OpenAI Assistants API Provides Agent Tooling. It Does Not Define Agent Structure.

by Nick Clark | Published March 28, 2026 | PDF

OpenAI's Assistants API provides a managed runtime for building AI agents with tools, files, and threads. Agents are created through API configuration: a model, a set of tools, instructions, and optional file references. The configuration creates a capable agent. But there is no canonical schema defining what an agent structurally is. Different assistants can have entirely different configurations with no shared structural definition. The gap is between configurable agent tooling and a canonical schema that defines the structural identity of an agent.

OpenAI's Assistants API represents a significant step toward accessible agent development. The managed runtime, code interpreter, and retrieval capabilities address real needs. The gap described here is about structural agent definition, not about API capability.

Configuration is not schema

An OpenAI Assistant is configured with a model, tools, instructions, and optional files. This configuration is a set of parameters. It is not a structural definition. Two assistants with different configurations are both valid assistants. There is no schema saying which fields are required, how fields relate to each other, or what makes an assistant structurally valid.

An assistant without memory is valid. An assistant without governance is valid. An assistant without lineage is valid. Everything is optional because there is no structural definition of what an agent must be.

Threads without governed memory

The Assistants API provides threads for conversation history. Threads are message sequences, not governed memory. There is no schema for memory structure, no lineage tracking, and no governance on memory mutations. A thread is a conversation log. Governed memory is a typed, lineage-preserving, policy-constrained field of the agent.

What a canonical agent schema provides

A canonical agent schema defines six typed fields: identity, memory, governance, capabilities, execution state, and lineage. These fields are not optional. They define what an agent structurally is. With this schema, OpenAI Assistants could be structurally validated, interoperable with agents built on other platforms, and governable by execution platforms that understand the schema.

Agent Schema All 21 steps →

Define what an autonomous agent is — structurally.

Patent
US 19/452,651 · filed
Primary Technical Disclosure
○ Cognition-Compatible Semantic Agent Objects and Structural Validation
Secondary Technical
○ Partial Agent Structural Validity: Fewer Fields, Still Deterministic○ Minimum Two-Field Validation Threshold: The Floor of Semantic Structure○ Field Interaction Rules: Deterministic Constraints Between Canonical Fields○ Field-Based Role Typing: Agent Roles Derived From Structural Composition○ Semantic Templates: Predefined Field Arrangements as Agent Class Contracts○ Structural Scaffolding Logic: Resolving Missing Fields Through Inference or Defaulting○ Field-Aware Default Resolution: Deterministic Behavior When Fields Are Absent○ Traceable Semantic Lineage Graph: Mutation History Embedded in Agent Objects○ Serialization With Stateless Compatibility: Reconstruction Without External Session State○ Schema Governance Through Versioned Policies: Cross-Version Structural Interoperability
Applications (General)
○ Enterprise AI Agent Interoperability Through Canonical Schema○ Robotic System Standardization via Structural Field Composition○ Multi-Vendor AI Agent Interoperability○ Digital Twin Standardization Through Canonical Fields○ Healthcare AI Agent Portability○ Defense Coalition Interoperability○ Insurance Claims Processing Through Standard Agents○ Legacy System Integration via Schema Bridging
Applications (Specific)
○ LangChain Built the Agent Framework. It Did Not Define What an Agent Is.○ AutoGen Enabled Multi-Agent Conversations. The Agents Have No Structural Definition.○ CrewAI Organized Agents Into Teams. The Agents Still Have No Schema.○ Semantic Kernel Integrated AI Into Enterprise Code. The Agents It Creates Have No Schema.● OpenAI Assistants API Provides Agent Tooling. It Does Not Define Agent Structure.○ Google Vertex AI Agents Provide Managed Agent Infrastructure. The Agents Have No Canonical Schema.○ Amazon Bedrock Agents Orchestrate Foundation Models. The Agents Have No Structural Definition.○ Haystack Built Composable NLP Pipelines. The Pipeline Components Have No Agent Schema.○ LlamaIndex Built the Data Framework for LLM Applications. The Data Objects Have No Agent Schema.○ Dify Made LLM Application Development Visual. The Applications Have No Agent Schema.
Agent Schema overview →
AQ
deterministic
autonomy

-
- nick@qu3ry.net
- 72 28 14 36 01

Invented by Nick Clark | Founding Investors: Devin Wilkie