

# **AGNTCY: Internet of Agents, by Committee**

AGNTCY, backed by Cisco and LangChain, builds the shared infrastructure for an Internet of Agents: directory, identity, schema, and observability for agents across the ecosystem. It standardizes everything around the agent. The canonical agent object at its center is the open question.

---

## **Initiative and Adoption Reality**

AGNTCY is an open-source initiative, backed by Cisco together with LangChain and other ecosystem participants, to build the shared infrastructure for an Internet of Agents: a common layer through which agents from many vendors can be discovered, composed, identified, and observed across the industry rather than within a single platform. Where a tool-access protocol standardizes how an agent reaches its tools and an agent-to-agent protocol standardizes how two agents talk, AGNTCY aims higher and wider, at the ecosystem fabric itself: directories so agents can be found, identity so they can be trusted, schemas so they can be composed, and observability so their collective behavior can be seen. It is a serious, well-sponsored attempt to keep the agent ecosystem open and interoperable rather than fragmenting into walled platforms, and the breadth of its ambition is exactly why the question of what an agent is becomes unavoidable for it.

## **The Structural Choice: Interoperability by Committee Around**

## **Agents-as-Services**

An ecosystem fabric needs a canonical notion of the thing it interconnects, and an initiative assembled by committee tends to standardize the interfaces around the agent, discovery, identity, observability, before it standardizes the agent object itself.

AGNTCY's layers describe how agents are found, attested, and watched, treating each agent largely as a service with a described capability surface. That is the pragmatic path to ecosystem interoperability, but it leaves the same gap one layer down that the tool and communication protocols leave: there is no single, portable, governed representation of the agent's intent, memory, policy, proposed mutation, and lineage that the directory lists, the identity layer attests, and the observability layer watches. The fabric standardizes everything around the agent and inherits whatever each agent's implementation happens to be as the agent.

## **What the Agent Schema Provides**

The agent schema supplies the canonical object an Internet of Agents needs at its center. A semantic agent object carries six canonical typed fields, intent, context, memory, policy reference, mutation descriptor, and lineage, so the entity a directory registers is a structured object, the entity an identity layer attests is governable by construction, and the entity an observability layer watches carries its own lineage rather than emitting opaque traces. Each of AGNTCY's layers is stronger when the agent it operates on is a semantic agent object: discovery over typed objects, identity over governable objects, composition over structurally validatable objects, observability over objects that record their own lineage. The initiative defines the fabric between agents; the schema defines the agent the fabric is built to carry.

## Composable, Not Competing

AGNTCY and the agent schema are not rivals; one is an ecosystem interoperability fabric and the other is the object that fabric interconnects. An Internet of Agents whose agents are semantic agent objects is an Internet of governable, portable objects rather than of opaque services, which is what an open and trustworthy ecosystem ultimately requires. The schema is the canonical agent definition that a by-committee fabric can adopt at its center without changing its directory, identity, or observability layers. No relationship, endorsement, or infringement is asserted; the comparison is architectural.

## Disclosure Scope

The semantic agent object and its six canonical typed fields, with structural validation, partial-agent support, and traceable semantic lineage that make the agent a portable, governable object, are disclosed in the agent schema filing (U.S. Application No. 19/452,651). This article compares that disclosed object model with the publicly described AGNTCY Internet of Agents initiative and positions the schema as the canonical agent object beneath the ecosystem fabric. References to AGNTCY, Cisco, and LangChain are to public materials and are used for comparison only.

---

### **Agent Schema** (</agent-schema>)

[All 36 steps → \(/inventive-steps\)](/inventive-steps)

Define what an autonomous agent is — structurally.

### **PRIMARY TECHNICAL DISCLOSURE**

- [Cognition-Compatible Semantic Agent Objects and Structural Validation \(/articles/cognition-compatible-semantic-agent-objects-and-structural-validation\)](/articles/cognition-compatible-semantic-agent-objects-and-structural-validation)

## SECONDARY TECHNICAL

- [Partial Agent Structural Validity: Fewer Fields, Still Deterministic \(/articles/agent-schema/partial-validity\)](/articles/agent-schema/partial-validity)
- [Minimum Two-Field Validation Threshold: The Floor of Semantic Structure \(/articles/agent-schema/two-field-threshold\)](/articles/agent-schema/two-field-threshold)
- [Field Interaction Rules: Deterministic Constraints Between Canonical Fields \(/articles/agent-schema/field-interaction-rules\)](/articles/agent-schema/field-interaction-rules)
- [Field-Based Role Typing: Agent Roles Derived From Structural Composition \(/articles/agent-schema/role-typing\)](/articles/agent-schema/role-typing)
- [Semantic Templates: Predefined Field Arrangements as Agent Class Contracts \(/articles/agent-schema/semantic-templates\)](/articles/agent-schema/semantic-templates)
- [Structural Scaffolding Logic: Resolving Missing Fields Through Inference or Defaulting \(/articles/agent-schema/scaffolding-logic\)](/articles/agent-schema/scaffolding-logic)
- [Field-Aware Default Resolution: Deterministic Behavior When Fields Are Absent \(/articles/agent-schema/default-resolution\)](/articles/agent-schema/default-resolution)
- [Traceable Semantic Lineage Graph: Mutation History Embedded in Agent Objects \(/articles/agent-schema/lineage-graph\)](/articles/agent-schema/lineage-graph)
- [Serialization With Stateless Compatibility: Reconstruction Without External Session State \(/articles/agent-schema/stateless-serialization\)](/articles/agent-schema/stateless-serialization)
- [Schema Governance Through Versioned Policies: Cross-Version Structural Interoperability \(/articles/agent-schema/versioned-policies\)](/articles/agent-schema/versioned-policies)

## APPLICATIONS · GENERAL

- [Enterprise AI Agent Interoperability Through Canonical Schema \(/articles/agent-schema/enterprise-interoperability\)](/articles/agent-schema/enterprise-interoperability)
- [Robotic System Standardization via Structural Field Composition \(/articles/agent-schema/robotic-standardization\)](/articles/agent-schema/robotic-standardization)
- [Multi-Vendor AI Agent Interoperability \(/articles/agent-schema/multi-vendor-ai-agents\)](/articles/agent-schema/multi-vendor-ai-agents)
- [Digital Twin Standardization Through Canonical Fields \(/articles/agent-schema/digital-twin-standardization\)](/articles/agent-schema/digital-twin-standardization)
- [Healthcare AI Agent Portability \(/articles/agent-schema/healthcare-agent-portability\)](/articles/agent-schema/healthcare-agent-portability)
- [Defense Coalition Interoperability \(/articles/agent-schema/defense-coalition-interop\)](/articles/agent-schema/defense-coalition-interop)
- [Insurance Claims Processing Through Standard Agents \(/articles/agent-schema/insurance-claims-agents\)](/articles/agent-schema/insurance-claims-agents)
- [Legacy System Integration via Schema Bridging \(/articles/agent-schema/legacy-system-integration\)](/articles/agent-schema/legacy-system-integration)

## APPLICATIONS · SPECIFIC

- [LangChain Built the Agent Framework. It Did Not Define What an Agent Is. \(/articles/agent-schema/langchain\)](/articles/agent-schema/langchain)
- [AutoGen Enabled Multi-Agent Conversations. The Agents Have No Structural Definition. \(/articles/agent-schema/autogen\)](/articles/agent-schema/autogen)
- [CrewAI Organized Agents Into Teams. The Agents Still Have No Schema. \(/articles/agent-schema/crewai\)](/articles/agent-schema/crewai)
- [Semantic Kernel Integrated AI Into Enterprise Code. The Agents It Creates Have No Schema. \(/articles/agent-schema/semantic-kernel\)](/articles/agent-schema/semantic-kernel)
- [OpenAI Assistants API Provides Agent Tooling. It Does Not Define Agent Structure. \(/articles/agent-schema/openai-assistants\)](/articles/agent-schema/openai-assistants)
- [Google Vertex AI Agents Provide Managed Agent Infrastructure. The Agents Have No Canonical Schema. \(/articles/agent-schema/google-vertex-agents\)](/articles/agent-schema/google-vertex-agents)
- [Amazon Bedrock Agents Orchestrate Foundation Models. The Agents Have No Structural Definition. \(/articles/agent-schema/amazon-bedrock-agents\)](/articles/agent-schema/amazon-bedrock-agents)
- [Haystack Built Composable NLP Pipelines. The Pipeline Components Have No Agent Schema. \(/articles/agent-schema/haystack\)](/articles/agent-schema/haystack)
- [LlamaIndex Built the Data Framework for LLM Applications. The Data Objects Have No Agent Schema. \(/articles/agent-schema/llamaindex\)](/articles/agent-schema/llamaindex)
- [Dify Made LLM Application Development Visual. The Applications Have No Agent Schema. \(/articles/agent-schema/dify\)](/articles/agent-schema/dify)
- [Microsoft AutoGen and CrewAI Multi-Agent Frameworks \(/articles/agent-schema/autogen-crewai\)](/articles/agent-schema/autogen-crewai)
- [LangChain and LangGraph Agent Framework \(/articles/agent-schema/langchain-langgraph\)](/articles/agent-schema/langchain-langgraph)
- [LlamaIndex Agent Framework \(/articles/agent-schema/llamaindex-agents\)](/articles/agent-schema/llamaindex-agents)
- [ROS 2 \(Robot Operating System\) Middleware \(/articles/agent-schema/ros2-robotics\)](/articles/agent-schema/ros2-robotics)
- [Cursor AI-Native Code Editor \(/articles/agent-schema/cursor-coding-agent\)](/articles/agent-schema/cursor-coding-agent)
- [Replit Agent and Replit Workspaces \(/articles/agent-schema/replit-agent\)](/articles/agent-schema/replit-agent)
- [Model Context Protocol: Agent Tool Use as Protocol \(/articles/agent-schema/anthropic-mcp\)](/articles/agent-schema/anthropic-mcp)
- [Google Agent2Agent: Inter-Agent Communication, Top-Down \(/articles/agent-schema/google-a2a\)](/articles/agent-schema/google-a2a)
- **[AGNTCY: Internet of Agents, by Committee \(/articles/agent-schema/cisco-langchain-agntcy\)](/articles/agent-schema/cisco-langchain-agntcy)**

---

[Agent Schema overview → \(/agent-schema\)](/agent-schema)

