triangle contains two definitional errors

1. Decentralization = absence of coordination
2. Security implies crypto keys as names

### Zooko's triangle contains two definitional errors

#### 1. Decentralization = absence of coordination

- Defines decentralization by the *absence of naming coordination*
  - not by the *distribution* of control authority
- Under this definition, the only means of decentralizing naming is self-sovereign issuance
  - Immediately kills semantic meaningfulness

#### 2. Security implies crypto keys as names

### Zooko's triangle contains two definitional errors

- 1. Decentralization = absence of coordination**
- 2. Security implies crypto keys as names** (Cryptographic verifiability of name ownership)
  - If a name is required to guarantee binding integrity and unforgeability *on its own*, the only way to achieve that is to make the name *be* the key — or a hash of it.
  - A crypto string carries no human-interpretable context — sacrifice semantic meaningfulness by construction

## REBUTTLE ZOOKO TRIANGLE

### Zooko's trilemma comes from these two (erroneous) definitions

1. Decentralization = absence of coordination
2. Security implies crypto keys as names

Both push toward (crypto-based) self-sovereign identifiers, making that solution space appear as the *only* way to achieve decentralization and security simultaneously.

The trilemma is essentially a rationalization of the design choice; nevertheless, it has been *taken as a fundamental impossibility result* by multiple efforts

## DECENTRALIZED IDENTIFIER (DID)

### DID's premise: decentralization starts by self-sovereign identifiers

- **DID specifies universally**
  - A name format: `did:method:method-specific-identifier`
  - A resolution interface: DID documents containing keys and endpoints
- DID itself is essentially a namespace wrapper
  - The interesting analysis is at the method level

### Key method families and their tradeoffs

Method	Example	Name issuance	Resolution	Name type
Ledger-based	did:ethr, did:ion	Permissionless, on-chain	Blockchain lookup	Crypto key/hash
Web-based	did:web	DNS-delegated	HTTPS fetch	DNS name + path
Peer/ephemeral	did:peer, did:key	Self-generated	Local/out-of-band	Crypto key

## CENTRALIZATION IN PRACTICE

### Centralization seems to be creeping back in for each

#### **did:web**

*Resolves via DNS + HTTPS*

Fully dependent on DNS and domain ownership. If DNS viewed “centralized”, did:web is centralized by inheritance. This approach implicitly concedes the argument.

#### **did:ethr / blockchain methods**

*Data on-chain, clients query via centralized service*

In practice, most clients hit a centralized provider (e.g., Infura). Decentralized data storage, centralized access layer

#### **Universal Resolver (provided by DIF)**

*Decentralized by design, centralized in deployment*

DIF (Decentralized Identity Foundation) hosts a reference instance (dev.uniresolver.io) that most people use, making it a de facto centralized resolution endpoint

#### **In addition: per-method silos**

*Each method manages its own resolution independently*

No cross-method discoverability — a fragmented collection of isolated islands. Interoperability fails in global scope.

### DID's premise: decentralization starts by self-sovereign identifiers

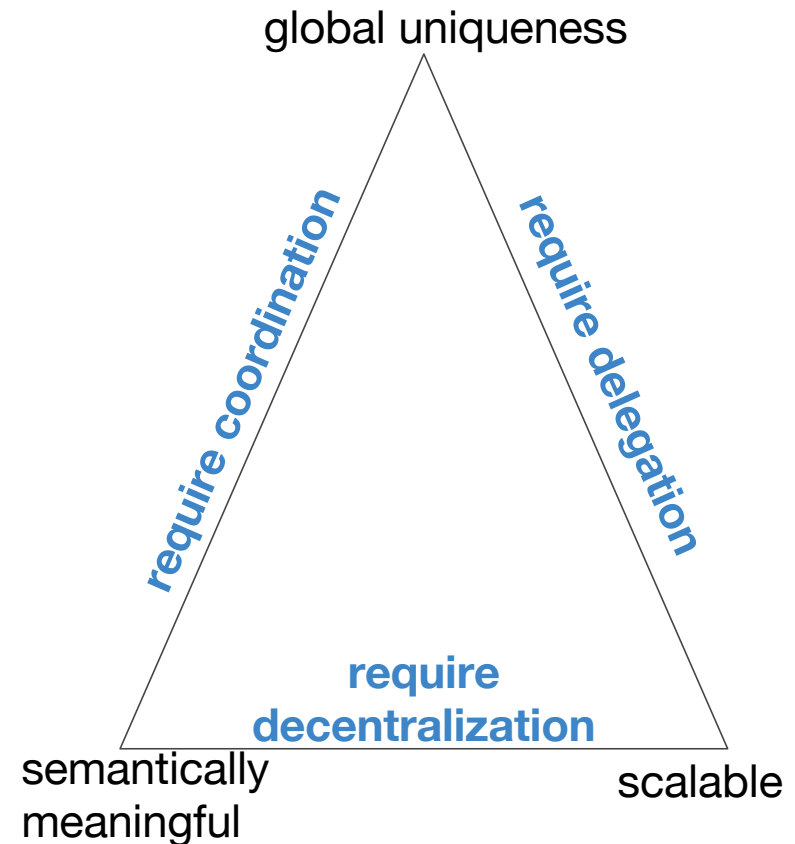
- **"DNS is centrally controlled"**
  - Namespace coordination is not the same as central control.
  - The delegation model distributes authority — it does not concentrate it.
    - Each zone owner controls their own sub-namespaces.
- **"Decentralize the identifier first"**
  - Decentralization is about *control power*, which flows from *trust*
  - Decentralizing the identifier alone does not solve the problem of where control concentrates.
- **"Self-sovereign identifiers enable trust"**
  - A self-issued identifier + self-signed key establishes no *trust relation*
  - DID provides the identifier but leaves out the trust infrastructure — the hard problem —unaddressed.

### Naming Requirements Triangle

Three necessary properties for the Internet namespace:

- **Global uniqueness** — no two entities can hold the same name; a delegation model structurally prevents collisions
- **Semantic meaningfulness** — names carry human-comprehensible context, enabling trust/security reasoning
- **Scalability** — no bottleneck, no single point of failure, no upper bound on namespace;
  - (DNS syntax has limit, which should be resolvable)

These are requirement for the **namespace design** — resolution availability and integrity are properties of the name resolution *system*, not of the *namespace* itself.



## DNS namespace: satisfying all the three requirements

- **Coordinated:** ICANN delegation model ensures global uniqueness. Each zone owner controls their sub-namespace — authority is distributed by design.
- **Structured:** 'mail.example.com' self-describes its resolution path: root → .com → example.com. Any resolver finds any name without a central directory.
- **Semantically Meaningful:** Domain names carry human-interpretable organizational and jurisdictional context. 'mit.edu' is a meaningful name to enable *trust reasoning*. Cryptographic operation assures authenticity.

In addition, already globally deployed for decades

### Identity = Name + Key

- Name alone: can't prove ownership
- Key alone: semantically empty
- **Name + Key — Meaningful Trust**
  - "ucla.edu vouches for alice.ucla.edu"
  - Immediately interpretable: does UCLA have authority here?
  - Trust chains are human-navigable, not just machine-verifiable.
  - **Keys certified by localized trust anchors** — *global namespace, localized trust.*

## Functions that DNS provides

- Defining a **globally unique namespace**
- Operating a resolution service
- DNSSEC: use a global trust root to prove name ownership
- This talk: solely focus on DNS offering *unique, semantically meaningful names*

Many people view DNS as a centralized system, both for its namespace management and for its lookup service.

### The reality

- Namespace allocation is decentralized through delegation; only the TLD allocations are coordinated by ICANN as a necessity
- Name resolution service consolidation is observed — an operational phenomenon, independent of namespace design

### Today's naming and security practice shows three issues

- ***Only organizations and websites have DNS names in general***
  - Organizations may delegate names to devices (e.g. routers)
  - Users have no names in general, with an exceptionally small number of outliers
    - mostly used for personal websites, some in email but not in apps in general
- ***Security solutions are not built on a unified namespace***
  - Security solutions fragmented across multiple identifier spaces:
    - X.509 uses Distinguished Names, OAuth uses URIs and client IDs, SAML uses entity IDs, SSH uses raw public keys, email uses addresses.
    - DNS names appear incidentally but are never the authoritative anchor.
  - Each system defines its own identity authority, no principled way to reason trust relations across boundaries.
- ***Implicit coupling of Identity provider and trust authority (authorization)***
  - User identities default to globally unique email addresses controlled by dominant email provider, which plays the role of OAuth server (centralized by default)

### What agentic AI brings to networking

- Autonomous entities that **perceive, decide, act, and interact**
- **Autonomous actions:** fundamentally higher demands on security
- **Multi-agent collaboration**, not only as software components but also as embodied systems (e.g. intelligent robots) working alongside humans.
- **Orders of magnitude difference in scale** (billions-trillions), and **dynamics** (both long and short-lived agents)
- Current state of affairs:
  - Agent capability: rapid progress
  - Agent system architecture: remains fragile, with brittleness caused by the absence of an architectural foundation for **naming/identity, security, and accountability.**

## THE CENTRAL QUESTION

### Patch or Rebuild the Foundation?

- As AI agents become dominant actors in digital infrastructure, we face a fork in the road:

#### Option A — Patch & Adapt

- Extend existing identity and security protocols (OAuth, SAML, X.509) for agents
- Layer agent-specific mechanisms on top of today's fragmented namespace infrastructure
- Pro: Faster to deploy — reuses existing tooling, standards, and operational expertise
- Con: Inherits structural flaws — namespace fragmentation and trust anchor inconsistency — and amplifies them at agent scale

#### Option B — Start with Right Foundation

- Build agent identity on DNS as the unified, universal namespace: every agent gets a DNS name as its primary identifier
- Localize trust anchors: global namespace, localized trust
- Pro: The path toward genuinely decentralized, scalable agent infrastructure
- Con: Harder to bootstrap — requires namespace discipline and new tooling

## THE RIGHT FOUNDATION

### What Identity Infrastructure should look like in the age of AGI

- **DNS as the unified namespace for all networked entities**
  - All entities — humans, organizations, agents, services — have identities rooted in DNS
  - A single, globally consistent namespace enables unambiguous identity reasoning
- **Identity = name + key**
  - Each entity has a DNS name bound to a cryptographic key. The name carries semantic context; the key provides cryptographic verifiability.
- **Localized trust anchors: global namespace, local trust — independent layers**
  - Trust is built from locally meaningful relationships
  - Local trust anchors can interconnect through cross-certification
    - Scale via *trusted intermediaries*
  - No single authority can unilaterally revoke trust across the ecosystem
- **Delegation chains as native infrastructure**
  - Deep, dynamic human-to-agent and agent-to-agent delegation — with scope, constraints, and provenance — at agent scale

# Necessary Steps to Move Towards this New Direction

- **Decoupling namespace delegation from trust management**
  - Namespace hierarchy and trust relation structures are independent
- **Developing namespace management support to enable**
  - Every organization, user, and agent manages their own namespace slice
  - Every entity can run their own DNS services
- **Developing decentralized trust management to enable**
  - Every administrative entity manages their own trust anchors
  - Mutual authentication among peers
  - Fine-grained security policies for who can do what, under what conditions, with least privilege

## What the IRTF/DINRG Community Should Do

The AGI transition: a rare opportunity to establish the right foundation

- Resist the temptation to patch existing identity protocols for agents — patching inherits structural flaws and amplifies them at agent scale
- Treat **DNS as the authoritative identity namespace** for all including agent — not one option among many, but the foundation
- Adopt localized trust anchors built on DNS names as the model to decentralize trust, hence decentralize control power
- Design delegation chains as baseline infrastructure — **deep, dynamic, and verifiable by default**

## SUMMARY

# Global Namespace, Local Trust

—*Namespace Design for Decentralized Digital Infrastructure in the Age of AGI*

- **Decentralization is about distributing control power**
- Distributed control requires a **structured, semantically meaningful namespace**
  - DNS provides exactly that, and is globally deployed
- **Identity = DNS name + cryptographic key**
  - The name provides semantic context; the key provides cryptographic verifiability
  - Trust chains must be human-navigable; machine-verifiability alone is inadequate
- **Global namespace, local trust** — independent, complementary layers.
  - For AI agents, this becomes a scaling requirement, not an optional architecture.
- **Path forward:** DNS as the unified namespace with local trust anchors and principled delegation chains