Runtime Signed Adaptation Artifacts vs Training-Time Mutation

by Nick Clark | Published April 25, 2026

Training-Time vs Runtime Adaptation

Cognition's existing scope addresses training-time governance: depth-selective gradient routing, per-example provenance, training-data rights compliance. The architecture works on the training pipeline.

Runtime adaptation is different. The base model is trained and frozen; the adaptation lives outside the model in artifacts that load at inference time. LoRA weights modify specific layer projections. RAG indices supply retrieval context. Prompt configurations shape generation. MoE routing selects expert subnetworks. None of these touch training; all of them affect runtime behavior.

Why Runtime Needs Its Own Architecture

The skill marketplace that AI-agent platforms are building (Anthropic Skills, OpenAI Custom Actions, Google Gemini Extensions, Microsoft Copilot Studio, the HuggingFace ecosystem) is a runtime-adaptation marketplace. The economic substrate is signed adaptation artifacts loaded into base models for specific tasks.

Training-time governance does not address: which artifact applies to this inference, who certified it for this consumer's deployment, what dependencies must be active, what happens when the certifying authority revokes the artifact, and how does the consumer's policy modulate third-party-authored artifacts. These are the questions that runtime governance answers.

How Runtime Artifacts Are Governed

Each artifact carries a credential from its authoring authority, declared compatibility scope, declared dependencies, declared training provenance, and a content-hash binding to the actual artifact bytes. The consuming system runs the artifact through sandbox certification before activation, with the consumer's authority signing the certification.

At inference, the admissibility gate routes the request across the active certified artifacts according to composite admissibility against the consumer's policy. The same evaluator that gates execution gates skill activation, unifying the architecture rather than splitting it across separate skill and execution layers.

What This Enables for the Agent Skill Economy

Decentralized skill distribution becomes structurally tractable: artifacts can flow through the governed mesh, consumers certify on their own authority, dependencies cascade structurally on revocation. The platform-operator gating that current marketplaces depend on becomes optional rather than required.

Cross-model artifact portability follows. An artifact compatible across base models migrates as deployments shift between vendors. The lock-in pattern of current platforms — where artifacts are vendor-locked — gives way to portable runtime adaptation. The patent positions the primitive that the agent skill economy is building toward.