Nick Clark · Published April 25, 2026 · PDF

Edge and On-Device Agents: Hardware-Bound Identity Across Heterogeneous Inference Endpoints

Edge and on-device AI agents face a structural problem: they run across a fragmented fleet of inference endpoints (a phone NPU, a laptop GPU, an embedded accelerator, an occasional cloud call) with no continuous identity, no portable history, and no governed boundary on what leaves the device. This application addresses that problem using the Agent-Resident Execution Substrate, disclosed in U.S. Provisional Application No. 64/070,239, which carries a persistent, hardware-bound agent identity and an append-only lineage across every endpoint and every device the agent touches. It draws directly on the substrate's sibling primitives for hardware-anchored identity binding, cognitive-state-conditioned dispatch, cloud-burst forwarding, and cross-device federation.

What This Application Specifies

Edge and on-device deployment is the domain where an AI agent must run on the user's own hardware (a phone, a tablet, a laptop, a wearable, a vehicle head unit, an embedded controller) rather than as a session inside a remote service. The Agent-Resident Execution Substrate, disclosed in U.S. Provisional Application No. 64/070,239,

supplies the architecture for that setting: a semantic agent that operates as the persistent execution substrate of the device and treats inference endpoints as managed, governed assets subordinate to the agent rather than as the agent itself.

Concretely, the substrate maintains four persistent fields that survive the entire lifetime of the device: a persistent identity field, a cognitive state field, an append-only lineage field, and a governance policy field. Around those fields sits a managed inference tool registry holding one or more endpoints, each with a model artifact, an interface specification, and a governance scope. An agent-to-tool dispatcher routes each inference request to one or more registered endpoints. The model artifacts available at any moment are simply the agent's current capabilities; the agent's identity does not depend on any one of them and is preserved when an endpoint is replaced, retrained, archived, or removed.

For the edge domain, the load-bearing primitive is hardware-anchored identity binding. The provisional specifies that the persistent identity field can be cryptographically bound to a hardware security element of the device (a secure enclave, a trusted platform module, a hardware security module, or an embedded secure element) through a key-derivation operation that incorporates hardware-rooted values whose private key material is not extractable from the element. That binding is verifiable at any later substrate event by re-deriving the binding value, and a hardware-bound identity is not transferable to a device lacking the bound element except under a governed migration operation. If the hardware 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.

Why It Matters

The edge is heterogeneous by nature. A single user's effective deployment spans multiple hardware tiers (mobile, tablet, laptop, desktop, head-mounted, wearable, vehicular, and household appliance tiers among them), each with different memory,

compute, power, thermal, and form-factor characteristics. Conventional on-device inference frameworks treat this as request routing: they load and unload models and route to local endpoints, but they hold no persistent identity or state beyond user configuration, accumulate no lineage of outcomes, and preserve no continuous behavioral entity across model replacement. Network inference services hold the opposite problem; they maintain no persistent representation of an individual's body of work, and each request's context window is discarded.

That gap matters for three reasons specific to edge deployment. First, identity portability: a user who moves from phone to laptop to vehicle wants one agent, not three disconnected sessions. Second, the privacy boundary: edge deployment is often chosen precisely so that personal data stays on the device, which only holds if there is an enforced rule about what may cross the device boundary. Third, capability fragmentation: no single edge device can host every model a user needs, so the agent must compose smaller specialized endpoints and occasionally reach beyond the device, without losing the thread of who it is or what it has done.

The substrate answers each of these with disclosed mechanism rather than configuration. The hardware-bound identity field and the cross-device federation layer give portability. The privacy invariant and disclosure policy give an enforced boundary. The managed tool registry and cloud-burst forwarding subsystem give governed capability composition. None of these is a deployment convenience layered on top; each is part of the substrate the provisional describes.

How It Composes With the Domain

Start with a single edge device. The agent is provisioned at first power-on: the identity field is initialized with hardware-anchored binding where supported, the cognitive state field is seeded, an empty lineage field opens with a genesis record, an initial set of managed inference endpoints is installed under an installation policy, and a user policy object is derived from the user's authorization. From that point the dispatcher routes

each request by evaluating the input modality and task category against per-endpoint capability declarations (which enumerate admissible modalities, task categories, a latency profile, a resource envelope, and a capability confidence value), then applying thresholds drawn from the cognitive state field and selecting among admissible candidates by historical outcome quality recorded in the lineage.

Heterogeneous endpoints are first-class here. Multiple endpoints of distinct types and sizes can be co-resident: a general-purpose language model, task-specific fine-tuned models, a speech recognition model behind a voice input adapter, an embedding model, and one or more personal corpus models whose weights internalize the user's own accumulated work. The registry also supports adapter-based variants in which a single base artifact is shared and per-endpoint adapter weights specialize it, reducing the storage footprint that matters acutely on a constrained device. On an embedded or wearable tier, the registry simply holds a smaller set of more aggressively quantized endpoints, with retraining schedules and resource budgets configured for the reduced envelope.

The resource governance subsystem is what makes this survivable on real edge hardware. It enforces per-tool budgets for memory, storage, inference compute, retraining compute, network bandwidth, and power, and it can quiesce endpoints in response to thermal limits, battery pressure, or memory pressure, recording each decision in the lineage. Foreground inference is prioritized over background retraining and ingestion, and retraining is scheduled into idle or power-surplus windows. A vehicular or robotic deployment configures these budgets around thermal, power, real-time-response, and safety-critical constraints, prioritizing perception-driven inference over background work.

When local capability or capacity runs out, the cloud-burst forwarding subsystem can selectively forward a request to a remote endpoint, but only after an admissibility test: a capability test (does any local endpoint satisfy the request), a capacity test (can local compute meet the latency budget), a disclosure test (are the input artifacts admissible

for off-device disclosure), and a cost test (is projected remote cost within budget). A deferred mode queues forwarding for the next connectivity event and returns partial or surrogate responses while offline; a confidential-execution mode encrypts the payload under keys held only by the remote endpoint's trusted execution environment. Every forwarded payload is treated as an off-device disclosure event under the privacy invariant.

That privacy invariant is the boundary the edge domain depends on. Lineage records, model artifacts, training corpora, personal corpus model parameters, and counterparty identity records are not transmitted off the device except under an explicit disclosure policy object that names a recipient, a permitted scope, an authorization attestation, a retention requirement, and a revocation mechanism. Enforcement options include a substrate-runtime egress filter that inspects outbound traffic, per-component isolation so subordinate components cannot transmit on their own, release of transmission keys only after signed disclosure-policy preconditions are met, and hardware-anchored attestation that the runtime has not been tampered with. The invariant holds regardless of connectivity, and on-device inference is never itself a disclosure.

Across devices, federation unifies the fleet. Each device keeps its own agent, tool registry, and lineage store; the federation layer exchanges lineage records (not model weights) under a federation policy, so one device can incorporate outcome signals observed on another into its routing and its training corpus assembly. A federated agent identity record verifies through cross-device attestations that the agents correspond to a single user, so requests, lineage, scope changes, and counterparty encounters across the user's devices are treated as one agent identity, preserved across device additions, retirements, and hardware refresh. When a user replaces a device, transfer-provisioning and the governed agent migration operation (snapshot on the origin, attested transfer, restore on the destination, with attestations from both hardware elements recorded in lineage) carry the identity, lineage, cognitive state, policies, and personal corpus model to the new hardware.

What This Enables

For an edge deployment, the substrate makes a set of things buildable that on-device frameworks alone do not. A single user-owned agent can follow its user across phone, laptop, wearable, and vehicle as one continuous identity, carrying its accumulated history and its personal corpus model rather than resetting on each device. An on-device writing or coding assistant can run a personal corpus model that reflects the user's own terminology and conventions without sending that corpus anywhere, because the model's behavior lives in its weights and the privacy invariant fences the device boundary.

A constrained device can compose several small specialized endpoints into a multi-stage pipeline (classify, then a task-specific endpoint, then a verification endpoint) and reconfigure that pipeline through registry updates without touching the endpoints themselves. The same device can reach a remote endpoint only when the four-part admissibility test passes, giving a defensible answer to the question of exactly when and what data left the device, because every forwarding event is an auditable disclosure record. A household or family deployment can federate several devices under one policy while keeping model artifacts local, and a regulator or the user can audit the complete off-device disclosure history from the lineage on any participating device. A compromised or stolen device can be quarantined at the hardware-attestation layer, halting agent execution until integrity is restored or the agent is migrated under policy.

Boundary Conditions

This is a general-application article describing how a disclosed architecture maps onto the edge and on-device domain; it is not a benchmark report and asserts no latency, throughput, model-size, or accuracy figures, none of which appear in the provisional. The home invention is disclosed in a U.S. provisional application, which is an early-stage filing; the scope that ultimately issues, if any, is determined through prosecution, and nothing here should be read as a granted claim.

The substrate relies on capabilities that are external to it and are themselves prior art in the general sense: hardware security elements such as secure enclaves and trusted platform modules, parameter-efficient fine-tuning techniques such as low-rank adaptation, on-device model execution, and standard network transport. The provisional builds governance and continuity structure on top of these; it does not claim the underlying cryptographic hardware or the base fine-tuning methods. Several substrate primitives are described as implementable through incorporated sibling applications or through any equivalent mechanism, so a given deployment's specifics may differ. Hardware-anchored binding, federation, cloud-burst forwarding, and the privacy invariant are each described in the provisional as embodiments ("in an embodiment"), meaning they are disclosed options rather than mandatory features of every configuration. Domain and regulatory framing in this article (audit expectations, data-residency motivations, fleet composition) is context for why the architecture fits the edge, not a representation about any specific product, vendor, or legal requirement.

Disclosure Scope

The technology described here (the agent-resident execution substrate, its hardware-anchored identity binding, its cognitive-state-conditioned dispatch across heterogeneous managed inference endpoints, its resource governance and quiescence behavior, its cloud-burst forwarding admissibility test, its privacy invariant and governed off-device disclosure, and its cross-device federation and governed migration) is disclosed in U.S. Provisional Application No. 64/070,239. Every statement in this article about what the substrate does traces to that disclosure. The edge and on-device framing, including the description of device tiers, fleet fragmentation, data-residency motivations, and audit expectations, is external domain context offered to show a faithful enabling implementation; it is not part of the patent claims and should not be read as one. Mechanisms attributed above to sibling filings (for example, hardware-

rooted and continuity-based identity primitives, cryptographically enforced governance policy, and decentralized federation transport) are incorporated by reference in the provisional and are not independently claimed here.

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

All 40 steps → (/inventive-steps)

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)