Luca Muscariello

Distinguished Engineer, Cisco, LF AGNTCY & A2A TSC

6th of November 2025, A2A Summit, NYC

Democratizing Agent Discovery

A Decentralized Registry for Secure A2A Card Distribution and Collaboration

Executive Summary

Two phases define agent discovery:

- **Build-time** (developers assembling components) and
- **Runtime** (agents selecting peers).

Both hinge on two questions:

- what verifiable data (skills, cost, performance, trust) is required, and
- how to locate it in the wild.

In the Linux Foundation, A2A provides **Agent Cards** and AGNTCY a schema framework (OASF) to manipulate AI Cards (A2A and MCP) in an OCI based distributed Directory.

Naming Agent Cards

Why Naming Matters

- Discovery, caching, deduplication, trust
- Needs: **Integrity**, **Human Usability**, **Decentralization**

Two Naming Families

1. **Hash-Based (OCI digest)**
   - Guarantees: Integrity, immutability, global uniqueness
   - Limitations: Not human friendly, version evolution via new digest
2. **Human-Readable (URI)**
   - Strengths: Memorable, delegatable, governance & ownership
   - Limitations: Does not verify content; can drift

Phase 1: Build-Time Discovery

SBOM-like Verification Approach

- Developer-Centric process
- Component verification
- Integrity & provenance validation
- Supply chain security

Key Characteristics

- Human-driven decisions
- Pre-deployment verification
- Static controlled environment
- Cryptographic integrity/trust

Phase 2: Runtime Discovery

Autonomous Agent-to-Agent Discovery

- Dynamic Operation
- Agent-centric autonomy
- Real-time assessment
- Trust establishment
- Economic negotiations

Key Characteristics

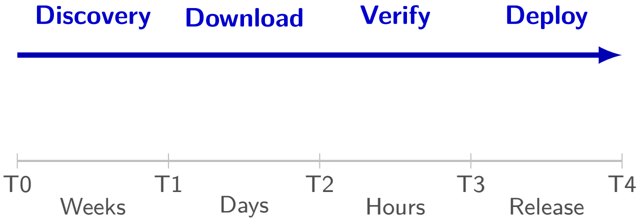
- Agent-driven autonomy
- Runtime adaptive evaluation
- Dynamic production environment
- Behavioral & economic trust layers

Phase 1 vs Phase 2 - Key Differences

Aspect	Phase 1 (Build-Time)	Phase 2 (Runtime)
Initiator	Human developers	Autonomous agents
Trust Focus	Provenance & integrity	Behavioral & economic
Environment	Controlled/static	Dynamic/production
Verification	Pre-deployment	Real-time
Relationships	One-time integration	Ongoing collaboration
Complexity	Compliance	Multi-dimensional trust

Phase 1: Build-Time Flow

Timeline: Weeks to Months



Discovery Download Verify Deploy

T0 T1 T2 T3 T4

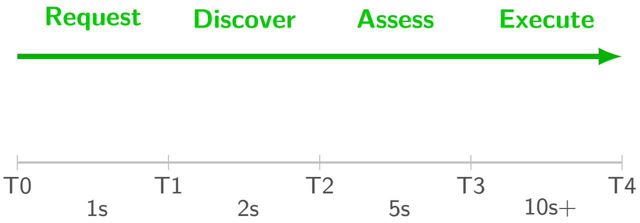
Weeks Days Hours Release

Key Phases

- **T0-T1**: Agent discovery and selection
- **T1-T2**: Download and integrity verification
- **T2-T3**: Development integration and testing
- **T3-T4**: Production deployment

Phase 2: Runtime Flow

Timeline: Seconds to Minutes



Key Phases

- **T0-T1**: Service need identification
- **T1-T2**: Peer/Agent discovery
- **T2-T3**: Real-time capability evaluation
- **T3-T4**: Economic negotiation & agreement
- **T4**: Task execution and completion

Trust Models - The Critical Distinction

Phase 1 (Build-Time)

- Package management security (integrity, provenance)
- Supply chain verification (provenance, SBOM mindset)
- Verification of claims: capabilities, costs.

Phase 2 (Runtime)

- Capability performance in live context
- Economic contract adherence
- Behavioral constraints & privacy
- Collaborative reliability across agents

Problem Dimensions

Dimension	Root Cause	Effect
Growth	Exploding number of agents & cards without shared quality signals	Noise > signal; duplication; evaluation fatigue
Fragmentation	Divergent schemas, packaging formats, naming & hosting silos	Low interoperability; brittle integration; poor reuse
Visibility Gap	Missing/verifiable descriptors (skills, cost, performance, provenance, trust)	Asymmetric decisions; higher risk & cost; slower adoption

Capability Assessment Complexity

Challenges

- Granularity: varying depth of skill claims
- Boundaries: unclear performance edges
- Evolution: capabilities drift over time
- Hidden / emergent skills: undocumented behaviors

Need

Standard taxonomy + benchmark suite for comparable evaluation.

Success Criteria - Phase 1

- Rich skill catalog coverage to make search efficient globally
- Transparent cost / resource profiles: observability and evaluation
- Reproducible performance baselines
- Verified security & integration readiness

Success Criteria - Phase 2

- Autonomous trust establishment
- Dynamic capability & quality scoring
- Efficient economic negotiation & settlement
- Privacy & reputation reinforced collaboration

MCP Registry: Benefits & Limitations

Benefits

- Simple metadata contract (server.json)
- Neutral & low operational burden

Limitations

- MCP-only scope
- Centralized control surface
- Dev-time discovery only (no runtime A2A)
- **Build-Time Only**: Development-time discovery, no runtime agent-to-agent

Introducing AGNTCY Directory

Core Elements

- OCI + ORAS: reuse hardened container infra
- OASF Open Agentic Schema Framework: a Framework to manage AI cards (MCP, A2A, others)
- Content-addressed (OCI digests): immutable artifacts
- DHT layer: decentralized lookup & federation to locate cards across OCI servers.
- Multi-registry: org boundaries + global reach

MCP Registry vs AGNTCY

Key Differences

MCP Registry	AGNTCY Directory
MCP servers only	All AI agents
Centralized	Distributed P2P
Dev-time only	Both dev & runtime
server.json	OCI + OASF
Single registry	Multi-registry

Complementary Approaches

- **MCP Registry**: Foundation for MCP servers
- **AGNTCY**: Broader ecosystem with runtime discovery

AGNTCY: OCI-Native Storage for AI Cards

Why OCI

- Seamless storage at build time and runtime
- Enterprise auth & RBAC already deployed
- Integrated signing (Notary / cosign) & provenance

Layout Pattern

tag (human alias) + digest (immutable) + taxonomy path (skill classification)

AGNTCY: Distributed Discovery Protocol (OCI extension)

Mapping Layers

- Skill (and other semantic taxonomies) → CID: capability index
- CID → OCI peer IDs: location / availability

Query Features

- Scalable global semantic search
- DHT replication for resilience

AGNTCY Security & Trust Model

Layers

- Integrity: digests / CIDs
- Provenance: Sigstore transparency + identity
- Isolation: org-scoped registries
- Reputation: evidence accumulation & scoring
- Zero-trust: verify each artifact & publisher

Implementation Roadmap in the Linux Foundation projects

Phase 1 (Foundation)

- Standardize Agent cards
- Build-time Discovery
- Runtime Discovery

Phase 2 (Integration)

- Standardize registry interfaces
- Federation & interconnection
- Support discovery in Kubernetes
- Web3 discovery

Call to Action

The Vision

A mature AI ecosystem where agents can be efficiently discovered, trusted, and integrated across both build and runtime contexts

Join the Linux Foundation projects

- Contribute to AI cards development
- Contribute to OCI-based standards development
- Deploy your registry and peer with the AGNCTY directory network
- Participate in AGNTCY pilot implementations
- Help build the future of AI agent collaboration

Q&A

Thank you for your attention!

Contact: lumuscar@cisco.com

Questions & Discussion

Links

- https://agntcy.org
- https://a2a-protocol.org
- https://github.com/a2aproject/A2A
- https://github.com/agntcy/dir
- https://github.com/agntcy/oasf
- https://github.com/agntcy/agentic-apps/tree/main/tourist_scheduling_system
- https://github.com/agntcy/dir/discussions/455 (To connect to the directory network)