Nick Clark · Published April 25, 2026 · PDF

Proving Unbroken Continuity in Long-Lived Autonomous Systems Across Substrate Migration and Atomic Model Substitution

A long-lived autonomous system (a robot, a vehicle controller, an industrial line agent) routinely outlives every model, accelerator, and firmware revision it started with, yet operators, auditors, and counterparties need to verify that the entity behaving today is provably the same entity that was commissioned, with no silent reset hiding in between. This application addresses that continuity-of-identity problem using the Agent-Resident Execution Substrate, disclosed in U.S. Provisional Application No. 64/070,239, whose continuity guarantee, append-only lineage with continuity-proof chaining, atomic governed lifecycle substitution, and dual-attested migration make unbroken identity a verifiable structural property rather than an operational assumption. It draws on sibling portfolio work in memory-native identity and authentication and in cryptographically enforced governance for autonomous agents.

What This Application Specifies

This application specifies how a long-lived autonomous system maintains, and can prove, a single unbroken identity across the entire chain of substitutions that accumulate over an operational lifetime measured in years. Such a system replaces its inference models many times, upgrades the software that runs it, swaps accelerators

and storage, and eventually migrates to wholly new hardware. The question that matters to an operator, a safety regulator, or a counterparty is not which model is loaded today but whether the behaving entity is provably continuous with the one that was commissioned, certified, and held accountable for everything it has done since.

Built on the Agent-Resident Execution Substrate disclosed in U.S. Provisional Application No. 64/070,239, the system is organized so that the persistent entity is a semantic agent that carries its own identity, cognitive state, lineage, and governance policy, while the inference models, the dispatcher, the lifecycle controller, and even the substrate runtime itself are subordinate managed components. The agent's identity is not dependent on any specific model artifact and is preserved across replacement, retraining, or removal of any subordinate model. Models loaded at any given moment merely constitute the agent's currently available inference capabilities; they are not the agent.

Three structures from the disclosure carry the continuity guarantee. The lineage field is an append-only sequence of records, each cryptographically chained to its predecessor, such that no prior record can be modified or deleted without producing a detectable continuity break. A continuity hash chained over that sequence is updated incrementally with each appended record and is verifiable by reproducing the chain from any prior reference value; a discrepancy between a computed hash and a stored reference is a detectable break that triggers escalation under the governance policy. And a continuity attestation cryptographically chains the agent's identity over the sequence of substrate-runtime versions, so any party verifying the agent can confirm its identity is unchanged across recorded runtime transitions.

Why It Matters

Autonomous systems that operate for years in the physical world face a problem that short-lived, stateless inference services never confront: their components churn faster than their accountability does. A model is retrained on a new corpus, a runtime is

patched against a vulnerability, an aging device is replaced. Each of these is, in conventional architectures, an opportunity for a silent reset, a re-provisioning that quietly severs the entity behaving after the change from the entity that was certified before it. When an incident is investigated, the operator cannot prove the system was continuous, and when continuity cannot be proven, neither can responsibility.

The disclosed substrate makes the system local to a single operation, with its accumulated lineage and personal corpus models held within the operation's governance boundary and disclosed off-device only under an explicit disclosure policy. That locality matters for long-lived deployments in regulated settings, but the deeper value is that continuity becomes auditable. Because every dispatch, every lifecycle operation, every runtime update, and every migration is recorded as a deterministic event in an append-only chain, the complete operational history is deterministically reconstructible, and any holder of the lineage can verify that the recorded history has not been altered. The entity does not merely claim to be continuous; it carries the proof.

How It Composes With the Domain

A long-lived autonomous deployment maps cleanly onto the substrate's primitives, with the domain supplying the deployment context and the disclosed mechanisms supplying the behavior.

Atomic model substitution is the most frequent continuity-relevant event. The disclosure performs retraining and substitution in a staging area distinct from the active tool registry, promoting an updated model artifact to the active state only upon successful completion and successful policy validation. If validation fails, the substitution is rolled back, the prior artifact remains active, and the failure is recorded in the lineage together with its cause. The tool lifecycle state machine makes this explicit: an endpoint advances from active to a retraining state, and from there either substitutes forward to an updated-active state or rolls back to active, never leaving the registry in an inconsistent intermediate. For an autonomous system that cannot

tolerate an undefined behavioral state mid-swap, this atomicity guarantee is the operative property, and the agent's identity, cognitive state, and lineage are preserved across the substitution regardless of which branch the state machine takes.

Substrate-runtime upgrade is the next continuity event. The disclosure treats an update to the runtime itself, the agent runtime, dispatcher, lifecycle controller, and governance and resource subsystems, as a governed operation distinct from model lifecycle operations, executed through a staged procedure that preserves operational state and rolls back to the prior version on failure. The persistent identity, cognitive state, and lineage are carried across the version transition by the continuity attestation, which chains the agent's identity over the runtime version sequence so that the post-upgrade entity is verifiably the pre-upgrade entity.

Hardware migration is the hardest case, and the disclosure handles it without sacrificing the continuity proof. The persistent identity field can be cryptographically bound to a hardware security element, a secure enclave, a trusted platform module, a hardware security module, or an embedded secure element, whose private key material is not extractable. A hardware-bound identity is not transferable to a device lacking the bound element except under a governed migration operation: the originating element attests that the migration is authorized, the destination element performs key derivation under a migration policy object, and the migration event is recorded in the lineage with attestations from both originating and destination devices. The companion agent migration lifecycle operation realizes this as an agent snapshot on the originating device, transfer of a snapshot artifact carrying a continuity attestation, and an agent restore on the destination that verifies the attestation before resuming. Migration is admissible only under a migration policy and may be denied if the destination fails declared hardware, governance, or trust criteria. The result is that a fleet device retired for hardware refresh hands its proven-continuous identity to its replacement, with both endpoints of the handoff cryptographically attested and permanently recorded.

For multi-unit deployments, the federation layer maintains a federated agent identity record that verifies, through cross-device attestations, that the agents on two or more devices correspond to a single identity, preserved across device additions, retirements, and hardware refresh within the population. Federation exchanges lineage records rather than model weights, so each unit retains local responsibility for its own artifacts while the population shares a coherent identity and outcome history.

What This Enables

The composition enables continuity-dependent capabilities that long-lived autonomous systems otherwise cannot offer. An operator can present a regulator or an insurer with a single reconstructible history spanning years and every component generation, in which model substitutions, runtime upgrades, and hardware migrations all appear as deterministic, attested events in one unbroken chain. A safety investigator examining an incident can verify that the system was continuous through the period under review, and can locate exactly which model version and policy version governed the behavior in question, because each dispatch records the governing policy version and each lifecycle event records its cause.

It enables hardware refresh without re-certification of identity: because migration is dual-attested and the destination must satisfy declared trust criteria, a fleet can rotate aging units into replacements while preserving each unit's accountable history. It enables governed succession, in which an agent transfers authority to a designated successor identity that records the originating agent as a recorded ancestor, and governed decommissioning, in which a unit is wound down with lineage preserved or destroyed under retention policy and a terminal attestation recorded, so device retirement and sale are themselves auditable events rather than silent disappearances. And it enables continuity to survive adversarial conditions: if the hardware security element fails attestation or is detected as compromised, the substrate can quarantine

agent execution, suspending dispatch and refusing governed lifecycle operations until integrity is re-established or the agent is explicitly migrated, so a tampered unit cannot continue to act under a borrowed identity.

Embodiments span the deployment configurations the disclosure enumerates, robotic devices, vehicular computing devices, embedded and industrial or operational-technology installations, each applying the same continuity primitives with deployment-specific resource budgets, foreground prioritization of perception-driven inference over background retraining, and counterparty records for the entities each unit encounters in service.

Boundary Conditions

This is a faithful application of the disclosed substrate to the long-lived-autonomy domain, not a separate invention, and the boundaries should be stated plainly. U.S. Provisional Application No. 64/070,239 is a provisional, early-stage filing; it establishes the architecture and its primitives, and a provisional does not by itself fix claim scope. The continuity guarantee is an architectural property the disclosure specifies through structural separation of the agent's persistent state from subordinate components and through cryptographic continuity proofs over the lineage; it is only as strong as the integrity of the hardware security element to which identity is bound and the soundness of the cryptographic chaining, and the disclosure itself anticipates compromise by specifying quarantine and re-attestation rather than assuming the element is inviolable. Several primitives this application relies on, hardware-rooted attestation, append-only chained logs, parameter-efficient fine-tuning, staged-update rollback, are well-established techniques and are prior art in their own right; the contribution lies in the architecture that composes them so that a single agent identity is preserved and provable across model substitution, runtime upgrade, and hardware migration, not in any one underlying primitive. Regulatory acceptance of a lineage-

based continuity proof as evidence in any particular jurisdiction or certification regime is a matter of external law and standards, addressed below as context rather than as a claim.

Disclosure Scope

The technology described here, the continuity guarantee, the append-only lineage with continuity-proof chaining, the continuity hash and continuity attestation, atomic staged model substitution with rollback, governed substrate-runtime update, hardware-anchored identity binding, dual-attested governed migration, and the agent snapshot, restore, migration, succession, and decommissioning lifecycle operations, is disclosed in U.S. Provisional Application No. 64/070,239. The long-lived-autonomy framing, including deployment scenarios in robotic, vehicular, embedded, and industrial settings, the operator and auditor and counterparty roles, and any reference to safety certification, incident investigation, or regulatory acceptance, is external context describing how the disclosed architecture can be applied; it is a faithful enabling implementation and is not itself a patent claim. References to standards bodies, regulators, or certification regimes describe the real-world domain in which such systems operate and do not represent endorsements or assertions of compliance. Sibling portfolio applications referenced for component primitives, including memory-native identity and authentication and cryptographically enforced governance for autonomous agents, are incorporated in the provisional by reference for their corresponding subject matter; this application does not claim their independent contributions.

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)**

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