



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

Healthcare AI Agent Portability

by [Nick Clark](#) | Published March 27, 2026 | [PDF](#)

Healthcare institutions are deploying AI agents for clinical decision support, patient monitoring, and administrative automation. Each agent is locked to the vendor platform that built it, preventing institutions from migrating agents between platforms, sharing agents across institutions, or verifying agent behavior through a common structural standard. A canonical agent schema enables healthcare AI portability by making governance, clinical memory, and compliance lineage intrinsic typed fields that persist regardless of platform.

The vendor lock-in problem in healthcare AI

Healthcare AI agents are deployed as platform-specific implementations. A clinical decision support agent built on one vendor's platform carries its clinical knowledge, governance rules, and operational history in vendor-specific formats. Migrating to a different platform requires rebuilding the agent

from scratch, losing the accumulated clinical experience and compliance history that made the original agent valuable.

This lock-in creates systemic risk. An institution that has deployed agents across clinical workflows becomes dependent on the vendor's continued existence, pricing decisions, and platform evolution. If the vendor discontinues the product, changes terms, or suffers a security breach, every clinical workflow that depends on the agent is affected simultaneously.

Cross-institutional agent sharing, which could accelerate clinical AI adoption by allowing institutions to share proven agents rather than building from scratch, is impossible when agents carry their state in vendor-specific formats. A sepsis detection agent validated at one hospital cannot be transferred to another hospital running a different platform without complete reimplementing.

Why interface standards do not solve portability

Healthcare integration standards like FHIR define how systems exchange clinical data. They do not define the structural requirements for an AI agent that consumes, processes, and acts on that data. An agent that conforms to FHIR for data exchange can still be structurally locked to its vendor platform for governance, memory, lineage, and execution logic.

The portability problem is not at the interface layer. It is at the agent layer. The agent's internal structure, what it knows, what it is allowed to do, what it has done, and how those properties relate to each other, must be portable for the agent itself to be portable.

How the canonical agent schema addresses this

A canonical agent schema defines the structural requirements for a healthcare AI agent. The governance field carries HIPAA compliance rules, institutional policies, and clinical scope constraints as typed, inspectable values. The memory field carries clinical knowledge and patient context. The lineage field preserves every clinical decision, data access, and state mutation with full provenance. The execution eligibility field determines whether the agent is authorized to execute in the current clinical context.

Portability becomes structural. When an institution migrates an agent from one platform to another, the canonical fields carry the complete agent state. The receiving platform validates the agent's governance against its own policy, inspects the lineage for compliance, and resumes operation. No clinical knowledge is lost. No compliance history is discarded.

Cross-institutional sharing becomes possible because the canonical schema provides a common structural contract. A sepsis detection agent validated at one institution carries its validation history in its lineage, its clinical constraints in its governance, and its accumulated knowledge in its memory. The receiving institution evaluates the agent against its own policy through the canonical fields rather than through platform-specific inspection.

What implementation looks like

A healthcare institution deploying agents on the canonical schema ensures that every clinical AI agent carries the six canonical fields regardless of which vendor built it. The governance field includes HIPAA constraints, institutional IRB requirements, and clinical scope limitations. The lineage field records every clinical interaction with enough detail for regulatory audit.

For clinical AI governance, the canonical schema provides a uniform inspection surface. Compliance officers evaluate governance fields using the same tools across all agents, regardless of vendor. Audit trails are structural, not reconstructed from vendor-specific logs.

For AI procurement, the canonical schema becomes a purchasing requirement. Vendors that implement the canonical fields provide agents that are portable, auditable, and interoperable by construction. Institutions avoid lock-in not by negotiating contractual exit terms but by requiring structural portability from the outset.

For clinical validation, an agent's lineage provides a machine-readable record of every clinical decision, data access, and outcome observation throughout its operational history. Validating the agent for a new clinical context involves inspecting its lineage rather than repeating the entire validation process from scratch.

[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