

Google Agent2Agent: Inter-Agent Communication, Top-Down

Google's Agent2Agent protocol standardizes how agents discover one another and exchange tasks, so an agent built by one team can delegate to an agent built by another. It standardizes the conversation between agents. It leaves each agent an opaque service.

Protocol and Adoption Reality

Google's Agent2Agent protocol, A2A, standardizes communication between agents rather than between an agent and its tools. As publicly described, an agent publishes an agent card that advertises its capabilities and endpoint, other agents discover it, and they exchange tasks and messages over a defined protocol so that an agent built by one team or vendor can delegate to and collaborate with an agent built by another. It addresses a genuine and growing problem: as organizations field many agents from many sources, those agents need a common way to find each other and hand work back and forth, and without a standard every pairing is a custom integration. A2A, introduced with broad industry backing, is a credible answer to inter-agent communication, complementary to tool-access protocols rather than overlapping them.

The Structural Choice: An Agent Is an Endpoint That Advertises Capabilities

A2A models an agent as a service: a discoverable endpoint with a capability card and a messaging interface. The protocol defines how agents announce themselves, how tasks are framed and exchanged, and how results return, which is exactly what inter-agent communication requires. What it deliberately does not define is the internal constitution of the agent behind the endpoint. The agent card describes what an agent can do, not what an agent is made of: there is no standardized, portable representation of the agent's intent, memory, governing policy, proposed mutation, or lineage as a structured object that could move between systems intact. A2A standardizes the conversation between agents while leaving each agent an opaque service whose internal state is its own implementation's concern.

What the Agent Schema Provides

The agent schema standardizes the thing behind the endpoint. A semantic agent object is a structured entity carrying six canonical typed fields, intent, context, memory, policy reference, mutation descriptor, and lineage, so the agent is a portable, serializable, governable object rather than an opaque service. Inter-agent collaboration over A2A gains from this directly: when the agents that discover and delegate to each other are semantic agent objects, a delegation can carry the governing policy and the lineage of the work, the receiving agent can validate the structure of what it is asked to do, and the trail of which agent did what under what authority is recorded rather than lost at the endpoint boundary. A2A defines how agents talk; the schema defines what each agent is as it talks, and the combination makes multi-agent collaboration governable rather than merely connected.

Composable, Not Competing

A2A and the agent schema sit at different layers: a communication protocol between agents, and a structural definition of the agent. Agents represented as semantic agent objects can discover and delegate to one another over A2A, so the two are additive. The schema supplies the portable, governed object that A2A's messaging moves between, turning a network of opaque services into a network of governable objects. 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 Google's publicly documented Agent2Agent protocol and positions the schema as the agent-object layer beneath A2A's inter-agent communication. References to A2A are to its public specification 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)

