Nick Clark · Published June 30, 2026 · PDF

Agent-to-Agent Commerce With Counterparty Identity Records and Egress-Controlled Disclosure

When one autonomous agent transacts or collaborates with another, each side needs to know who it is dealing with, what it may reveal, and whether the exchange is auditable, problems that ephemeral, stateless agent sessions cannot solve. This application is built on the Agent-Resident Execution Substrate, disclosed in U.S. Provisional Application No. 64/070,239, which gives each agent a persistent hardware-bound identity, substrate-resident counterparty identity records under governed persistence promotion, and a privacy invariant that blocks off-device disclosure except under an explicit, recorded disclosure policy. It draws on the substrate's federated cross-device agent identity and its publisher attribution and compensation routing for marketplace transactions.

What This Application Specifies

This application describes agent-to-agent commerce and collaboration: scenarios in which a semantic agent acting for one party transacts, negotiates, or shares work with an agent acting for another party. The buyer's agent and the seller's agent settle terms. A household device's agent exchanges a calendar slot with a service provider's agent. An enterprise procurement agent solicits and compares offers from supplier agents. A robot's agent and a visiting technician's agent agree on what each is permitted to see. In every case two persistent computational entities, each carrying its own identity and policy, must establish who the other is, decide what may be disclosed, and leave an auditable record.

The Agent-Resident Execution Substrate (U.S. Provisional Application No. 64/070,239) supplies the primitives this requires. Each agent on the substrate is a persistent identity-bearing entity, not an ephemeral session. Its persistent identity field can be cryptographically bound to a hardware security element (a secure enclave, trusted platform module, hardware security module, or embedded secure element) so that the counterparty is dealing with a verifiable, non-transferable identity rather than a disposable account. The substrate maintains counterparty identity records for the entities its agent encounters, peer agents on other devices, individual humans, devices, and organizational entities, each record carrying a counterparty identifier, a counterparty scope object stating what interactions are admissible, an encounter history in the append-only lineage field, a persistence designation, and attestation references that verify the counterparty's claimed identity.

Two further substrate primitives make the commerce safe to conduct. First, the privacy invariant: lineage records, model artifacts, training corpora, personal corpus model parameters, scope-local context, 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. Second, every off-device disclosure, every inter-agent encounter that transmits state, is recorded in the lineage field as a deterministic disclosure event, producing a verifiable trail of exactly what left the device and under whose authority.

Why It Matters

The market problem is that the natural unit of commerce among autonomous agents is identity and disclosure, and the prevailing agent designs have neither. As the background of the disclosure observes, tool-using agent frameworks define agents as ephemeral sessions or stateless dispatch functions; they do not persist across sessions, do not own the assets they invoke, and do not accumulate a verifiable history. An agent with no durable identity cannot be a counterparty in any meaningful sense. The other side has nothing to attest to, nothing to hold accountable across interactions, and no way to build the kind of standing relationship that repeat commerce depends on.

The disclosure problem is the mirror image. An agent negotiating on a user's behalf holds sensitive material, the user's accumulated body of work internalized in a personal corpus model, scope-local context, the encounter histories of other counterparties. Conventional cloud-mediated agents routinely ship context off-device by default. For a buyer's agent to safely reveal a budget ceiling to a seller's agent, or for a clinic's device to share a referral with a partner clinic's device, disclosure must be the deliberate, bounded, auditable exception rather than the silent default. The substrate's privacy invariant inverts the default: nothing leaves unless a disclosure policy says it may, and the egress is logged.

How It Composes With the Domain

A transaction or collaboration between two substrate agents composes from primitives the disclosure already specifies.

Encounter and attestation. The substrate's inter-agent encounter mechanisms, peer agent discovery, mutual identity attestation, and information exchange under a negotiated encounter scope, establish who each agent is. A counterparty identifier is selected at runtime by entity type: a dynamic agent hash, dynamic device hash, public-key-anchored identifier, or hardware-anchored identifier for peer agents and devices; a credential-anchored, pseudonymous, or enrollment-anchored identifier for human counterparties; an organizational policy object reference signed by an organizational identity authority for organizational entities. The substrate supports multiple identifier schemes concurrently. The encounter is recorded as an encounter event referencing the counterparty record it updates, capturing the negotiated scope, the information exchanged in summary form, and the duration.

Scoped disclosure during negotiation. The counterparty scope object on each record states the admissible categories of inference dispatch involving the counterparty, the admissible categories of lineage record disclosure to the counterparty, and the admissible categories of training corpus contribution from the counterparty. Negotiation thus proceeds inside a declared envelope: the buyer's agent may disclose certain lineage records to the seller's agent and no others, and each disclosure is evaluated against the disclosure policy and the counterparty scope, then logged. The privacy invariant's enforcement mechanisms, a substrate-runtime egress filter intercepting outbound traffic, per-component isolation, signed disclosure-policy preconditions checked before any transmission key is released, and hardware-anchored attestation that the runtime is untampered, make the envelope structural rather than advisory.

Relationship persistence. A first contact begins as an ephemeral counterparty record retained only for the active session. Under a promotion policy object, an ephemeral record is promoted to persistent status on explicit user authorization, on interaction frequency above a threshold, on accumulated interaction duration, or on policy-declared event categories. A counterparty an agent transacts with repeatedly accrues a persistent record that accumulates lineage across distinct interactions, which is precisely the substrate's analogue of a trading relationship and a reputation history. The encounter history feeds the agent's routing and corpus-assembly decisions just as any other lineage does.

Settlement and compensation. The substrate enumerates payment instruction issuance among its effector output adapters, so a settled transaction can drive a governed side effect under the governance policy field. For marketplace exchanges of inference capability itself, the substrate records publisher attribution metadata for each installed managed inference endpoint and runs a compensation routing subsystem that attributes dispatch, retraining, and consultation events to a publisher and computes obligations under a compensation policy object specifying rate schedules, payment routing destinations, and audit requirements. One agent consuming another party's tool, or a capability sourced from a peer substrate device as a tool source authority, therefore carries attribution and a compensation path.

Multi-device parties. A counterparty is often not a single device. The substrate's federation layer maintains a federated agent identity record that verifies through cross-device attestations that two or more federated agents correspond to a single user identity, so the agents are treated as one across the user's device population. A transacting party can thus present a coherent identity whether its agent is reached on a phone, a laptop, or a vehicle, and the federated identity persists across device additions, retirements, and hardware refresh.

What This Enables

Concrete deployments follow directly. A household substrate device negotiates a service appointment with a provider's agent, disclosing only the calendar window and address category its disclosure policy permits, and retains a persistent record of a provider it uses regularly. An enterprise procurement agent solicits offers from supplier agents under organizational policy, where counterparty records correspond to clients, vendors, and contractual partners and persistence promotion is governed organizationally; every offer disclosure and counter-offer is in the lineage for audit. An autonomous vehicle's agent and a charging-station agent attest identities and settle a session, the station appearing as a counterparty record with hardware-anchored identity. A robot operating among people grants a visiting technician's agent a narrow, time-bounded scope and revokes it at session end, with the encounter and its disclosure events logged. Agents can also trade capability: one agent consults another's specialized endpoint, with publisher attribution preserved and compensation routed under policy.

Because the substrate is local and the privacy invariant holds regardless of network connectivity, these exchanges work without routing the parties' private state through a central broker. Each side keeps its lineage, corpus models, and counterparty records within its own governance boundary, disclosing only under an explicit, revocable, recorded policy.

Boundary Conditions

The home invention is disclosed in a U.S. provisional application and is at an early, pre-examination stage; provisional disclosure establishes a priority position but is not an issued patent, and claim scope will be determined in prosecution. The substrate specifies the mechanisms by which counterparty identity is recorded, disclosure is gated and logged, and compensation is attributed and routed; it does not specify, and this article does not assert, any particular pricing model, settlement network, payment rail, or commercial protocol. The cryptographic, payment, and networking building blocks the substrate composes (hardware security elements, public-key attestation, signed policy objects, parameter-efficient fine-tuning) are established prior art used here as enabling components, not claimed as inventions of this disclosure. Some attestation, transport, content-identity, and compensation primitives are described as suppliable by separately identified applications incorporated by reference, and may be implemented through any equivalent primitives suitable for the deployment context. The market, regulatory, and trust-relationship framing in this article is external context describing where the technology applies, not a representation about legal enforceability, regulatory approval, or the conduct of any counterparty.

Disclosure Scope

The technology described here, the persistent hardware-bound agent identity, counterparty identity records with governed persistence promotion, the privacy invariant gating and recording off-device disclosure, federated cross-device agent identity, and publisher attribution and compensation routing, is disclosed in U.S. Provisional Application No. 64/070,239, "Agent-Resident Execution Substrate." Every statement in this article about what the substrate does (how it records counterparties, how it gates and logs disclosure, how it attributes and routes compensation, how it unifies identity across devices) traces to that disclosure. The agent-to-agent commerce and collaboration domain, the procurement, household-service, vehicular, robotic, and enterprise scenarios, and any reference to standards, regulators, or market practice, is external context illustrating a faithful enabling implementation, not part of the patent claims and not a representation about commercial or regulatory outcomes.

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)
- 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)

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)

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