

# Cloudflare Agents (Durable Objects) vs an agent-resident execution substrate: portable hardware-bound identity and continuity-proof lineage

Cloudflare Agents, built on Durable Objects, gives each stateful agent its own single-writer micro-server with an embedded SQLite database that persists across restarts, deploys, and hibernation on Cloudflare's edge. That model solves durable per-agent state elegantly, but the agent's identity and its state are keyed to a Durable Object instance inside one provider's runtime. This article compares that approach to the Agent-Resident Execution Substrate, disclosed in U.S. Provisional Application No. 64/070,239, which roots a single agent identity in a hardware-anchored, continuity-proof lineage that survives across substrates rather than living inside a provider-managed object.

---

## What Cloudflare Agents (built on Durable Objects) Does

Cloudflare Agents is a platform for building and running stateful AI agents on Cloudflare's global network. Its central design choice is to make each agent a Durable Object: the Agent class is built on Cloudflare's Durable Object base class, so every agent instance is a single, addressable, stateful micro-server with its own embedded SQLite database, its own WebSocket connections, and its own scheduler. Cloudflare positions

Durable Objects as single-threaded and cooperatively multitasked, which means each instance behaves as a single-writer actor whose state cannot be corrupted by concurrent requests.

The result is a genuinely strong story for durable per-agent state. State persists automatically across requests, restarts, deploys, and hibernation cycles; developers declare the shape of their state and the runtime handles loading, saving, and broadcasting changes. Each instance is inexpensive while idle, and the SQLite-backed storage supports tables and offers point-in-time recovery over a recent window. Agents can reach each other through Durable Object routing, and can be driven over RPC, WebSockets, email, and other event channels, which makes composing specialized agents into cooperating systems feel like assembling microservices. For teams that want stateful agents without operating their own state layer, this is a clean, well-engineered model, and it is fair to credit its concurrency safety and operational simplicity.

## **The Architectural Axis**

The axis this comparison addresses is not durability of state in the abstract, but where an agent's identity and its accumulated history physically reside, and what that binding does when the substrate changes. In the Cloudflare model, an agent is a Durable Object, and its state lives in that object's storage inside Cloudflare's runtime. Identity is effectively the object's addressable name, and continuity is the continuity of that object and its storage as managed by the provider.

That is a coherent and useful design. It also frames a structural boundary: the agent is defined by, and coupled to, the object that hosts it. The state travels wherever the provider's runtime places that object, but it is the provider's runtime that owns the persistence, the identity handle, and the recovery semantics. This is a difference in architecture, not a defect. Many applications never need identity to exist independently

of the hosting object, and for those applications the Durable Object binding is exactly the right abstraction. The disclosed substrate simply draws the identity boundary in a different place.

## **How the Disclosed Approach Differs**

The Agent-Resident Execution Substrate makes the agent, rather than the inference endpoint or the hosting object, the persistent thing. As described in U.S. Provisional Application No. 64/070,239, the semantic agent is instantiated as a persistent computational entity carrying four persistent fields: a persistent identity field, a cognitive state field, an append-only lineage field, and a governance policy field. Inference endpoints are managed assets subordinate to the agent inside a tool registry; they can be installed, retrained, replaced, archived, or removed as governed lifecycle operations. Crucially, the spec provides a continuity guarantee under which the agent's identity, cognitive state, and lineage are not modified by any lifecycle operation on a subordinate component, and are preserved even across replacement of every endpoint and across update of the substrate runtime itself.

Two mechanisms in the disclosure sit directly on the identity-location axis. First, the identity field may be cryptographically bound to a hardware security element of the device, such as a secure enclave or trusted platform module, through a key-derivation operation that incorporates hardware-attested values whose private key material is not extractable from the element. A hardware-bound identity is not transferable to a device lacking that element except under a governed migration operation, which requires attestation from both the originating and destination hardware elements and records the migration in the lineage. Second, the lineage field is append-only under continuity proofs: each lineage record carries a cryptographic reference to its predecessor, and the identity field can include a continuity hash chained over the sequence of substrate events, so any party can verify that the recorded history has not been altered and that the agent's identity is unchanged across runtime version transitions.

The consequence on this axis is that the agent's identity does not depend on any particular hosting object, model artifact, or provider runtime. The spec describes agent-level snapshot, restore, and migration operations, and a multi-device federation in which two or more substrate devices coordinate by exchanging lineage records rather than model weights, maintaining a federated agent identity record verified through cross-device attestations so that events across the devices are treated as originating from a single agent identity. The disclosure is explicit that this differs structurally from cloud-account-based device association, which links devices to an account identifier but does not unify a persistent, attestation-verified agent identity across them. Where Cloudflare anchors continuity to a Durable Object inside one runtime, the disclosed substrate anchors continuity to a portable, hardware-rooted identity plus a verifiable lineage chain that can move across substrates under governed migration.

## **Where They Fit Together**

These are not mutually exclusive, and in many real deployments they compose. Cloudflare Agents is a strong fit when the workload is naturally cloud-resident, when horizontal scale across many independent agent instances matters, and when a team wants the provider to own state durability, concurrency safety, and recovery. The disclosed substrate is oriented toward the opposite pull: keeping the agent's identity, lineage, and personal corpus model on the user's own devices under a privacy invariant, where lineage records and model parameters are not transmitted off-device except under an explicit disclosure policy.

The two can meet through the substrate's cloud-burst forwarding subsystem, which the spec describes as selectively forwarding inference requests to a remote network-accessible endpoint when local endpoints lack capability or capacity, under a cloud-burst policy that governs admissible destinations, disclosure scope, encryption, and cost limits, with each forwarding event recorded as a disclosure event in the lineage. In such a composition, a locally identity-bound agent could burst heavy inference to

Cloudflare-hosted capacity while retaining ownership of its identity and history locally. The choice is less compete-versus-compose and more about which layer holds the durable identity: the provider's object, or the user's hardware-rooted agent.

## **Boundary Conditions**

The disclosed substrate is at an early stage. U.S. Provisional Application No. 64/070,239 is a provisional filing; it describes an architecture and a family of embodiments rather than a shipping product, and the strength of any resulting claims will depend on prosecution and on how the embodiments are reduced to practice. Several of its guarantees rest on capabilities that carry real-world limits: hardware-anchored identity depends on the availability and integrity of a secure element, and the spec itself contemplates quarantine of agent execution if hardware attestation fails; on-device personal corpus models are bounded by local memory, storage, and compute, so the substrate trades the effectively elastic scale of a cloud runtime for locality and control. Continuity proofs establish that a lineage has not been altered, but they do not by themselves establish that recorded outcomes were correct.

On the Cloudflare side, the characterization here is limited to genuine, widely-documented architecture-level facts: agents are Durable Objects, state is single-writer and SQLite-backed, and persistence survives restarts and hibernation. Nothing in this article should be read as asserting a defect in Cloudflare Agents. Coupling agent identity to a provider-managed object is a deliberate and reasonable design that many workloads benefit from; it is simply a different placement of the identity boundary than the one this disclosure explores. Specific product details, limits, and pricing on the Cloudflare side evolve, and readers should confirm current behavior against Cloudflare's own documentation.

## Disclosure Scope

This article describes subject matter disclosed in U.S. Provisional Application No. 64/070,239 and compares it, on a single architectural axis, to Cloudflare Agents built on Durable Objects. Statements about our technology trace to that disclosure; statements about Cloudflare Agents are offered as external context drawn from publicly available material and are not claims of the filing. Nothing here asserts that Cloudflare Agents or Durable Objects is deficient, and the comparison is not a representation of Cloudflare's roadmap, internals beyond what Cloudflare documents, or commercial terms. The framing of markets, competitors, and use cases is context only; the scope of any patent rights is defined by the claims as they issue, not by this article.

---

### **Agent-Resident Execution**

[All 40 steps → \(/inventive-steps\)](#)

### **Substrate** ([/agent-resident-execution-substrate](#))

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 \(/articles/agent-resident-execution-substrate/cognitive-state-conditioned-dispatch\)](/articles/agent-resident-execution-substrate/cognitive-state-conditioned-dispatch).
- [Governed Tool Lifecycle \(/articles/agent-resident-execution-substrate/governed-tool-lifecycle\)](/articles/agent-resident-execution-substrate/governed-tool-lifecycle).
- [Continuity-Proof Lineage \(/articles/agent-resident-execution-substrate/continuity-proof-lineage\)](/articles/agent-resident-execution-substrate/continuity-proof-lineage).
- [Substrate Runtime Continuity \(/articles/agent-resident-execution-substrate/substrate-runtime-continuity\)](/articles/agent-resident-execution-substrate/substrate-runtime-continuity).
- [Personal Corpus Model Training \(/articles/agent-resident-execution-substrate/personal-corpus-model-training\)](/articles/agent-resident-execution-substrate/personal-corpus-model-training).
- [Heterogeneous Inference Endpoints \(/articles/agent-resident-execution-substrate/heterogeneous-inference-endpoints\)](/articles/agent-resident-execution-substrate/heterogeneous-inference-endpoints).
- [Atomic Lifecycle Substitution \(/articles/agent-resident-execution-substrate/atomic-lifecycle-substitution\)](/articles/agent-resident-execution-substrate/atomic-lifecycle-substitution).
- [Integrity Signal Feedback \(/articles/agent-resident-execution-substrate/integrity-signal-feedback\)](/articles/agent-resident-execution-substrate/integrity-signal-feedback).
- [Hardware-Bound Identity \(/articles/agent-resident-execution-substrate/hardware-bound-identity\)](/articles/agent-resident-execution-substrate/hardware-bound-identity).
- [Cognitive State Append-Only Invariant \(/articles/agent-resident-execution-substrate/cognitive-state-append-only-invariant\)](/articles/agent-resident-execution-substrate/cognitive-state-append-only-invariant).
- [Counterparty Identity Records \(/articles/agent-resident-execution-substrate/counterparty-identity-records\)](/articles/agent-resident-execution-substrate/counterparty-identity-records).
- [Privacy Egress-Controlled Disclosure \(/articles/agent-resident-execution-substrate/privacy-egress-controlled-disclosure\)](/articles/agent-resident-execution-substrate/privacy-egress-controlled-disclosure).
- [Federated Cross-Device Agent Identity \(/articles/agent-resident-execution-substrate/federated-cross-device-agent-identity\)](/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 \(/articles/agent-resident-execution-substrate/personal-cross-device-agents\)](/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 \(/articles/agent-resident-execution-substrate/enterprise-agent-fleets\)](/articles/agent-resident-execution-substrate/enterprise-agent-fleets).
- [Audit-Grade Agent Identity for Regulated Finance and Healthcare: Continuity-Proof Lineage Across the Agent Lifecycle \(/articles/agent-resident-execution-substrate/regulated-industry-agents\)](/articles/agent-resident-execution-substrate/regulated-industry-agents).
- [Edge and On-Device Agents: Hardware-Bound Identity Across Heterogeneous Inference Endpoints \(/articles/agent-resident-execution-substrate/edge-and-on-device-agents\)](/articles/agent-resident-execution-substrate/edge-and-on-device-agents).
- [Agent-to-Agent Commerce With Counterparty Identity Records and Egress-Controlled Disclosure \(/articles/agent-resident-execution-substrate/agent-to-agent-commerce\)](/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 \(/articles/agent-resident-execution-substrate/managed-to-ol-ecosystems\)](/articles/agent-resident-execution-substrate/managed-to-ol-ecosystems)
- [Proving Unbroken Continuity in Long-Lived Autonomous Systems Across Substrate Migration and Atomic Model Substitution \(/articles/agent-resident-execution-substrate/long-lived-autonomous-systems\)](/articles/agent-resident-execution-substrate/long-lived-autonomous-systems)

## APPLICATIONS · SPECIFIC

- [LangGraph Platform \(LangChain\) vs an agent-resident execution substrate: orchestration-graph state versus a portable, hardware-anchored agent runtime \(/articles/agent-resident-execution-substrate/langgraph-platform\)](/articles/agent-resident-execution-substrate/langgraph-platform)
- [OpenAI AgentKit and the Assistants/Responses API vs agent-carried, hardware-anchored identity with governed tool lifecycle \(/articles/agent-resident-execution-substrate/openai-agentkit\)](/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 \(/articles/agent-resident-execution-substrate/microsoft-copilot-studio\)](/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 \(/articles/agent-resident-execution-substrate/google-vertex-agent-engine\)](/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? \(/articles/agent-resident-execution-substrate/aws-bedrock-agentcore\)](/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 \(/articles/agent-resident-execution-substrate/letta-memgpt\)](/articles/agent-resident-execution-substrate/letta-memgpt)
- [Cognition's Devin, an autonomous AI software-engineering agent vs a portable, continuity-proofed agent-resident runtime \(/articles/agent-resident-execution-substrate/cognition-devin\)](/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 \(/articles/agent-resident-execution-substrate/cloudflare-agents\)\*\*](/articles/agent-resident-execution-substrate/cloudflare-agents)

---

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