Haystack Built Composable NLP Pipelines. The Pipeline Components Have No Agent Schema.

by Nick Clark | Published March 28, 2026 | PDF

Haystack provides a composable framework for building NLP and RAG pipelines with retrievers, readers, generators, and custom components. The pipeline composition model is flexible. But Haystack pipeline components are functional units with inputs and outputs, not structurally defined agents. There is no canonical schema for what a component or agent is, how governance relates to capability, or how memory and identity persist across pipeline executions. The gap is between composable NLP tooling and canonical agent definition.

Haystack's composable pipeline architecture and rich component ecosystem provide genuine NLP engineering value. The gap described here is about structural agent definition, not about pipeline flexibility.

Components without structural identity

Haystack components are Python classes with defined inputs and outputs. A retriever takes a query and returns documents. A generator takes a prompt and returns text. Components are functional units. They have no structural identity, no governance field, no memory that persists across invocations, and no lineage.

A pipeline assembled from Haystack components is a data flow graph. It is not a structurally defined agent with canonical fields.

Pipeline state without governed memory

Haystack pipelines pass data between components through typed connections. The data flowing through a pipeline is transient. There is no governed memory accumulating across pipeline executions, no lineage tracking how memory evolved, and no policy constraining memory mutations.

What a canonical agent schema provides

A canonical agent schema would allow Haystack pipeline assemblies to be wrapped as structurally defined agents with identity, memory, governance, capabilities, execution state, and lineage as typed fields. The pipeline would provide the processing capability. The schema would provide the structural definition that makes the agent validatable, governable, and interoperable.

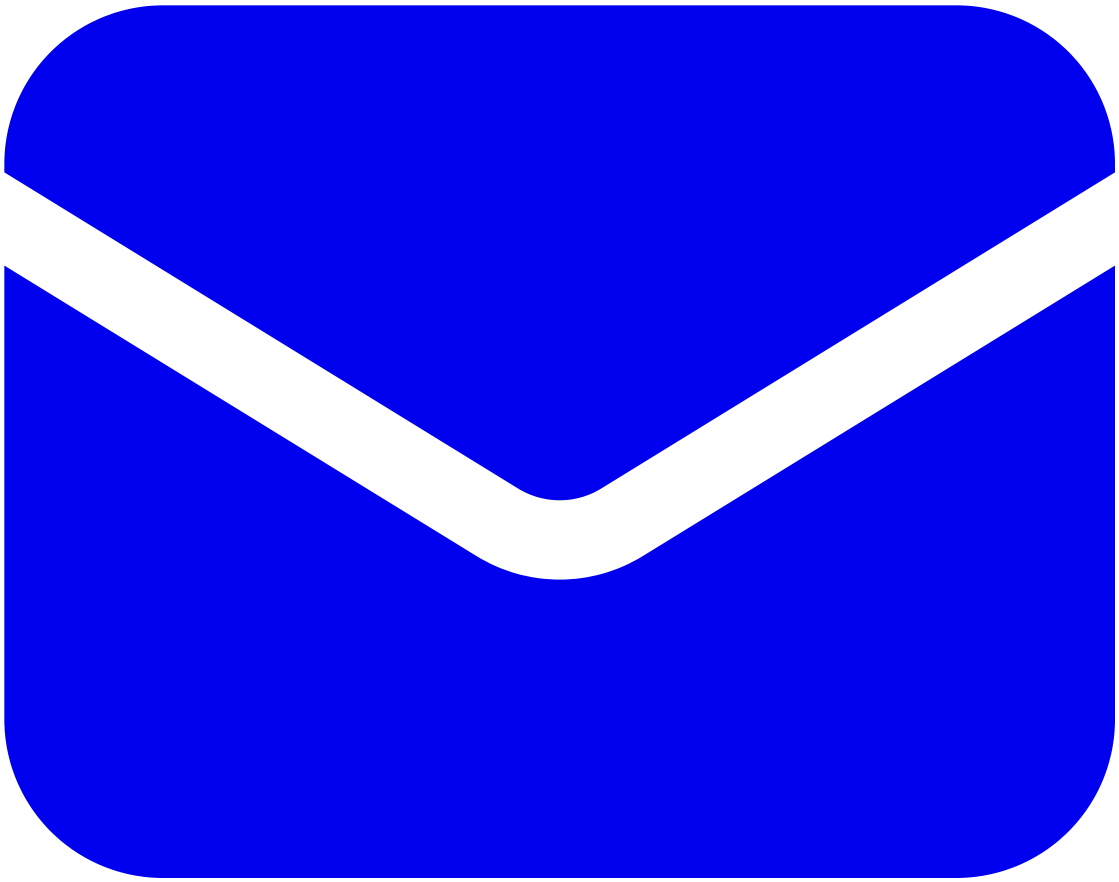
Agent Schema All 21 steps →

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

Legal

Last updated: 2026-03-03



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

Invented by Nick Clark | Founding Investors: Devin Wilkie