A Global Switchboard for the Agentic Web: Connecting Discovery Islands Across Enterprises, SMBs, and Individuals

Ramesh Raskar (MIT Media Lab), Pradyumna Chari (MIT Media Lab), Luca Muscariello (Cisco), Samuel Sharaf (Google), Karan Bharadwaj (Project NANDA), Vijoy Pandey (Cisco)

V0.1 Draft Document. Request for Comments. Preprint.

Abstract

Agent discovery is diverging into multiple useful approaches — a natural consequence of innovation across different communities, use cases, and deployment environments. AI Catalog is a useful starting point for public enterprise agent and AI-resource discovery. It provides a typed, nestable, machine-readable container for heterogeneous AI artifacts: A2A Agent Cards, MCP server descriptors, nested catalogs, tools, skills, datasets, gateways, and other AI resources.

However, AI Catalog needs to be strengthened by allowing other discovery mechanisms to connect through it. These include DNS-AID, ANS, .well-known, A2A cards, MCP, gateways, and platform registries. In addition, telecom directories, EdgeAI registries, IoT systems, and country- or sector-specific discovery systems will each continue to solve part of the problem. These non-compatible or emerging approaches are valuable — the product of real innovation — but they can also create discovery islands when left unconnected.

This paper argues for a NandaIndex Federated Resolution Architecture: a global switchboard that uses the AI Catalog format itself as the common index-record format. NandaIndex strengthens AI Catalog by making it the preferred framework through which other discovery systems can be bridged. The switchboard proposed by NANDA and AGNTCY, built jointly by Outshift (Cisco) and MIT Media Lab, provides the starting point for this architecture. This paper is a next step that considers improvements to AI Catalog and extends the switchboard model to cover enterprises, SMBs, and individuals.

The focus is on three practical needs:

- Enterprise heterogeneity, where some companies use AI Catalog and others use DNS-AID, gateways, .well-known, or legacy registries.
- SMBs such as MoonBakery, where agents may be hosted on AWS while agent cards are hosted by a third party such as list39.org.
- Individuals such as john@hotmail.com, who do not own a domain name, where a personal agent may be hosted on AWS, Azure, GCP, or another provider, while the agent card lives elsewhere.

This paper presents an architecture proposal and initial design. It identifies the missing bootstrap layer between identity and discovery, defines the role of NandaIndex within that layer, and outlines the identity, trust, and federation mechanisms that must be formalized.

1. Introduction

The agentic web is emerging as a new layer of the internet, where autonomous and semi-autonomous software agents discover, compose, and interact with capabilities across organizational and technological boundaries. Enterprises are beginning to expose agents, tools, skills, and data services, while platforms and ecosystems are rapidly developing new ways to publish and consume these capabilities. At the same time, small businesses and individuals are starting to participate, expanding the potential scope of the

ecosystem far beyond traditional enterprise systems.

Innovation drives divergence. Different communities solving real problems — enterprises managing compliance, developers building open tools, governments standing up sovereign registries, telecom operators integrating agents into their networks — will naturally arrive at different discovery systems. This is not a coordination failure. It is what healthy ecosystems look like.

The answer is not to consolidate into one system. The answer is to build bridges.

Multiple discovery mechanisms are emerging in parallel:

- AI Catalogs for structured capability metadata
- domain-based mechanisms such as DNS-AID
- enterprise gateways enforcing access control and policy
- agent representation formats such as A2A Agent Cards and MCP descriptors
- platform registries, telecom directories, and vertical-specific systems

Each approach addresses real requirements — identity, governance, performance, usability, or domain-specific constraints. But collectively, they create disconnected discovery islands, where capabilities are only visible within a given system and difficult to discover, trust, or use across boundaries.

AI Catalog is a strong starting point. When an enterprise publishes an AI Catalog at a well-known endpoint and controls the domain, direct resolution already works — NandaIndex is not in the critical path. No extra infrastructure is needed. Any requester can fetch the catalog directly.

But AI Catalog does not cover every actor or every discovery system. Innovation will continue to push new registries, new formats, and new contexts into the picture:

- Sovereign national directories will be required by some governments
- Telecom registries will integrate agents into carrier infrastructure
- IoT and EdgeAI systems will develop their own lightweight discovery mechanisms
- Vertical registries in health, finance, aviation, and education will emerge with sector-specific governance

Meanwhile, SMBs and individuals need paths to discoverability that require no enterprise infrastructure at all. Population-scale citizen-agent initiatives, in which every individual is issued a personal agent that must remain discoverable without owning a domain, make this need concrete [**?**, **?**].

The central challenge is not the absence of discovery mechanisms. It is the absence of a unifying layer that makes heterogeneous discovery systems interoperable.

The Agentic Resource Discovery (ARD) specification and the AGNTCY Agent Directory provide a critical step forward. ARD standardizes how agents query and consume discovery information, while AGNTCY demonstrates how decentralized, federated directory infrastructure can support discovery at scale. AI Catalog provides a shared, extensible format for representing agentic resources across these systems.

Yet even with these advances, one key problem remains unresolved: how to map a stable identity or known resource identifier to the correct discovery mechanism, not just the final resource.

NandaIndex is proposed as a global switchboard: an AI Catalog-formatted federation layer that bridges these discovery islands by mapping stable identities to the correct discovery object — whether that is an AI Catalog, a DNS-AID record, a gateway, a platform registry, or a hosted agent card.

The global switchboard proposed by NANDA and AGNTCY, developed jointly by Outshift (Cisco) and MIT Media Lab, provides the early working implementation of this vision. This paper builds on that

foundation by incorporating AI Catalog improvements and extending the model to the full range of enterprise, SMB, and individual use cases.

2. Discovery Diversity: From Divergence to Bridges

2.1 Why Divergence Is Healthy

When AI Catalog emerged from collaboration between the A2A and MCP communities, it showed what good interoperability looks like: rather than creating yet another isolated registry, two communities built a shared index format that preserves what each community had already built. The result is a typed, nestable container that delegates the specification of each artifact type to the relevant protocol owner. The interoperability surface is minimal; the delegation is maximal.

That same principle applies at a larger scale. AGNTCY, the open-source collective building the Internet of Agents — spearheaded by Cisco/Outshift, LangChain, and Galileo — takes a similar approach: define common specifications (OASF, ACP) while letting each component deliver standalone value and integrate through common APIs. The AGNTCY Agent Directory Service (ADS) uses DHT-based routing, OCI-aligned content addressing, and Sigstore-backed integrity — a purpose-built system optimized for decentralized agent discovery that complements, rather than replaces, catalog-based approaches.

The rapid evolution of the agentic web is producing a diverse set of discovery mechanisms, each designed to address specific technical, operational, or organizational requirements. This diversity is not incidental — it reflects the fundamentally heterogeneous environments in which agentic systems operate:

- AI Catalog provides a structured, extensible format for describing heterogeneous AI resources, including agents, tools, skills, MCP servers, and nested catalogs. It is particularly well suited for enterprises that want to expose governed, machine-readable inventories of their capabilities.
- Domain-based mechanisms such as DNS-AID enable efficient, cacheable discovery rooted in domain ownership. These approaches are well aligned with existing internet infrastructure and are effective when the requester already knows the domain of interest.
- Enterprise gateways act as controlled entry points that enforce authentication, authorization, rate limiting, and policy. Rather than exposing full inventories, organizations may prefer dynamic, policy-aware disclosure of capabilities depending on the requester.
- Identity systems, including ANS, digital wallets, and decentralized identifiers (DIDs), provide mechanisms for verifying ownership, provenance, and trust.
- Platform and vertical registries — operated by cloud providers, agent platforms, telecom operators, or industry consortia — support onboarding, lifecycle management, billing, observability, and compliance within specific ecosystems.
- Agent-specific descriptors, such as A2A Agent Cards and MCP server descriptors, define how to interact with individual agents and tools.

Innovation continues to diverge in exactly this way. We should not fight it. We should support it with bridges and switchboards.

2.2 Where Each System Fits — and Where Bridges Are Needed

Discovery mechanism	Best suited for	Where a bridge adds value
AI Catalog	Rich enterprise resource indexing; typed, nestable, multi-artifact	Entities without domain names; cross-system linking

Discovery mechanism	Best suited for	Where a bridge adds value
DNS-AID	Domain-controlled, fast, cacheable enterprise resolution	SMBs, individuals, non-DNS systems
AGNTCY ADS	Decentralized DHT routing; OCI storage; verifiable integrity	Linking into catalog-based or index-based lookups
Gateways	Access control, privacy, throttling, compliance	Discovery of agents not behind that gateway
ANS / DIDs	Identity and ownership proof	Resource indexing, routing
ARD	Standardized capability search and finder-to-finder referrals	Standardized capability search and finder-to-finder referrals
Platform Registries	Onboarding, hosting, billing, developer experience	Cross-platform and cross-community discovery
Telecom / IoT / Sovereign	Sector-specific governance, carrier integration, national compliance	Bridging into global index
NandaIndex	Identity-to-discovery-object resolution for all entity types	Not applicable — this is the switchboard layer itself

2.3 Domain Owners vs The Rest

For enterprises that publish AI Catalog at a well-known endpoint and own the domain, direct resolution already works. NandaIndex is not in the critical path. This is by design. NandaIndex adds value at the edges: for actors who cannot publish a well-known catalog, for discovery systems that are not AI Catalog-compatible, and for bridging across the growing number of registries that innovation will continue to produce.

The consequences of leaving these systems disconnected are significant:

- limited interoperability, as agents cannot easily discover or use capabilities outside their native ecosystem
- duplication of effort, as similar capabilities are reimplemented across systems
- fragile integrations, relying on hardcoded endpoints or proprietary connectors
- constrained network effects, preventing the emergence of large-scale, composable multi-agent systems

The challenge, therefore, is not to replace these mechanisms with a unified standard. It is to enable them to coexist and interoperate within a common discovery framework.

3. A Layered Model of the Agentic Web

The fragmentation described in the previous section can be better understood — and systematically addressed — by viewing the agentic ecosystem as a set of separable but interdependent layers.

A consistent layered model clarifies roles, responsibilities, and interfaces across systems, and avoids the common design mistake of conflating formats, APIs, and infrastructure into a single monolithic solution. Instead, the agentic web emerges from the interaction of distinct layers, each solving a different part of the problem.

At a high level, four layers can be identified:

3.1 Representation Layer: Describing Agentic Capabilities

The representation layer defines how agentic resources are described in a machine-readable way.

Formats such as AI Catalog, along with artifact-specific descriptors like A2A Agent Cards and MCP server descriptors, provide structured metadata that answers fundamental questions:

- What is this resource?
- What can it do?
- How is it invoked?
- Who published it?

This layer is intentionally protocol-agnostic. It provides a common, typed representation of heterogeneous resources, enabling interoperability at the data level across different systems and execution environments.

However, representation alone is static. It describes capabilities, but does not define how they are discovered or accessed dynamically.

3.2 Discovery Layer: Querying and Finding Capabilities

The discovery layer defines how agents search for relevant capabilities at runtime.

The Agentic Resource Discovery (ARD) specification provides a standard interface for querying and retrieving agentic resources across catalogs and registries. It enables:

- intent-based search rather than hardcoded lookup
- dynamic discovery across multiple sources
- consistent interpretation of results across systems

This layer transforms discovery from a static process — where clients load predefined catalogs — into a dynamic, query-driven interaction.

However, discovery APIs alone do not solve where data resides, how results are aggregated across systems, or how trust is enforced. These concerns are delegated to the infrastructure layer.

3.3 Infrastructure Layer: Federation, Routing, and Trust

The infrastructure layer provides the operational foundation that makes discovery usable in practice.

Systems such as the AGNTCY Agent Directory Service (ADS) implement this layer by managing:

- storage and distribution of records
- routing and federation across nodes (e.g., via distributed hash tables)
- identity, signing, and verification of published artifacts
- policy enforcement and governance controls

This layer enables multiple independent discovery systems — operated by different organizations — to form a federated network rather than isolated registries.

It ensures that:

- discovery queries can span organizational boundaries
- results can be verified for authenticity and provenance

- data can be distributed efficiently without requiring a single discovery protocol or a single authoritative resource registry

Together, the representation, discovery, and infrastructure layers form a coherent stack for publishing, finding, and verifying agentic capabilities.

3.4 Resolution Layer: Mapping Identities to Discovery Systems

The three layers above handle how resources are described, found, and distributed. However, a fourth concern remains distinct: given a stable identity or known resource identifier, which discovery mechanism should a requester use in the first place?

The resolution layer answers this question by mapping identities to the appropriate discovery entry point — whether that is an AI Catalog, a DNS-AID record, a gateway, or another system. This layer is described in full in Section 5, where the Federated Resolution Architecture and NandaIndex are introduced as its concrete realization.

4. Permissionless Deployment: Runtime, Card, and Domain Can Be Separate

One of NandaIndex's most important properties is that it supports permissionless deployment — meaning any entity can register a stable identity or known resource identifier and point to its discovery object without needing to control a single unified infrastructure stack.

Specifically, NandaIndex makes it possible for:

- The agent runtime to live on one provider (AWS, Azure, GCP, Railway, Cloudflare)
- The agent card to be hosted by a separate third-party card host (e.g., list39.org)
- The website or brand domain to sit on yet another platform

No single party needs to control all three. No enterprise gateway or DNS certificate is required. A bakery, a freelancer, a civic group, or an individual user can register a stable identity and delegate card hosting and runtime to whatever providers they choose.

This model naturally extends to emerging systems. As sovereign directories, telecom registries, and IoT discovery systems innovate and mature, they can each publish a NandaIndex entry that points into their native system — without needing to conform to AI Catalog internally. The switchboard does not prescribe how any target system works. It only needs to know how to reach it.

5. Federated Resolution Architecture

The layered model described previously explains how agentic resources are represented, discovered, and managed. However, a fundamental gap remains:

> **Given a stable identity or known resource identifier, how does a system determine the correct discovery mechanism?**

In practice, multiple mechanisms coexist — AI Catalogs, DNS-based discovery (e.g., DNS-AID), gateways, platform registries, and direct agent descriptors. Each is authoritative within its domain, but there is no universal way to select among them.

This section introduces a Federated Resolution Architecture that provides this missing layer by mapping identities to their appropriate discovery paths.

5.1 Conceptual Model

Federated resolution separates two concerns:

- Resolution: determining where and how discovery should begin
- Discovery: finding capabilities via mechanisms such as ARD

This yields a simple abstraction:

Identity → Resolution → Discovery → Invocation

The four stages serve different functions:

- Identity: a stable identifier, such as a domain-anchored identifier, platform identity, DID, or provider-verified account identity.
- Resolution: selection of an authoritative discovery entry point.
- Discovery: retrieval or search of capabilities through AI Catalog, ARD, DNS-AID, a gateway, or another native mechanism.
- Invocation: interaction with the selected resource through A2A, MCP, REST, or another supported protocol.

Resolution is therefore a bootstrap operation. This separation enables interoperability without enforcing a single discovery system.

5.2 AI Catalog as a Universal Resolution Format

The architecture reuses AI Catalog as a unified record format for resolution.

Each resolution entry is an AI Catalog–compatible record that acts as a typed pointer, indicating:

- the discovery mechanism (e.g., DNS-AID, catalog, gateway)
- the location or access method
- metadata required for interpretation and trust

This allows a consistent representation for both:

- direct resources (agent cards, MCP descriptors)
- indirect discovery paths (catalogs, DNS, registries)

No new schema is introduced — AI Catalog becomes the bridge across heterogeneous systems.

5.3 Resolution Targets

A resolution entry may point to:

- AI Catalog endpoints
- DNS-based discovery records (DNS-AID)
- ARD-compatible directories
- enterprise gateways
- A2A Agent Cards or MCP descriptors
- platform or vertical registries
- SMB or hosted agent cards

- personal agent endpoints
- telecom, IoT, EdgeAI, or sovereign national registries

The resolution layer does not replace these systems — it connects them.

5.4 Resolution Flow

A typical flow is:

1. A requester starts with an identity (domain, email, or URN)
2. The requester queries a resolution system
3. The system returns AI Catalog–formatted resolution entries
4. Each entry specifies a discovery path
5. The requester follows that path (ARD, DNS lookup, gateway, etc.)
6. Standard discovery, verification, and invocation proceed

This enables dynamic, protocol-agnostic routing across discovery systems.

5.5 Federation and Infrastructure Integration

Federated resolution builds on existing components:

- AI Catalog → record format
- ARD → discovery interface
- AGNTCY Agent Directory (ADS) → federation, routing, identity, trust

In this stack:

- resolution determines the path
- discovery retrieves capabilities
- infrastructure ensures distribution and verification

This results in a loosely coupled, interoperable architecture.

5.6 NandaIndex: A Federated Resolution Instance

A concrete instantiation of this architecture is NandaIndex, which functions as a global switchboard for agent discovery.

Core design choice: NandaIndex is an AI Catalog-formatted database of index records. It does not introduce a new record format. Each entry is an AI Catalog Catalog Entry pointing to the correct next discovery object for a given identity.

NandaIndex concept	**Implementation**
Entry	AI Catalog Catalog Entry
Identity	URN / URI identifier
Target	URL or inline routing data
Routing hints	Metadata fields

NandaIndex is a federated index of AI Catalog–formatted resolution records:

- each entry maps an identity → to a discovery entry point
- entries can reference: catalogs, DNS-based systems, gateways, registries, or agent cards

For enterprises already on AI Catalog with a well-known endpoint: NandaIndex is optional. Direct resolution already works. A NandaIndex entry remains useful for fallback, federation visibility, migration, and anti-squatting — but it is not required in the critical path.

Rather than acting as a centralized registry, NandaIndex is designed to serve as a distributed index of pointers, where:

- authoritative discovery remains with each system
- multiple resolution paths can coexist
- federation enables cross-boundary discovery

Resolution semantics

```
NandaIndex(identity)
  → AI Catalog entry
  → discovery mechanism
  → target resource
```

This allows clients to dynamically:

- determine which protocol to use
- route across heterogeneous systems
- avoid hardcoded discovery assumptions

How Innovation Flows Through the Switchboard

As new discovery systems emerge — telecom registries, IoT catalogs, national directories, sector-specific systems — they can become reachable through participating resolvers by publishing a NandaIndex record pointing to their native discovery mechanism. They do not need to adopt AI Catalog internally. They do not need to modify their existing systems. The switchboard allows additional mechanisms to be incorporated without requiring conformance.

Role in the Ecosystem

NandaIndex is optional when discovery endpoints are known, but becomes important when:

- discovery mechanisms are unknown
- multiple systems coexist
- identities span organizations, SMBs, or individuals

It provides a consistent entry point into a federated discovery ecosystem, without requiring centralization or protocol lock-in.

Relationship to NANDA Index and AgentFacts

This architecture extends the earlier NANDA Index and AgentFacts model [17]. NANDA identifiers, ownership proofs, and update mechanisms provide the identity and control substrate. AI Catalog provides an interoperable representation through which NANDA resolution records can be exposed to external clients. A NandaIndex resolution record may therefore point to an AgentFacts document, an AI Catalog,

an ARD finder, a DNS-AID name, a gateway, a platform registry, or a terminal agent descriptor. The NANDA Resolution Profile defines how these alternatives are represented and selected.

5.7 Summary

Federated resolution fills the gap between identity and discovery by:

- mapping identities to discovery mechanisms
- using AI Catalog as a universal format
- integrating with ARD and federated infrastructure

NandaIndex demonstrates how this model can be realized as a distributed, global switchboard, enabling agents to discover and interact across heterogeneous systems.

6. Deployment Contexts and Use Cases

The federated resolution architecture applies across heterogeneous environments with different infrastructure, identity, and governance models.

Rather than enforcing uniformity, it enables systems to resolve identities to the appropriate discovery mechanism, allowing interoperability across enterprises, SMBs, and individuals.

6.1 Enterprise — Pure AI Catalog (example.com)

This is the simple case. Direct resolution works. NandaIndex is not required.

An enterprise publishes at https://example.com/.well-known/ai-catalog.json. Any requester can fetch it directly. A NandaIndex entry is optional — useful for federation, fallback, and anti-squatting. Note that urn:ai: identifiers below are used illustratively to demonstrate the resolution model. Their final namespace syntax and registration policy remain to be standardized.

```
{
  "identifier": "urn:ai:domain:example.com",
  "displayName": "Example.com Enterprise AI Catalog",
  "type": "application/ai-catalog+json",
  "url": "https://example.com/.well-known/ai-catalog.json",
  "description": "Public enterprise AI Catalog for agents, tools, skills, gateways, and
  AI resources exposed by example.com.",
  "tags": [
    "enterprise",
    "ai-catalog",
    "public-agent-discovery"
  ],
  "publisher": {
    "identifier": "urn:ai:domain:example.com",
    "displayName": "Example.com",
    "identityType": "dns"
  },
  "metadata": {
    "org.projectnanda.resolutionRole": "nested-ai-catalog",
    "org.projectnanda.preferredDiscovery": "ai-catalog",
    "org.projectnanda.nandaIndexRole": "optional-fallback-entry"
  }
}
```

Resolution: Requester fetches AI Catalog directly → selects agent, tool, MCP server, or gateway → follows artifact URL.

6.2 Enterprise — DNS-AID (skyblue.com)

SkyBlue uses DNS-AID at refunds._agents.skyblue.com. NandaIndex does not replace DNS-AID — it makes the DNS-AID path reachable from the global switchboard.

```
{
  "identifier": "urn:ai:domain:skyblue.com:agent:refunds",
  "displayName": "SkyBlue Refunds Agent DNS-AID Pointer",
  "type": "application/vnd.dns-aid+json",
  "data": {
    "method": "dns-aid",
    "domain": "skyblue.com",
    "organizationDiscoveryName": "_agents.skyblue.com",
    "agentDiscoveryName": "refunds._agents.skyblue.com",
    "serviceHint": "refunds",
    "expectedResult": "DNS-AID returns a gateway, catalog, or agent-card pointer
    controlled by skyblue.com"
  },
  "description": "Federated AI Catalog entry pointing to SkyBlue's DNS-AID-based
  discovery path for its refunds agent.",
  "tags": [
    "enterprise",
    "dns-aid",
    "airline",
    "refunds",
    "gateway"
  ],
  "publisher": {
    "identifier": "urn:ai:domain:skyblue.com",
    "displayName": "SkyBlue Airlines",
    "identityType": "dns"
  },
  "metadata": {
    "org.projectnanda.resolutionRole": "dns-aid-pointer",
    "org.projectnanda.preferredDiscovery": "dns-aid",
    "org.projectnanda.authoritativeSystem": "skyblue.com DNS",
    "org.projectnanda.nandaIndexRole": "federated-pointer"
  }
}
```

Resolution: NandaIndex → DNS-AID lookup → SkyBlue gateway or Agent Card → auth → agent.

6.3 SMB — Moon Bakery

Moon Bakery owns moonbakery.com but runs no enterprise infrastructure. Its runtime, agent card, and domain are with three separate providers — a practical example of permissionless deployment.

```
identity:    orders@moonbakery.com
agent card:  https://agentcards.list39.org/moonbakery.com/orders.json  (third-party host)
runtime:     AWS  (separate provider)
```


```
{
  "identifier": "urn:ai:domain:moonbakery.com:agent:orders",
  "displayName": "Moon Bakery Orders Agent",
  "type": "application/a2a-agent-card+json",
  "url": "https://agentcards.list39.org/moonbakery.com/orders.json",
  "description": "Ordering agent for Moon Bakery. Supports menu lookup, order placement,
  pickup scheduling, and order status.",
  "tags": [
    "smb",
    "bakery",
    "orders",
    "commerce",
    "a2a-agent-card"
  ],
  "publisher": {
    "identifier": "urn:ai:domain:moonbakery.com",
    "displayName": "Moon Bakery",
    "identityType": "dns"
  },
  "metadata": {
    "org.projectnanda.resolutionRole": "smb-agent-card",
    "org.projectnanda.preferredDiscovery": "nandaindex",
    "org.projectnanda.agentCardHost": "list39.org",
    "org.projectnanda.runtime.provider": "AWS",
    "org.projectnanda.runtime.url": "https://moonbakery-orders.aws.example.com",
    "org.projectnanda.auth.metadata": "public",
    "org.projectnanda.auth.execution": "payment_or_session_token_required"
  }
}
```


Resolution: NandaIndex → Agent Card at list39.org → AWS runtime → payment or session token required.

Moon Bakery needs only a stable identity and a delegated path. No dedicated agent-discovery DNS records, enterprise gateway, or organization-operated catalog endpoint are required. NandaIndex becomes the primary discovery entry point for SMBs.

6.4 Individual — john@hotmail.com

John has no domain. His runtime is on Azure; his agent card is with a third-party host. No personally controlled domain is required.

```
identity:    john@hotmail.com
```

```
agent card:  https://agentcards.list39.org/personal/john@hotmail.com/card.json
runtime:     Azure
```

```
{
  "identifier": "urn:ai:email:john@hotmail.com",
  "displayName": "John's Personal Agent",
  "type": "application/a2a-agent-card+json",
  "url": "https://agentcards.list39.org/personal/john%40hotmail.com/card.json",
  "description": "Personal agent associated with john@hotmail.com. Public metadata is
  minimal; private actions require user consent.",
  "tags": [
    "personal-agent",
    "individual",
    "email-identity",
    "a2a-agent-card"
  ],
  "publisher": {
    "identifier": "urn:ai:email:john@hotmail.com",
    "displayName": "John",
    "identityType": "email"
  },
  "metadata": {
    "org.projectnanda.resolutionRole": "personal-agent-card",
    "org.projectnanda.preferredDiscovery": "nandaindex",
    "org.projectnanda.agentCardHost": "list39.org",
    "org.projectnanda.runtime.provider": "Azure",
    "org.projectnanda.runtime.url": "https://john-agent.azure.com",
    "org.projectnanda.auth.metadata": "public_minimal",
    "org.projectnanda.auth.execution": "user_consent_required"
  }
}
```

Resolution: NandaIndex → Agent Card at list39.org → Azure runtime → user consent required for private actions.

NandaIndex enables identity-first discovery for individuals, when the underlying account identity is verifiably bound to the resolution record.

6.5 Summary

Across all contexts, the same resolution model applies:

- Enterprises with well-known AI Catalog endpoints → direct resolution; NandaIndex optional
- Enterprises on DNS-AID, gateways, or legacy registries → federated via NandaIndex
- SMBs → delegated infrastructure, resolved via NandaIndex
- Individuals → identity-based agents, resolved via NandaIndex

The key invariant is:

A single AI Catalog–formatted resolution layer maps identities to discovery systems, independent of infrastructure.

7. Case Study: AGNTCY

AGNTCY — the open-source collective spearheaded by Cisco Outshift, LangChain, and Galileo, and now a Linux Foundation project — provides a strong and relevant open-source example of distributed agent-directory infrastructure. It is both a demonstration of how layered interoperability works in practice and a direct collaborator in the global switchboard that this paper builds upon.

7.1 Overview

The AGNTCY Agent Directory (ADS) provides a practical implementation of the discovery and federation layers described in this paper. It demonstrates how AI Catalog and the Agentic Resource Discovery (ARD) specification can be integrated within a distributed system to support agent discovery at scale.

AGNTCY's discovery stack:

- Agent Directory Service (ADS): The backend registry, using IPFS Kademlia DHT for decentralized routing, OCI-aligned content addressing for immutable storage, and Sigstore-backed integrity verification. Deployable as open-source by any operator.
- AI Catalog interface: ADS exposes an AI Catalog-compatible query interface, making it reachable from any AI Catalog-aware tooling.
- ARD (Agent Resource Discovery): A standard read-only finder API that AGNTCY implements, allowing external agents to search the directory through the same interface used with other ARD-conforming systems (Hugging Face, GitHub).
- Agent Gateway (SLIM): Low-latency communication for multi-agent coordination, supporting peer-to-peer and pub/sub modes.

7.2 Mapping to the Layered Model

ADS naturally aligns with the layered architecture introduced in Section 3:

Layer	Role in ADS
Representation	AI Catalog entries describing agents and resources
Discovery	ARD APIs for querying capabilities
Infrastructure	Federation, routing, identity, and trust via ADS
Resolution	External (e.g., NandaIndex or equivalent layer)

ADS provides components relevant to the discovery and infrastructure layers, including distributed routing, artifact distribution, integrity verification, and catalog-compatible representations. However, it does not prescribe a universal resolution mechanism. Instead, it is designed to be resolved into, rather than to serve as a global entry point itself.

What this already achieves:

- An agent can be described in AGNTCY's ADS and reached via AI Catalog lookup, ARD search, or NandaIndex resolution — all from the same underlying entry

- AGNTCY interoperates with AI Catalog and DNS-AID, demonstrating the switchboard model
- The collaboration between NANDA and AGNTCY established the first working cross-organizational switchboard for agent discovery

7.3 Integration with Federated Resolution

Within the federated resolution architecture, ADS functions as a target discovery system.

A NandaIndex entry may, for example, resolve an identity to an ADS node:

```
{
  "identifier": "urn:ai:domain:acme.com",
  "displayName": "ACME Federated Directory",
  "type": "application/ai-registry+json",
  "url": "https://directory.acme.com/ard",
  "description": "ARD-compatible discovery endpoint backed by AGNTCY Agent Directory.",
  "tags": [
    "enterprise",
    "ard",
    "agent-directory"
  ],
  "publisher": {
    "identifier": "urn:ai:domain:acme.com",
    "displayName": "ACME Corp",
    "identityType": "dns"
  },
  "metadata": {
    "org.projectnanda.resolutionRole": "ard-endpoint",
    "org.projectnanda.preferredDiscovery": "ard",
    "org.projectnanda.nandaIndexRole": "federated-pointer"
  }
}
```

This illustrates how ADS integrates seamlessly into a federated resolution model:

- it does not replace other systems
- it participates as one discovery domain among many
- it enables secure, federated discovery once the correct entry point is known

7.4 Complementarity with NandaIndex

ADS and NandaIndex operate at different layers:

- ADS provides distributed directory and discovery infrastructure
- NandaIndex supplies an identity-based bootstrap mechanism that can direct a requester to an appropriate ADS deployment or other discovery system.

This separation is essential:

- ADS ensures that discovery works within and across federated directory nodes
- NandaIndex ensures that clients can find the correct directory or discovery mechanism in the first place

Together, they form a representative resolution path:

Identity → NandaIndex → ADS (ARD) → AI Catalog → Agent

7.5 What Remains Open — and Where NandaIndex Adds the Next Layer

- Resolution of SMB and individual identities that no single enterprise registry owns
- Bridging to sovereign national directories, telecom registries, and IoT systems as those continue to innovate
- Formalizing the AI Catalog entry format for non-catalog discovery paths (DNS-AID, gateways, hosted cards)

The global switchboard is the starting point. This paper proposes the next step: a formalized NandaIndex architecture that extends the same model to the full range of actors and emerging registries.

8. Conclusion

The emergence of the agentic web is driving rapid innovation in how agents are built, deployed, and interconnected. At the same time, it is exposing a fundamental challenge: agent discovery is fragmenting across multiple systems, formats, and operational models.

AI Catalog, ARD, DNS-based discovery, gateways, and platform registries each address critical aspects of the problem. Systems such as the AGNTCY Agent Directory demonstrate that federated discovery infrastructure is both feasible and already operational. However, as these mechanisms evolve independently, they risk forming disconnected discovery islands that limit interoperability and constrain the growth of multi-agent systems.

Innovation in agent discovery will keep producing new systems: sovereign national directories, telecom registries, IoT catalogs, sector-specific platforms. Each is solving a real problem. The goal is not to consolidate them into one system. The goal is to build bridges that let innovation continue without creating permanent isolation.

This paper argues that the key missing piece is not another discovery protocol, but a federated resolution layer.

By separating resolution from discovery, and by using AI Catalog as a universal, machine-readable index record, the proposed architecture enables identities to be mapped to the appropriate discovery mechanism — without requiring convergence on a single approach. In this model:

- AI Catalog provides a common representation of resources and resolution records
- ARD defines how discovery queries are performed
- ADS and similar systems provide federated infrastructure for storage, routing, and trust
- NandaIndex provides a distributed resolution layer that connects identities to discovery systems

For enterprises that publish AI Catalog at a well-known endpoint and own the domain, direct resolution already works — NandaIndex is not in the critical path. This is the clean case, and it should remain simple.

For everyone else:

- Enterprises on DNS-AID, gateways, or legacy registries get a single NandaIndex entry that makes their existing system reachable from the global switchboard

- SMBs like Moon Bakery can have runtime, agent card, and domain with three separate providers — NandaIndex resolves the identity regardless
- Individuals like john@hotmail.com get discoverability with no domain, no DNS, no enterprise infrastructure
- Emerging registries — telecom, IoT, sovereign — can join the switchboard through a verified external entry, without requiring changes to their internal discovery implementation.

The introduction of a global switchboard abstraction, instantiated through NandaIndex, allows heterogeneous discovery mechanisms to coexist while remaining interoperable. This shifts the focus of standardization:

from defining a single discovery system to enabling interoperability across many.

The implications are significant. By enabling discovery across organizational, technological, and operational boundaries, federated resolution unlocks the network effects required for large-scale, composable multi-agent systems. It allows capabilities to be reused, combined, and orchestrated dynamically, forming the basis of an open and interoperable Internet of Agents.

Future work will focus on:

- refining resolution semantics and identity models
- strengthening trust, verification, and policy mechanisms
- standardizing interfaces between resolution, discovery, and infrastructure layers
- enabling richer query routing and multi-hop resolution across federated systems

By bridging discovery systems rather than replacing them, federated resolution provides a path toward a scalable, decentralized, and interoperable agent ecosystem — one where agents can reliably find, trust, and collaborate with each other, regardless of where or how they are built.

References

[1] Agentic Resource Discovery (ARD) Specification — https://agenticresourcediscovery.org/

[2] AI Catalog Specification (Linux Foundation, AI Catalog Working Group) — https://agent-card.github.io/ai-catalog/

[3] AGNTCY Agent Directory (ADS) – GitHub Repository — https://github.com/agntcy/dir

[4] AGNTCY Project (Linux Foundation) — https://agntcy.org/

[5] DNS-AID: AI Agent Discovery via DNS / The Missing Layer in AI Interoperability: Discovery — https://aaif.io/blog/the-missing-layer-in-ai-interoperability-discovery/

[6] Nottingham, M. (2019). RFC 8615: Well-Known URIs — https://datatracker.ietf.org/doc/html/rfc8615

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

[8] SPIFFE Project – Secure Production Identity Framework for Everyone — https://spiffe.io/

[9] SPIRE: SPIFFE Runtime Environment — https://spiffe.io/spire/

[10] Maymounkov, P., & Mazieres, D. (2002). Kademlia: A Peer-to-Peer Information System Based on the XOR Metric

[11] libp2p Networking Stack — https://libp2p.io/

[12] Open Container Initiative (OCI) Distribution Specification — https://github.com/opencontainers/distribution-spec

[13] Model Context Protocol (MCP) — https://modelcontextprotocol.io/

[14] Agent Cards / A2A Specification — https://agent-card.github.io/

[15] Clark, D. D. (1988). The Design Philosophy of the DARPA Internet Protocols

[16] Cisco Outshift + MIT Media Lab: Building a Switchboard for the Internet of Agents: Integrating AGNTCY Directory and NANDA Index — https://outshift.cisco.com/blog/ai-ml/outshift-mit-agentic-web

[17] Raskar et al.: Beyond DNS: Unlocking the Internet of AI Agents via the NANDA Index and Verified AgentFacts — https://arxiv.org/abs/2507.14263

[18] KumbhDoot: Every Pilgrim Empowered with a Personal AI Agent and Civic AI Services — A Proposal for Agentic Management at Nashik Kumbh Mela 2027 — https://www.kumbhdoot.org/docs/KumbhDoot_Proposal.pdf

[19] Raskar et al.: Doot — The AI Agent for Every Indian Citizen: National Architecture of "Agent One" — https://digidoot.in/Doot_WhitePaper.pdf