Nick Clark · Published April 25, 2026 · PDF

How to Prove Every AI Action Satisfied Identity, Policy, and Audit

You are running autonomous agents that sense, decide, and act, and someone eventually asks you to prove that a specific action was permitted: that its source was who it claimed to be, that policy allowed it, and that you can reconstruct exactly why it happened. Bolting three separate systems together for identity, policy, and audit leaves gaps at the seams. This guide describes an architectural approach that composes those properties into a single chain so an action only proceeds when every property validates end to end. It describes an architecture disclosed in U.S. Provisional Application No. 64/049,409, the Five-Property Governance Chain inventive step, not a shipping library you can install.

What You Are Building

You are building a system where every action an autonomous agent takes carries its own proof of governance. When an agent senses something, decides something, or actuates something in the physical or digital world, you want to be able to answer three questions after the fact, and ideally block the action before the fact if any answer is wrong:

- **Identity**: was the source of this input the device or party it claimed to be, and not a spoof or replay?

- **Policy**: was this action permitted given the source's authority and the governing rules in force?
- **Audit**: can you deterministically reconstruct what inputs led to this action, who contributed them, and how the decision was reached?

The people who need this are teams deploying agents that touch consequential state: vehicle and infrastructure coordination, industrial actuation, multi-party sensing networks, anything where "the model did it" is not an acceptable answer to a regulator or an incident review. The goal is not three dashboards. The goal is that identity, policy, and audit are structural properties of every action, checked together, such that an action that fails any one of them cannot execute.

The approach described here is the Five-Property Governance Chain, disclosed in U.S. Provisional Application No. 64/049,409.

Why the Obvious Approaches Fall Short

The normal way to build this is to assemble three mature, independent subsystems, each of which is good at its own job.

For identity, you reach for public-key infrastructure: enroll each device with a certificate authority, issue it a credential, and check the signature on each message. PKI is a well-understood, widely deployed standard, and for many systems it is exactly right. Its structural cost here is that it authenticates a credential, not the ongoing genuineness of a device. A valid credential that has been stolen still validates, and enrollment assumes each device can reach an enrollment server before it operates.

For policy, you reach for an authorization engine: a rules service that returns permit or deny for a given principal and action. This too is standard and useful. Its structural limit is that the decision is typically binary and evaluated against a single dimension at

a time, and it usually treats the caller's identity as an already-settled fact handed to it by a separate layer.

For audit, you reach for logging: emit an event to a log store on every significant action. Logging is essential. Its structural limit is that logs are written after decisions and reference identity and policy by copied-over identifiers, so reconstructing a full decision chain means correlating three separate systems that were never designed to compose.

The gap is not in any one piece. It is in the seams. Because identity, policy, and audit live in separate substrates, there is no single object that carries all three, and no single checkpoint that requires all three to hold at once. An action can pass authentication, pass a policy check evaluated in isolation, and be logged, while the audit record cannot actually prove that the policy check saw the correct, verified identity, or that the input was genuine rather than a replay with a stolen credential.

The Architecture

The disclosed approach closes the seams by making governance a single composed chain imposed on every governed mutation, rather than three subsystems consulted independently. Per the filed specification, the chain has five properties, and an action is only permitted when every property validates end to end.

1. Authority-credentialed observation. Every input is a *governed observation*, not raw telemetry. Per the spec, a governed observation carries, at minimum, an authority-credential field, a device-identity field, spatial and temporal references, a time-to-live, a payload, and a lineage field. The authority credential is evaluated against a governance-configurable *authority taxonomy* that carries hierarchical trust semantics, not a binary valid-or-invalid flag. The taxonomy supports dynamic escalation and de-escalation, where an entity is temporarily elevated under policy-defined conditions with a maximum duration and scope, and every escalation is itself recorded.

2. Evidential weighting in a shared store. Observations entering a shared governed observation store are weighted by the contributing authority level, the sensing modality's reliability, and inter-source consistency. A high-authority, corroborated observation carries more evidential weight than a low-authority or inconsistent one. This is what lets policy be graduated rather than a single binary gate.

3. Composite admissibility evaluation. Before any mutation is admitted, it is evaluated across multiple cognitive-domain fields together: per the spec, dispositional, integrity, confidence, and capability fields. The evaluation produces a graduated outcome, described in the spec as accepted, gated, deferred, or rejected, rather than a bare permit or deny. This is the checkpoint where the composed identity and policy evidence is judged as a whole.

4. Governed actuator execution. Every physical actuation requires composite admissibility approval. Actuation is not a separate code path that trusts an upstream decision; it is gated by the same evaluation, so there is no way to actuate around the chain.

5. Lineage-recorded provenance. Every observation, evaluation, and action is linked through a deterministic lineage field. Per the spec, the lineage field records the contributing-device identifier, references to source observations where the observation is derived from others, a derivation-function identifier where applicable, and a cryptographic integrity attestation over the fields. Lineage fields compose: one observation's lineage links to the lineage of the observations it was derived from, producing a cross-device, cross-authority provenance record that permits deterministic reconstruction of how a decision was reached.

Two properties of the composition are what make this more than the sum of the parts.

First, **identity is continuity, not a static certificate.** The disclosed identity mechanism is *trust-slope continuity*. Each transmitting device computes a *dynamic device hash* from inputs such as device entropy, sensor readings, configuration and

clock state, and prior transmission content, so the hash evolves gradually across successive transmissions in a way reflective of the device's real operational state. Receivers keep a history of a device's hashes and compute a trust-slope metric measuring whether a newly received hash is consistent with genuine operational evolution. Per the spec, this detects spoofing and replay through discontinuities in the hash sequence *regardless of whether the spoofing device possesses a valid static credential*, and it does not require enrollment with a central certificate authority or storage of long-lived secrets. The continuity-validation output is consumed directly by the composite admissibility evaluator, so identity feeds the same checkpoint as policy rather than living in a separate layer.

Second, **the chain is recursive and closed.** Per the spec, every primitive's output enters the chain as an authority-credentialed observation, and every actuation passes back through every primitive's governance. There is no privileged path that skips a property. That closure is what lets you make the strong claim the search query asks for: not that an action was probably fine, but that it structurally could not have executed unless identity, policy (admissibility), and audit (lineage) all held together.

How to Approach the Build

This is an architecture you implement yourself. The steps below are the order a developer would sensibly follow.

Step 1: Define your governed observation as the only unit of exchange. Nothing enters your system as raw telemetry. Every input is wrapped in a structure carrying, at minimum, the fields the spec enumerates. An illustrative interface sketch, faithful to the disclosed fields but not a library:

```
// Illustrative only — you implement this to fit your domain
GovernedObservation {
  authorityCredential   // source's credential in your taxonomy
  deviceIdentity        // dynamic device hash (see Step 3)
  spatialRef, temporalRef, ttl
  payload               // domain-specific content
  lineage {             // provenance
    contributingDeviceId
    sourceObservationRefs[]   // what this was derived from
    derivationFn
    integrityAttestation      // signature over the fields
  }
}
```

Step 2: Model your authority taxonomy. The taxonomy is governance-configurable and domain-specific: the spec gives examples for defense, healthcare, and warehouse/port domains, among others. Decide your levels, and decide the escalation and de-escalation conditions, maximum durations, and scopes for temporary elevation. Record every escalation as its own governed event.

Step 3: Implement trust-slope continuity for identity. At each transmitter, compute a dynamic device hash from evolving device state and attach it to every observation. At each receiver, keep a policy-windowed history of hashes per source and implement a trust-slope validator that scores a new hash against the sequence. Its output is not a boolean you branch on directly; it is evidence fed into Step 5. Note the tradeoff: continuity needs a history to judge against, so a brand-new source has a short slope, and you will want policy-defined tolerance windows for legitimate discontinuities like maintenance and device replacement, exactly as the spec calls for.

Step 4: Stand up the shared observation store with evidential weighting. Weight each observation by authority level, modality reliability, and inter-source consistency. This is what turns three separate yes/no checks into a single weighted body

of evidence.

Step 5: Build the composite admissibility evaluator as your single checkpoint. This is the heart of the build. Evaluate each proposed mutation across your cognitive-domain fields together, dispositional, integrity, confidence, and capability, with the trust-slope output and the evidential weighting as inputs, and emit a graduated outcome (accepted, gated, deferred, rejected). Route every actuation through this same evaluator; do not let any actuation path bypass it.

Step 6: Make lineage a write requirement, not an afterthought. Every observation, every evaluation, and every action writes a lineage record that references its sources and carries an integrity attestation. Because lineage composes, a later audit walks the references backward and deterministically reconstructs the full chain. Design this before you write your first actuator, because retrofitting composable lineage onto a system that already logs to a side channel is the hard path.

Step 7: Verify closure. Audit your own call graph for any path that senses, decides, or actuates without going through Steps 1 through 6. The security property comes from closure; a single ungoverned path defeats it.

What This Does Not Give You

Be clear-eyed about the boundaries.

This is an architecture, not a drop-in library. There is no package to install and no SDK behind this guide. You implement every component yourself, and the interface sketches above are illustrative, not runnable.

It is disclosed in a patent filing. It has not been presented here as a benchmarked or production-proven product, and this guide states no performance numbers, latency figures, or security guarantees beyond the structural properties the specification describes.

The strength of the guarantee is structural and only as good as your closure. If any code path can sense, decide, or actuate outside the chain, the end-to-end claim does not hold for that path. The architecture makes governance provable when it is universal; it does not retroactively govern what bypasses it.

It does not replace the primitives it composes. You still need real cryptography for the integrity attestations, a real store for observations, and real policy authoring for your taxonomy. Trust-slope continuity is an identity approach with a different threat model than PKI, not a superset of it, and it depends on having enough history to judge continuity, so it fits systems with ongoing communication between parties better than one-shot interactions.

Finally, the disclosed examples center on spatial and sensor-actuator systems. The chain is described as technology-neutral, but you are responsible for judging whether it fits your domain.

Disclosure Scope

The architectural approach described in this guide, the Five-Property Governance Chain, is disclosed in U.S. Provisional Application No. 64/049,409. Every mechanism described above, the governed observation primitive, the authority taxonomy, trust-slope continuity identity, evidential weighting, composite admissibility evaluation, governed actuator execution, and lineage-recorded provenance, traces to that filing. This guide is educational: it explains an architecture so that a skilled developer can understand and build it themselves. It is not a warranty, not a benchmark, and not an offer of software, and nothing here should be read as a claim that a shipping implementation exists.

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)