Nick Clark · Published July 30, 2026 · PDF

# When the Agent Rents Everything It Depends On

A regulatory documentation lead loses two years of accumulated working judgment on an ordinary Tuesday, not because a file was deleted, but because that judgment lived nowhere except a hosted endpoint she does not control and a session context that does not survive the request carrying it. The documents remain. The record of which suggestions she accepted, which she rewrote, and which she threw away does not, and it is not something she can reconstruct by re-uploading anything. U.S. Provisional Application No. 64/070,239 describes an architecture in which the persistent thing is the agent and the models are subordinate assets that can be replaced underneath it.

---

## The Tuesday the Endpoint Changed Underneath Her

A regulatory documentation lead at a mid-size medical device manufacturer opens her drafting environment on an ordinary Tuesday, four weeks into a submission cycle. For about two years she has worked through an assistant that reaches a hosted inference endpoint over the network. She has corrected that assistant more times than she could count: the way it phrases a risk statement, the order it puts a device description in, the words it is not allowed to use in a section that a reviewer reads literally.

On this Tuesday the endpoint reports a version string she has not seen before. Her first paragraph comes back in a voice that is not the one she spent two years shaping. The sentence structure is fine. The conventions are gone.

The specific thing she cannot do is show anyone, including herself, what changed. In her setup, everything the assistant appeared to know about her house style rode along inside each request as supplied context and did not survive the request that carried it. There is no artifact on her machine that holds the accumulated shape of her corrections. She cannot roll the assistant back, because in her deployment there is nothing local to roll back to. She cannot diff the old behavior against the new one, because the old behavior was never a thing she held.

She reopens a section she signed off on in March and does not know which endpoint version produced the sentence she approved.

## What She Cannot Reconstruct

Her documents are intact. They sit in her firm's document management system, versioned, with her name on them. That is not what she lost.

What she lost is the record of her own judgment applied to a machine's output: which suggestions she took verbatim, which she rewrote and how, which she discarded outright, and which ones a downstream reviewer sent back. Across two years and several thousand interactions, that record is the thing that made her assistant worth using. In her setup it existed only as a temporary condition of a session, and each session ended.

She can re-supply the documents. She cannot re-supply the decisions. She cannot re-derive an accept-or-revise judgment from a finished paragraph months later, because the paragraph in front of her does not say whether it arrived that way or arrived wrong

and got fixed. The signal that carried her working conventions was in the difference between what was proposed and what she shipped, and in her deployment that difference was never written down anywhere she controls.

The compliance consequence lands in the same place. When her quality auditor asks which model version contributed to which passage in a submission dated last October, she has an answer only for the parts a human touched. For her purposes, an answer she cannot evidence is not an answer.

She will rebuild. It will take her the better part of another two years, against a hosted endpoint that may change again inside that window, and the rebuilt version will not be the same one, because she is a different drafter now than she was when she started correcting it.

## Why Her Setup Puts the Memory in the Replaceable Part

Her assistant is assembled out of things she rents. The inference endpoint she calls belongs to a vendor and is versioned on that vendor's schedule. The retrieval index over her department's documents is rebuilt by a service she does not operate. The orchestration layer that strings the two together treats each of her work sessions as a transient dispatch, and treats her assistant as a configuration rather than as a thing with a history.

That arrangement puts every part she depends on in the category of components that can be swapped, and it puts the accumulated memory inside those same components. When the swap happened on Tuesday, her accumulated memory went with it. In her deployment, continuity is a property of a vendor's version number.

The shape of the difficulty for her is not that a model changed. Models will keep changing under her, and she has no objection to a better one. The difficulty is that in her configuration there is no layer that outlives the change. Were her assistant's

operating history held in something she owned, a model version change would read as an inventory event: this endpoint replaced that one, on this date, under this authorization, with the behavioral record intact on her side of the line. As her deployment is configured today, the same event reads as amnesia with no incident report.

She also has no way, in her present setup, to keep her professional drafting separate from the two other kinds of work she does on the same machine. One assistant, one undifferentiated context, one place where everything she types goes. When she is drafting a submission section, that is a governance problem she cannot express, because the system she uses has no place to express it.

## What the Filed Architecture Does

U.S. Provisional Application No. 64/070,239 discloses an agent-resident execution substrate in which the ordering she is missing is inverted. In one or more embodiments, the semantic agent (100) is the persistent entity on the computing device, and inference endpoints are managed assets subordinate to it.

As described in the filing, the agent comprises a persistent identity field (102), a cognitive state field (104), a lineage field (106), and a governance policy field (108). Managed inference endpoints (112a), (112b), (112c) are registered in a managed inference tool registry (110) local to the device, each comprising a model artifact, an interface specification, and an associated governance scope. An agent-to-tool dispatcher (120) routes inference requests to registered endpoints. A tool lifecycle controller (130) performs installation, retraining, replacement, archival, and removal, with each operation evaluated against the governance policy field and recorded in the lineage field.

The filing describes the lineage field as an append-only sequence of records. For each inference request dispatched through the dispatcher, the described lineage controller appends a record comprising an input descriptor, an endpoint identifier, an output descriptor, a timestamp, and a downstream-outcome reference. That downstream-outcome reference identifies subsequent events through which the output was evaluated, including user acceptance, user revision, downstream execution success, downstream execution failure, and integrity-signal feedback from coupled cognitive domain fields. The filing describes the lineage field as structured for append-only mutation under continuity proofs, such that a prior record cannot be modified or deleted without producing a detectable continuity break, with integrity enforceable through cryptographic chaining of each record to its predecessor.

The lifecycle of an endpoint is described in the filing as a state machine spanning an uninstalled state (200), an installed-inactive state (210), an active state (220), a retraining state (230), an updated-active state (240), an archived state (250), and a removed state (260). In an embodiment, retraining and substitution are performed in a staging area distinct from the active registry, and the updated artifact is promoted only upon successful completion and successful policy validation. Where policy validation of the updated artifact fails, the filing describes the substitution being rolled back and the failure being recorded in the lineage field together with its cause. A removed endpoint is deregistered while its lineage of prior operations is preserved in the agent's lineage field.

For the accumulated-conventions problem specifically, the filing describes a personal corpus model whose parameters are fine-tuned against a training corpus derived from artifacts authored, curated, or designated by the user under the user's governance policy field, configured to perform inference reflective of the user's terminology and structural conventions without retrieval over the body of artifacts at inference time. The closed loop runs from an authored artifact (400) through a corpus assembly module (410) that selects artifacts admissible under the corpus policy object, a fine-tuning

module (420) applying a parameter-efficient operation within the device's local compute envelope, and a governed substitution module (430) promoting the updated personal corpus model (440) into the registry.

In an embodiment described in the filing, under the continuity guarantee, the agent's identity field, cognitive state field, and lineage field are not modified, reset, or interrupted by lifecycle operations on subordinate components or by substrate-runtime update operations, except by appending to the lineage field under the continuity proof. The filing further describes named scope records within the cognitive state field, each with a scope identifier, a scope-local corpus policy object, a scope-local tool subset reference, and a scope-local lineage partition, permitting several independent contexts under one persistent agent identity.

## Where the Disclosed Architecture Stops for Her

It does not give her back the two years. The filing describes a provisioning sequence that initializes an empty lineage field with a genesis lineage record; a lineage that begins at provisioning begins there. Her prior interaction history is outside anything the disclosed architecture reaches.

It does not decide, on her behalf, which of her drafts belong in a training corpus. The corpus assembly module described in the filing selects artifacts admissible under a corpus policy object and applies declared redaction or anonymization rules. Someone in her organization still has to author that policy, and a policy she writes carelessly produces a corpus that reflects the carelessness.

It does not promise her a better draft. The filing describes governed assembly, fine-tuning, validation, and substitution; the quality of any particular fine-tuned artifact in her deployment depends on the corpus, the technique, and the compute available on her device.

It does not remove her dependence on remote inference where her local endpoints lack capability or capacity. The filing describes cloud-burst forwarding conditioned on an admissibility test comprising a capability test, a capacity test, a disclosure test, and a cost test, with forwarded payloads evaluated as off-device disclosure events under the applicable disclosure policy object. For her, that changes forwarding from something invisible into something conditioned and recorded. It does not make the remote endpoint hers.

And the privacy invariant described in the filing is conditioned on the disclosure policy object her organization declares. Where that policy authorizes disclosure, disclosure occurs and is recorded as a deterministic disclosure event; where it does not, the filing describes the transmission being denied by the substrate runtime and recorded as a denial event. The architecture gives her an enforcement point and an audit surface. It does not choose her policy for her.

## Disclosure Scope

This article is a technical description of subject matter disclosed in U.S. Provisional Application No. 64/070,239, titled "Agent-Resident Execution Substrate with Governed Inference Tool Registry and Lineage-Derived Personal Corpus Model Training." Reference numerals and mechanism names used above follow that filing. Nothing in this article characterizes the scope of any claim, and nothing here is an admission regarding the state of the art. The scenario described is illustrative and does not depict any actual person, organization, or product.

---

Persistent execution environment carried by the agent, not the host — identity, state, and lineage across power cycles, devices, and upgrades.

Provisional application

## PRIMARY TECHNICAL DISCLOSURE

- Agent-Resident Execution Substrate, Articles (/articles/agent-resident-execution-substrate)

## SECONDARY TECHNICAL

- Persistent Semantic Agent (/articles/agent-resident-execution-substrate/persistent-semantic-agent)
- Managed Inference Tool Registry (/articles/agent-resident-execution-substrate/managed-inference-tool-registry)
- Agent-to-Tool Dispatcher (/articles/agent-resident-execution-substrate/agent-to-tool-dispatcher)
- Lineage-Derived Training Signal (/articles/agent-resident-execution-substrate/lineage-derived-training-signal)
- Identity Preservation Across Upgrades (/articles/agent-resident-execution-substrate/identity-preservation-across-upgrades)
- Cognitive State-Conditioned Dispatch (/articles/agent-resident-execution-substrate/cognitive-state-conditioned-dispatch)
- Governed Tool Lifecycle (/articles/agent-resident-execution-substrate/governed-tool-lifecycle)
- Continuity-Proof Lineage (/articles/agent-resident-execution-substrate/continuity-proof-lineage)
- Substrate Runtime Continuity (/articles/agent-resident-execution-substrate/substrate-runtime-continuity)
- Personal Corpus Model Training (/articles/agent-resident-execution-substrate/personal-corpus-model-training)
- Heterogeneous Inference Endpoints (/articles/agent-resident-execution-substrate/heterogeneous-inference-endpoints)
- Atomic Lifecycle Substitution (/articles/agent-resident-execution-substrate/atomic-lifecycle-substitution)
- Integrity Signal Feedback (/articles/agent-resident-execution-substrate/integrity-signal-feedback)
- Hardware-Bound Identity (/articles/agent-resident-execution-substrate/hardware-bound-identity)
- Cognitive State Append-Only Invariant (/articles/agent-resident-execution-substrate/cognitive-state-append-only-invariant)
- Counterparty Identity Records (/articles/agent-resident-execution-substrate/counterparty-identity-records)

- Privacy Egress-Controlled Disclosure (/articles/agent-resident-execution-substrate/privacy-egress-controlled-disclosure)
- Federated Cross-Device Agent Identity (/articles/agent-resident-execution-substrate/federated-cross-device-agent-identity)

## APPLICATIONS · GENERAL

- Personal AI Agents That Survive Device Loss: One Continuous Identity and a Private Corpus Across Every Device (/articles/agent-resident-execution-substrate/personal-cross-device-agents)
- Enterprise Agent Fleets: Stable Agent Identity and Governed Tool Access Across Model Upgrades and Infrastructure Migration (/articles/agent-resident-execution-substrate/enterprise-agent-fleets)
- Audit-Grade Agent Identity for Regulated Finance and Healthcare: Continuity-Proof Lineage Across the Agent Lifecycle (/articles/agent-resident-execution-substrate/regulated-industry-agents)
- Edge and On-Device Agents: Hardware-Bound Identity Across Heterogeneous Inference Endpoints (/articles/agent-resident-execution-substrate/edge-and-on-device-agents)
- Agent-to-Agent Commerce With Counterparty Identity Records and Egress-Controlled Disclosure (/articles/agent-resident-execution-substrate/agent-to-agent-commerce)
- Governed Tool Lifecycles for Managed Inference-Provider Ecosystems: A Substrate Approach to Owning, Routing, and Retiring AI Tools (/articles/agent-resident-execution-substrate/managed-tool-ecosystems)
- Proving Unbroken Continuity in Long-Lived Autonomous Systems Across Substrate Migration and Atomic Model Substitution (/articles/agent-resident-execution-substrate/long-lived-autonomous-systems)
- Personal-Model Personalization: A User's Own Corpus-Internalized Model on the Agent-Resident Execution Substrate (/articles/agent-resident-execution-substrate/personal-model-personalization)
- On-Device Agent Identity for Robots and Autonomous Vehicles: An Auditable Substrate for Embodied Physical-World Agents (/articles/agent-resident-execution-substrate/embodied-physical-world-agents)
- **When the Agent Rents Everything It Depends On (/articles/agent-resident-execution-substrate/agent-resident-execution-substrate-stakes)**

## APPLICATIONS · SPECIFIC

- LangGraph Platform (LangChain) vs an agent-resident execution substrate: orchestration-graph state versus a portable, hardware-anchored agent runtime (/articles/agent-resident-execution-substrate/langgraph-platform)
- OpenAI AgentKit and the Assistants/Responses API vs agent-carried, hardware-anchored identity with governed tool lifecycle (/articles/agent-resident-execution-substrate/openai-agentkit)

- Microsoft Copilot Studio vs an agent-resident execution substrate: platform-hosted agent authoring versus portable, device-resident agent identity and continuity (/articles/agent-resident-execution-substrate/microsoft-copilot-studio)
- Google Vertex AI Agent Engine (managed runtime for deploying and scaling agents, with sessions/memory) vs an agent-carried, continuity-proofed identity substrate (/articles/agent-resident-execution-substrate/google-vertex-agent-engine)
- AWS Bedrock AgentCore (runtime, memory, identity, and gateway services for deploying agents at scale) vs an agent-resident execution substrate: where does the agent identity actually live? (/articles/agent-resident-execution-substrate/aws-bedrock-agentcore)
- Letta (formerly MemGPT) vs an append-only cognitive-state substrate: what a memory-management framework does not provide (/articles/agent-resident-execution-substrate/letta-memgpt)
- Cognition's Devin, an autonomous AI software-engineering agent vs a portable, continuity-proofed agent-resident runtime (/articles/agent-resident-execution-substrate/cognition-devin)
- Cloudflare Agents (Durable Objects) vs an agent-resident execution substrate: portable hardware-bound identity and continuity-proof lineage (/articles/agent-resident-execution-substrate/cloudflare-agents)
- Ollama alternative: from local model runner to a governed agent-resident substrate (/articles/agent-resident-execution-substrate/ollama)
- Apple Intelligence (on-device foundation models, Private Cloud Compute) vs a persistent agent-resident substrate: who owns identity, lineage, and the model? (/articles/agent-resident-execution-substrate/apple-intelligence)

## HOW-TO GUIDES

- How to Build a Personal AI That Runs on Your Own Device (/articles/guides/personal-ai-on-your-own-device)
- How to Build an Offline-First Personal AI That Syncs When Connected (/articles/guides/offline-first-personal-ai-that-syncs)
- How to Fine-Tune a Personal Model on Your Own Writing Without Sending It to the Cloud (/articles/guides/fine-tune-personal-model-without-cloud)
- How to Give an On-Device AI Agent Tools It Can Use Safely (/articles/guides/governed-tools-for-an-on-device-agent)

## TERMINOLOGY

- agent-resident execution substrate (/glossary#agent-resident-execution-substrate)
- continuity attestation (/glossary#continuity-attestation)
- continuity hash (/glossary#continuity-hash)

- lineage-derived training corpus (/glossary#lineage-derived-training-corpus)
- personal corpus model (/glossary#personal-corpus-model)

---

Agent-Resident Execution Substrate overview → (/agent-resident-execution-substrate)