Nick Clark · Published April 25, 2026 · PDF

How to Prove Regulatory Compliance Across an Entire Fleet of Agents

You operate dozens or thousands of autonomous agents or devices, and an auditor asks you to prove that every action taken across the whole fleet was authorized, in-policy, and reconstructable. This guide describes an architecture for making that provable uniformly, so any agent's every action carries its own evidence. The approach is disclosed in U.S. Provisional Application No. 64/049,409 and centers on the Five-Property Governance Chain inventive step. It is a design you build yourself, not a shipping library.

What You Are Building

You run a fleet: autonomous vehicles, warehouse robots, delivery drones, or software agents acting on their own authority in the physical or digital world. A regulator, insurer, or internal safety board wants a single, uniform answer to one question, asked of any unit at any moment: on what authority did this agent do that, and can you reconstruct the evidence it acted on?

The naive framing is "collect logs and hope they add up." What you actually need is stronger: every action any agent takes should carry, in the action's own record, proof of who was allowed to contribute the inputs, proof that the action passed the policy in

force, and a provenance link back to the observations that justified it. When that property holds for every unit identically, compliance stops being a per-agent forensics project and becomes a structural property of the fleet.

This guide describes an architecture for that property. It is drawn entirely from the disclosure in U.S. Provisional Application No. 64/049,409, which frames it around a composed governance chain of five properties enforced and recorded per action. You implement it; nothing here is a package you install.

Why the Obvious Approaches Fall Short

The common approaches all work, up to a point, and it is worth being precise about where the point is.

Centralized logging and dashboards. Each agent ships telemetry to a cloud collector; you query it later. This is real and useful, but the disclosure notes a structural gap: the telemetry from each device tends to be disconnected per-device data, so reconstructing a cross-device causal chain ("this actuation was based on that neighbor's observation") means stitching records after the fact, and the record depends on transient telemetry that may not have survived.

Signed messages and PKI. Public-key infrastructure and credential-management systems authenticate messages and are genuinely good at answering "was this message from who it claims." The disclosure's observation is narrower and fair: a plain signed-message scheme carries a binary valid-or-invalid attribute consumed by a simple authentication check. It tells you the message is authentic; it does not, by itself, carry hierarchical trust semantics (how much this source's word should weigh) or compose one message's provenance with another's into a cross-source record.

Per-agent policy engines. Each agent enforces rules locally. This governs the single agent but does not, on its own, make the *decision* auditable back to its inputs across agents, nor guarantee that two agents in different domains enforce the same chain.

The gap common to all three: identity, policy, decision, and provenance are handled by separate subsystems that are not linked *inside the action itself*. Proving fleet-wide compliance then means correlating across subsystems, per unit, after the fact. The architecture below closes that gap by making the five properties one chain that every action passes through and records.

The Architecture

The disclosure's core primitive is the **governed observation**: the single unit of exchange across the whole fleet. Every communication a unit emits, and every contribution to the shared world model, is formatted as a governed observation carrying, at minimum, an authority-credential field, a dynamic-device-hash field encoding the emitting device's identity continuity, spatial and temporal reference fields, a time-to-live, a payload, and a lineage field recording provenance. Because every message has this shape, the same governance applies uniformly regardless of sensing modality, medium, or device tier.

On top of that primitive, the disclosure composes what it presents (FIG. 28D) as a **five-property governance chain**, enforced in sequence and closing recursively so that every primitive's output enters the chain and every actuation passes through every primitive's governance. Mapped to the five governance properties this guide is organized around:

1. **Identity.** Each contribution carries an authority credential, and device identity is established through *trust-slope continuity*: a dynamic device hash that evolves across successive transmissions, which receivers use to validate a continuity-

preserving identity. The disclosure contrasts this with static credentials such as long-lived API keys, certificates, or shared secrets. Identity is a property the receiver re-verifies, not a token it trusts once.

2. **Policy.** The authority credential binds the sender to a position in an *authority taxonomy*: a policy-configurable hierarchical trust structure that specifies, per authority level, the behavioral semantics, the evidential weighting a contribution earns, mutation-injection privileges, and supersession relationships. Policy is not a side check; it is the meaning the credential carries into every consuming agent.
3. **Admissibility.** A consuming agent does not simply accept an authenticated message. It runs the contribution through a *composite admissibility evaluator* that weighs the proposed mutation against multiple cognitive domain fields and produces a graduated outcome. The disclosure describes outcomes such as accepted, gated, deferred, or rejected, and elsewhere a wider graduated set (admit, gate, defer, solicit, reject, handoff, escalate, quarantine, degraded-mode). Admissibility is where "who may contribute what, under what present conditions" is decided.
4. **Lineage.** Every observation, evaluation, and actuation is recorded in a lineage field that links causes to effects deterministically. Lineage fields compose: one observation's lineage links to its source observations, across any number of derivation steps, producing a cross-device, cross-authority provenance record. The disclosure states plainly that this deterministic provenance record supports regulatory compliance reporting and incident investigation without reliance on transient telemetry. This is the property that makes compliance provable rather than reconstructed.
5. **Continuity.** The chain is designed to hold across boundaries. The same authority-credentialed observation, composite admissibility, and lineage mechanisms are described as operating identically across distributed, centralized, and hybrid topologies, and a cross-authority boundary agent can translate an observation from

one domain's authority taxonomy into an equivalent observation in another's without losing governance or lineage continuity. Continuity is what lets the *fleet* share one compliance story rather than a patchwork of per-domain ones.

The unifying claim is separation of layers: a governance-semantic layer (credential, weighting, admissibility, graduated response, lineage) held invariant, and a transduction/transport layer (specific sensors, radios, actuators, crypto) that can be substituted underneath without changing the governance properties. That separation is precisely what makes the compliance property uniform across a heterogeneous fleet.

How to Approach the Build

The following order mirrors the dependency structure in the disclosure. Everything is yours to implement.

1. **Define the observation record first.** Fix a message format with the mandatory fields: authority credential, device-identity hash, spatial reference, temporal reference, time-to-live, payload, and lineage. Nothing else works until this is stable, because it is the single unit every property attaches to.
2. **Design the authority taxonomy for your regulatory domain.** Enumerate the authority levels a regulator or your safety policy recognizes, and specify per level: how much evidential weight a contribution earns, what mutations it may inject, and what supersedes what. Keep it policy-configurable; the disclosure treats the taxonomy as data, not code, so amendments do not require rearchitecting.
3. **Implement continuity-based identity.** Have each device derive a dynamic device hash that evolves per transmission, and have receivers validate the trust slope rather than trusting a static key. An illustrative interface sketch (illustrative only, faithful to the disclosed roles):

```
# Illustrative only. You implement the actual mechanisms.
cred      = authority_credential(device)          # position in taxonomy
ddh       = dynamic_device_hash(device, prev_ddh)  # trust-slope continui
obs       = GovernedObservation(cred, ddh, space, time, ttl, payload,
                                lineage=[source_refs, deriv_fn, attestatio
```

4. **Build the composite admissibility evaluator on the consumer side.** For each incoming observation, evaluate the proposed mutation against your agent's domain fields and emit a graduated outcome (at minimum accept / gate / defer / reject). Record the outcome; do not let anything reach an actuator ungated.

```
# Illustrative.
outcome = evaluate_admissibility(obs, agent.domain_fields)  # graduated
if outcome in ACTUATING_OUTCOMES:
    actuate(...)
lineage.record(obs, outcome, evidence=obs.lineage)          # always
```

5. **Make lineage recording non-optional and composable.** Every observation, every admissibility decision, and every actuation writes a lineage entry that references the observations it derived from and carries a cryptographic integrity attestation over the fields. This is what later answers "on what evidence is this decision based." Because lineage composes across steps, a cross-device chain reconstructs deterministically.
6. **Handle continuity across boundaries last.** Where your fleet crosses authority domains, implement boundary agents that translate observations between taxonomies while preserving credential and lineage continuity, and confirm the same chain runs whether your topology is centralized, distributed, or hybrid.

7. **Turn the chain into compliance output.** Because every action already carries its provenance, your compliance report is a traversal of lineage, not a data-collection scramble. Build the query surface that, given any action, walks its lineage back to source evidence and forward to effects.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install, no SDK, and nothing here has been benchmarked or productized. The disclosure describes mechanisms and their relationships; it does not hand you a reference implementation, tuned parameters, or performance numbers, and this guide does not either. Where the disclosure leaves a choice open (which forward-error-correction scheme, which cryptographic primitive, how domain fields are computed), you make and validate that choice.

It also does not replace your legal determination of what "compliant" means. The chain makes *the evidence for an action* uniform and reconstructable; whether a given policy satisfies a given regulation is a judgment you and your counsel make, encoded into your authority taxonomy and admissibility rules. The architecture enforces and records the policy you write; it does not write it.

Finally, the property is only as good as your discipline: if any code path emits an action without passing the chain or writing lineage, that action is exactly the un-governed, un-reconstructable gap the architecture exists to eliminate. Uniformity is the whole value, so the chain has to be the only path to actuation.

Disclosure Scope

The approach described in this guide is disclosed in U.S. Provisional Application No. 64/049,409, which presents the Five-Property Governance Chain and the governed-observation primitive it composes. This guide is educational: it explains an architecture

a developer can build. It is not a shipping product, a benchmark result, a warranty, or an offer of software, and nothing here should be read as a claim that an implementation exists, performs to any stated level, or satisfies any particular regulation. Every mechanism described traces to that filing; design choices left open in the filing are left open here.

Five-Property Governance Chain (/governance-chain)

All 40 steps → (/inventive-steps)

The umbrella primitive: every mutation passes through the same five-property structural test.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Five-Property Governance Chain: The Architectural Umbrella (/articles/five-property-governance-chain-the-architectural-umbrella)

SECONDARY TECHNICAL

- Authority-Credentialed Observations (/articles/governance-chain/authority-credentialed-observation)
- Evidential Weighting in Governance Chain (/articles/governance-chain/evidential-weighting)
- Composite Admissibility Evaluation (/articles/governance-chain/composite-admissibility)
- Governed Actuator Execution (/articles/governance-chain/governed-actuator-execution)
- Lineage-Recorded Provenance (/articles/governance-chain/lineage-recorded-provenance)
- Recursive Closure Across Governance Chain (/articles/governance-chain/recursive-closure)
- Hierarchical Governance Composition (/articles/governance-chain/hierarchical-composition)
- Technology-Neutral Governance (/articles/governance-chain/technology-neutrality)
- Structural Distinction Test for the Five-Property Governance Chain (/articles/governance-chain/structural-infringement-test)

APPLICATIONS · GENERAL

- Cross-Domain Governance: One Auditable Authority Chain Across Defense, Cyber, Health, and Finance (/articles/governance-chain/cross-domain-governance-umbrella)
- Multi-Jurisdiction Compliance Without a Single Supervening Authority: A Federated Governance Chain (/articles/governance-chain/federated-governance-umbrella)
- Multi-Jurisdictional Compliance for Cross-Border Data Pipelines: A Governance Chain Umbrella (/articles/governance-chain/multi-jurisdictional-governance-umbrella)
- AI Governance Umbrella Across Regulatory Regimes (/articles/governance-chain/ai-governance-umbrella)
- Climate Governance Umbrella (/articles/governance-chain/climate-governance-umbrella)
- Multi-Tier Supply Chain Provenance: One Governance Substrate for NIST 800-161, CISA SSDF, NDAA 5949, CSDDD, and DSCSA (/articles/governance-chain/supply-chain-governance-umbrella)
- A CCPA and CPRA Compliance Architecture for Verifiable Consumer Rights and ADMT (/articles/governance-chain/ccpa-cpra-privacy)
- EU CSDDD Supply-Chain Due Diligence: A Credentialed Governance-Chain Architecture (/articles/governance-chain/eu-csddd-due-diligence)
- CSRD Audit-Ready Sustainability Reporting: A Governance-Chain Architecture for ESRS Assurance (/articles/governance-chain/eu-csrd-sustainability)
- EU Data Act Compliance for Connected-Product Data: A Credentialed Data-Flow Architecture (/articles/governance-chain/eu-data-act)
- NIS2 Compliance Architecture: How to Meet EU 24-Hour Cyber Incident Reporting (/articles/governance-chain/eu-nis2-cyber)
- FedRAMP High and DoD IL5/IL6: Continuous Compliance Evidence as a System Property (/articles/governance-chain/fedramp-il5-il6)
- A GDPR Article 22 Compliance Architecture for Automated Decision-Making and Profiling (/articles/governance-chain/gdpr-article-22)
- HIPAA Security Rule Compliance for Cross-Organization ePHI Access (/articles/governance-chain/hipaa-security-rule)
- How to Produce Auditable IEEE 7000 Series Conformance Evidence (/articles/governance-chain/ieee-7000-series)
- How to Build an ISO/IEC 42001 AI Management System on a Governance-Chain Substrate (/articles/governance-chain/iso-iec-42001-ai-management)
- ITAR and EAR Export Control Compliance: Per-Access Provenance for Deemed Exports (/articles/governance-chain/itar-ear-export-controls)
- NDAA Section 1709 Compliance Architecture: Runtime China-Origin Controls for DoD Supply Chains (/articles/governance-chain/ndaa-1709-china-controls)

- NIST SP 800-53 Rev 5 Compliance for Autonomous and AI-Mediated Systems (/articles/governance-chain/nist-800-53-controls)

APPLICATIONS · SPECIFIC

- AWS IAM Cross-Account vs a Governed Cross-Authority Chain (/articles/governance-chain/aws-iam-cross-account)
- Hyperledger Fabric vs Governed Cross-Authority Composition (/articles/governance-chain/hyperledger-fabric)
- Microsoft Entra ID vs Governed Agent Execution: The Governance Chain Alternative (/articles/governance-chain/microsoft-entra-id)
- AWS Verified Permissions vs a Governed Physical-World Actuation Chain (/articles/governance-chain/aws-verified-permissions)
- CyberArk PAM vs Governed Actuator Execution: The Governance-Chain Substrate (/articles/governance-chain/cyberark-pam)
- Okta Alternative: Governed Cross-Authority Identity Beyond a Single Broker (/articles/governance-chain/okta-identity)
- Ping Identity vs Governed Actuation: A Cross-Vendor Governance Chain (/articles/governance-chain/ping-identity)
- SailPoint IGA vs the Governance Chain: Credentialed Identity Mutation (/articles/governance-chain/sailpoint-iga)
- EU eIDAS 2 and the EUDI Wallet vs a Governed Agent-Execution Chain (/articles/governance-chain/eu-eidas-2-eudi-wallet)
- What happens after Microsoft Entra Verified ID verifies a credential? (/articles/governance-chain/microsoft-entra-verified)
- Pollen Mobile vs Governed Coverage Evidence: A DePIN Governance Chain (/articles/governance-chain/pollen-mobile)

Five-Property Governance Chain overview → (/governance-chain)