Nick Clark · Published June 29, 2026 · PDF

Moving Governed AI Agents Across Clouds and Vendors Without Losing Identity: Substrate Portability via the Cross-Patent Architecture

Enterprises that run autonomous agents are locked to whichever cloud or vendor holds the agent's state, so a migration means rebuilding identity, governance, and behavioral history from scratch (or accepting that they are lost). The Cross-Patent Architecture, disclosed in United States Patent Application 19/647,395, addresses this by making the agent carry its own complete cognitive state while the execution substrate provides only computation, so a governed agent can move across infrastructure and providers without losing identity or governance. It composes a state-preserving transport tier, trust-slope substrate identity with revocation, a cryptographic policy tier, and an integrity-trajectory governance authority into one portable, vendor-neutral whole.

What This Application Specifies

This application describes how an autonomous, governed AI agent can move between execution substrates (different servers, devices, network nodes, clouds, and vendors) without losing the identity, governance constraints, and behavioral history that make it

the agent it is. The mechanism is an architectural inversion, disclosed in United States Patent Application 19/647,395 (the Cross-Patent Architecture, built on the cognition platform's cross-domain coherence and composition).

In conventional distributed systems, the server holds state and retains authority over the data objects passing through it; the data object is a passive payload. The same assumption pervades conventional AI infrastructure, where the inference server holds the model weights and the prompt and response carry no persistent state, no governance, and no capacity for self-assessment. This application inverts that relationship. The semantic agent (the traveling object) carries its own complete cognitive state: its affective disposition, integrity field, confidence assessment, capability awareness, policy constraints, lineage, and the bidirectional feedback pathways of its coherence engine. The execution substrate provides computational resources and environmental conditions but does not hold authority over the agent's state transitions, does not determine its behavioral trajectory, and cannot alter its lineage without producing a detectable trust-slope discontinuity. As the specification puts it, the agent is the locus of intelligence and the substrate is the locus of resources.

For portability across substrates and vendors, this composes several tiers of the portfolio into one unified system. A state-preserving transport tier carries the agent's complete state (payload, memory field, cryptographic signatures, and the cognitive domain fields) across network hops, so the agent's state is carried intact rather than reconstructed at the destination. A trust-slope identity tier establishes substrate and agent identity through accumulated behavioral continuity rather than static credential matching, which is what makes revocation meaningful. A cryptographic policy tier enforces signed governance constraints on every state transition, including the export and import events of a migration. And an integrity-trajectory governance authority lets the agent evaluate a governance claim against its own accumulated record of normative consistency, not solely against a signature. The cognition application is the foundation

that discloses this cross-domain coherence and composition; the substrate execution, content anchoring, spatial mesh, and physical tiers are referenced by category as sibling portfolio filings.

Why It Matters

Vendor lock-in for stateful agents is not a billing inconvenience; it is a structural risk. When the substrate holds the agent's state, the provider owns the agent's identity. Migrating to a different cloud, a different vendor, or an on-premises deployment means the behavioral history, the relational memory built up with users, the governance bindings, and the accumulated integrity record are stranded with the old provider or must be reconstructed, which breaks continuity and resets trust.

The specification makes explicit why the inversion is a prerequisite rather than a preference. If the substrate held authority over the agent's state, the agent could not migrate between substrates while preserving behavioral continuity. If the substrate determined the agent's trajectory, the coherence engine could not operate as an internal self-regulatory mechanism. If the substrate retained the agent's state between interactions, the lineage would fragment across substrates, destroying the deterministic reconstructibility that is the foundation of trust-slope validation. Portability, in other words, is not bolted on; it falls out of the same structure that makes the agent governable in the first place.

The result the specification names is a behavioral continuity guarantee: the assurance that an agent exhibits identical state and behavioral dynamics regardless of which substrate provides the underlying computational resources. For a regulated enterprise, that is the difference between an agent that is a portable, auditable asset and an agent that is a hostage of its current host.

How It Composes With the Domain

Consider an enterprise that runs a customer-facing support agent on one cloud provider and needs to relocate it to a second provider (for cost, data-residency, or resilience reasons) without the agent forgetting who it is or escaping its governance.

A migration begins as a governed export. Because the substrate retains no authority over the agent's cognitive state and no agent state between interactions, the complete cognitive state exists as a self-contained, structured data object: all cognitive domain fields, the coherence engine's coupling functions and feedback configurations, the experiential observation store, the per-entity relational records, the goal queue, the lineage field, and all policy bindings. The user exports that object from the source substrate, transfers it to a substrate operated by a different provider, and imports it to resume operation with identical field values, identical relational histories, and identical behavioral disposition. The export is self-sufficient: no substrate-resident state, no provider-maintained index, and no platform-specific configuration is needed to reconstruct the agent. The exported state is the user's data, and the export format, integrity protections, import validation, and substrate-compatibility evaluation are all governed by the same policy infrastructure that governs every other state transition. Each export and import is recorded in lineage as a governed migration event, preserving reconstructibility across the provider boundary.

During the move itself, the agent occupies a transit cognitive state that arises from the interaction of the execution tier and the transport tier. In transit the agent is neither executing (no substrate is providing compute), nor in non-executing cognitive mode (no compute is available for speculative reasoning), nor dormant (its state is actively in transport). The transit state freezes the cognitive domain field values at their pre-transit levels while the lineage field continues to accumulate transit events (departure timestamp, transport path, and arrival validation). On arrival at the destination substrate, a capability evaluation confirms the destination provides sufficient resources, and the confidence governor evaluates whether the transit duration, transit

path characteristics, and destination capabilities warrant a confidence adjustment before execution resumes. The agent does not simply pick up where it left off; it re-establishes its readiness against the new environment.

Substrate identity with revocation governs the case where a host can no longer be trusted mid-flight. When an agent is actively executing on a substrate and that substrate's dynamic device hash validation fails (indicating the substrate's identity continuity has been compromised), the capability envelope immediately reclassifies the substrate as unverified, the confidence governor receives a reduced readiness signal proportional to the severity of the failure, and the agent transitions to non-executing cognitive mode pending re-validation or migration to a verified substrate. Crucially, the agent's cognitive state is preserved throughout, because the cognitive domain fields are carried by the agent, not by the substrate. Revoking trust in a host does not cost the agent its identity.

Two further compositions keep governance intact across the move. Policy freshness under asynchronous execution: when an agent resumes after an asynchronous interval and detects that the governance policy in force at suspension has been superseded, the confidence governor treats stale policy as a confidence input, reducing confidence in proportion to the governance significance of the change and, if the reduction crosses the authorization threshold, transitioning to non-executing mode and issuing an inquiry for the current policy before acting. And the integrity-trajectory governance authority: when an agent arrives under a governance policy signed by an authority its trust-slope history does not recognize, it evaluates the claim against its own integrity trajectory (the accumulated pattern of normative consistency in its lineage) rather than relying on signature validation alone. A migrated agent can therefore accept legitimate governance in a new environment without blindly accepting any policy that happens to be signed.

What This Enables

The application enables a range of portability embodiments. An agent can move between heterogeneous substrates (centralized, federated, decentralized, and embodied) while carrying its identity, governance, and behavioral history, producing identical behavioral dynamics on each. A user can treat the agent's cognitive state as an owned, exportable, importable artifact and relocate it across providers, which structurally undoes vendor lock-in for stateful agents. An operator can revoke trust in a compromised host without losing the agent, falling back to non-executing cognitive mode until a verified substrate is available. A compliance team can require that every migration appear in lineage as a governed event, so a relocation across clouds remains auditable end to end.

Because a compressed narrative identity (the agent's distilled self-model of who it is and how it has behaved) travels with the agent and constrains its forecasting and integrity evaluation, the specification describes long-horizon individual persistence across thousands of interactions and multiple substrate migrations without per-context reprogramming. The same primitives support multi-vendor resilience strategies, data-residency-driven relocation, and on-premises-to-cloud or cloud-to-edge moves, all governed by the one policy infrastructure that governs the agent everywhere it runs.

Boundary Conditions

The behavioral continuity guarantee applies to the agent's cognitive state and behavioral dynamics; it does not claim that every destination substrate offers equivalent raw performance or capability. The capability evaluation and confidence adjustment on arrival exist precisely because destinations differ, and a destination with insufficient resources may keep the agent in non-executing cognitive mode rather than resuming execution. Portability also presumes a destination substrate that participates in the disclosed transport, identity, and policy mechanisms; moving to infrastructure that does not honor them is outside what the specification describes.

This application discloses the cognition platform's cross-domain coherence and composition. The substrate execution, content anchoring, spatial mesh, and physical layers are referenced by category as sibling portfolio filings and are not claimed here as this application's specifics. Domain framing such as cloud cost, data residency, and multi-vendor resilience is an enabling deployment context, not part of the patent claims, and no performance numbers, benchmarks, or interoperability guarantees with any particular commercial platform are asserted.

Disclosure Scope

The mechanisms described here (the architectural inversion, state-preserving transport of cognitive domain fields, the transit cognitive state, substrate identity revocation through the capability envelope and dynamic device hash, policy freshness under asynchronous execution, the integrity-trajectory governance authority, and user-owned portable agent state) are disclosed in United States Patent Application 19/647,395. The portability across substrates and vendors described in this article is a faithful, enabling implementation of that disclosed technology. References to clouds, vendors, data-residency regimes, regulatory expectations, and enterprise migration scenarios are external domain and market context provided to illustrate application; they are not patent claims and do not extend the scope of the application. Sibling portfolio tiers are identified by category for architectural context and are not claimed as this application's contributions.

Cross-Patent Architecture (/cross-patent-architecture)

All 40 steps → (/inventive-steps)

Cross-cutting architectural principles that compose every primitive into a coherent platform.

Chapter 1 (/patents/19-647395/chapters/foundation)

PRIMARY TECHNICAL DISCLOSURE

- Cross-Patent Architecture, Articles (/articles/cross-patent-architecture)

SECONDARY TECHNICAL

- Transit Cognitive State (/articles/cross-patent-architecture/transit-cognitive-state)
- Substrate Identity Revocation During Active Cognition (/articles/cross-patent-architecture/substrate-identity-revocation)
- Policy Freshness Across Asynchronous Execution (/articles/cross-patent-architecture/policy-freshness-asynchronous-execution)
- Governance Authority Evaluation via Integrity Trajectory (/articles/cross-patent-architecture/governance-authority-integrity-trajectory)
- Discovery Agent as Schema-Conformant Index Traverser (/articles/cross-patent-architecture/discovery-agent-schema-index-traverser)
- Unified Substrate for Governed Information Acquisition (/articles/cross-patent-architecture/cross-tier-navigation-world-as-model)

APPLICATIONS · GENERAL

- One Governed Platform, Not Four Integrated Systems: A Unified Architecture Spine for Agent Execution, Cognition, Content, and Spatial Tiers (/articles/cross-patent-architecture/unified-governed-platform)
- World-as-Model Systems: Navigating the Physical World, Cognition, and Discovery as One Governed Model (/articles/cross-patent-architecture/world-as-model-systems)
- End-to-End Lineage and Audit: Reconstructing Any Agent Action Across Every Tier of the Stack (/articles/cross-patent-architecture/end-to-end-lineage-and-audit)
- **Moving Governed AI Agents Across Clouds and Vendors Without Losing Identity: Substrate Portability via the Cross-Patent Architecture (/articles/cross-patent-architecture/portability-across-substrates)**
- Cross-Patent Architecture: Why a Coherent AI Platform Needs a Shared Governance Authority at the Foundation, Not as a Feature (/articles/cross-patent-architecture/ai-platform-foundation)
- Regulated Cross-Domain Deployment: One Governance Authority and Policy-Freshness Model Across Every Tier of an End-to-End System (/articles/cross-patent-architecture/regulated-cross-domain-deployment)

APPLICATIONS · SPECIFIC

- Palantir Foundry and AIP (the ontology-based data/operations platform plus its AI orchestration layer) vs a cross-tier governed architecture: where does end-to-end action attribution live? (/articles/cross-patent-architecture/palantir-foundry-aip)
- Microsoft's integrated AI stack (Azure AI Foundry, Microsoft Fabric, Entra, and Copilot) vs a single cross-domain governance architecture: how do coherence and one governance chain differ from an integrated product suite? (/articles/cross-patent-architecture/microsoft-ai-stack)
- Amazon Web Services' integrated AI/data stack (Bedrock, SageMaker, and surrounding data/identity services) vs a unified cross-tier governed agent architecture (/articles/cross-patent-architecture/aws-ai-stack)
- NVIDIA's full-stack AI platform (NVIDIA AI Enterprise, NIM microservices, and the CUDA/hardware-to-software stack) vs a substrate-independent governance architecture (/articles/cross-patent-architecture/nvidia-ai-enterprise)
- Databricks Data Intelligence Platform (lakehouse plus Mosaic AI, Unity Catalog governance, and agent tooling) vs an agent-resident cross-patent architecture: where governance lives (/articles/cross-patent-architecture/databricks-data-intelligence)

Cross-Patent Architecture overview → (/cross-patent-architecture)