



[Home](#) [Licensing](#) [Patents](#) [Articles](#)

Amazon Bedrock Agents Orchestrate Foundation Models. The Agents Have No Structural Definition.

by [Nick Clark](#) | Published March 28, 2026 | [PDF](#)

Amazon Bedrock Agents provides managed agent orchestration for foundation models with action groups, knowledge bases, and guardrails. Agents can invoke APIs, query knowledge bases, and maintain session context. The orchestration is capable. But Bedrock agents have no canonical schema. Each agent is a configuration of action groups and guardrails, not a structurally defined object. The gap is between agent orchestration capability and a canonical schema that defines what an agent is.

Amazon Bedrock's agent orchestration with action groups and guardrails provides practical enterprise agent capabilities. The gap described here is about structural agent definition.

Action groups without capability schema

Bedrock agents use action groups to define what actions an agent can take. Action groups specify API schemas and descriptions. But action groups are configuration, not a typed capability field of a canonical schema. There is no structural relationship between an agent's action groups and its governance, memory, or identity. The capabilities are configured independently of the agent's structural definition because there is no structural definition.

Guardrails as external policy, not intrinsic governance

Bedrock guardrails filter inputs and outputs against content policies. Guardrails are useful safety mechanisms. But guardrails are external filters applied to the agent's interactions. They are not intrinsic governance fields of the agent. The agent does not carry its governance. Governance is applied from outside.

What a canonical agent schema provides

A canonical agent schema would make governance an intrinsic typed field of the agent, not an external filter. Capabilities would be a typed field with defined relationships to governance and identity. The agent would carry its own structural definition, validatable by any platform that understands the schema, portable across providers, and governable through intrinsic fields rather than external guardrails.

[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

Subject to one or more pending U.S. and international patent applications, see [Patents](#) for the current list and status. No license, express or implied, is granted. Any use requires a separate written agreement—see [Licensing](#). Patent applications referenced on this site are pending. Claim scope, if any, is subject to examination and may issue in altered form or not at all. See [Legal](#) for terms and conditions.

Adaptive Query™ is a trademark of Nicholas Clark. U.S. federal registration is pending. federal registration. AQ™, AQ Inside™, Adaptive Index™, Adaptive Network™, Semantic Agent™, @AQ™, AQID™, and Adaptive Coin™ are used as trademarks in connection with the Adaptive Query platform and brand. Other names may be trademarks of their respective owners.

Platform operated by Adaptive Query LLC, which provides patent and trademark licensing services. Copyright © 2025-2026 Nicholas Clark. All rights reserved.

Last updated: 2026-03-03



- [Inventive Steps](#)
- [Licensing](#)
- [Patents](#)
- [Articles](#)
- [Legal](#)
- [Opportunities](#)
- [Sitemap](#)



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



[Invented by Nick Clark](#) | Founding Investors: Devin Wilkie