Nick Clark · Published April 25, 2026 · PDF

How to Log Operator Intent for Post-Incident Review of an Autonomous System

If your autonomous system logs what it did but not what the operator declared it should do, post-incident review turns into guesswork: you can see the action, but you cannot prove whether it matched intent. This guide describes an architecture for recording declared operator intent as a first-class, attested record so reviewers can compare intent to action after the fact. The approach is disclosed in U.S. Provisional Application No. 64/049,409, and it is anchored in the Operator Intent inventive step. This is an architecture you build yourself, not a shipping library.

What You Are Building

You are building an intent-logging layer for an autonomous or semi-autonomous system so that, after an incident, an investigator can answer a precise question: did the machine do what the operator declared it intended, or did action diverge from intent, and where?

The problem shows up the moment something goes wrong with a self-driving vehicle, a warehouse robot, a port vessel, an airfield ground vehicle, or any mixed-fleet operation where autonomous, semi-autonomous, and manual units share the same space. Most systems produce rich action logs: throttle, steering, actuator commands, sensor frames.

What they rarely produce is a durable, tamper-evident record of the intent that was declared before or during the action, attributed to a specific authority, timestamped, and retrievable in a form a reviewer can trust as evidence.

The goal here is an architecture where declared intent is captured as a governed record, bound to the unit and authority that produced it, and preserved on the same lineage backbone as the actions themselves, so that intent and action can be laid side by side during review. This guide describes that architecture as disclosed in U.S. Provisional Application No. 64/049,409. You implement it against your own system.

Why the Obvious Approaches Fall Short

The usual way teams try to reconstruct intent after an incident is to infer it from the action logs. You replay the telemetry and argue backward: the vehicle braked, so it must have intended to yield. This is circular. The action is exactly the thing under investigation, so using it as the source of intent begs the question the review is trying to answer.

A second common approach is to log operator inputs (a button press, a mode change, a route entry) as plain application events in a general logging pipeline. This is better, but it has structural gaps for forensic use. Plain log lines usually carry no authority attribution, so you cannot say which credentialed source asserted the intent. They are mutable or at least not tamper-evident, which weakens them as evidence. And they treat every input identically, with no way to distinguish a fully declared machine intent from a partial signal or an inference.

Vehicle-to-everything messaging standards illustrate the same limit from the communication side. Dedicated Short-Range Communications and Cellular Vehicle-to-Everything define cross-vehicle message formats and authenticate messages through public-key infrastructure. These are real and useful, but the spec notes that such systems authenticate messages homogeneously: an authenticated message is treated the

same regardless of the governance authority behind it, and there is no fidelity-tier structure distinguishing a fully shared intent from an inferred one. For post-incident review you need more than "this message was authentic." You need to know what kind of intent it was, who was entitled to assert it, and how much weight it carried.

The structural gap, then, is that intent is not being recorded as its own attributed, weighted, tamper-evident artifact on the same provenance record as the action. That is what the disclosed architecture supplies.

The Architecture

The disclosure treats operator intent as a first-class architectural primitive. Every intent record is a governed observation: a message that carries an authority credential, a temporal scope, and a cryptographic integrity attestation binding the record to the authority that produced it, plus a lineage field recording provenance. Intent is not a side log; it flows on the same governed-observation substrate as sensing and actuation.

Three parts of the disclosure matter most for review.

First, fidelity tiers. The disclosure classifies each operating unit into one of several fidelity tiers reflecting how much intent it can actually disclose. In the exemplary three-tier form: a full-fidelity tier, where a highly integrated unit shares cognitive state such as its planning graph, its committed execution, and its confidence and capability state; a structured partial-fidelity tier, where a unit shares specific structured signals extracted from an integrated bus (for a ground vehicle, examples given include pedal position, steering input, turn-signal state, and navigation destination); and a behavior-inferred tier, where the mesh infers intent for a legacy unit from externally visible cues and records that inference as a governed, credentialed observation. The point for review is that a record never silently pretends to be more authoritative than its source. A tier-3 inference is stored as an inference, attributed to the inferring agent and its inference function, distinct from a tier-1 declaration.

Second, the intent-lineage recorder. The disclosure specifies a component that records each intent emission, admission, fusion, verification, retraction, and downstream consumption in the governance-chain lineage field. This is the spine of post-incident review. It is not just "intent was declared." It is a chained record of the intent's whole life: when it entered, whether it was admitted, what it was fused with, whether it was later verified against outcome, whether it was retracted or corrected, and which downstream decisions consumed it.

Third, verification and corrigibility, which give review its comparison. The disclosed intent-verification feedback loop records each inferred-intent observation with its input cues, classification, and confidence, records the unit's subsequent observable action, and runs a verification evaluator comparing predicted intent against observed outcome. Separately, the intent-retraction and correction mechanism lets a contributing authority retract or correct an intent, producing a retracted-plus-superseded pair. Critically, the disclosure states that retracted intent observations remain in the governance chain as retracted-and-superseded rather than being deleted, preserving audit and forensic-reconstruction properties. Nothing is quietly overwritten.

These records connect to action through the governed actuation chain. The disclosure describes an actuation chain in which a proposed action is evaluated for admissibility, a graduated actuation mode is selected, the evaluation inputs and determinations are recorded in the lineage field before the actuator is commanded, the actuator executes, effects are observed and verified, and the executed action plus verified effect are recorded in the lineage field afterward. Because intent records and actuation records share the same lineage backbone, an investigator can trace from a declared intent forward to the actions that consumed it, and from an executed action backward to the intent evidence that was admissible when it fired.

Finally, the disclosure provides an explicit forensic path. Its temporal-reconstruction mechanism accepts a target time and query scope, retrieves the observations admissible at that time, and re-evaluates them under the admissibility rules in force at that time

per the governance-policy-version lineage, producing a reconstructed view. The disclosure lists incident investigation as an explicit pattern: reconstructing the view state leading up to an incident for causal analysis. That reconstruction is itself a governed observation with lineage linking it to the historical records it was built from.

How to Approach the Build

Work outward from the record, then from review back to capture.

1. Define an intent record as a governed observation, not a log line. Give it an authority credential (who asserted this intent), a temporal scope (when it is valid), a cryptographic integrity attestation binding it to that authority, and a lineage field. A faithful, illustrative interface sketch (you implement the internals):

```
// Illustrative only; not a shipping API.
IntentRecord {
  authority_credential   // who declared it
  fidelity_tier          // full | structured-partial | behavior-inferred
  temporal_scope         // valid-from / valid-until
  classification         // the declared or inferred intent
  confidence             // for inferred or partial-fidelity intent
  inference_function_id  // present only for behavior-inferred records
  lineage                // provenance chain
  integrity_attestation  // binds the above to the authority
}
```

2. Classify the source into a fidelity tier before you store the record, and never launder the tier upward. Full-fidelity declarations, structured partial signals from an integrated bus, and mesh inferences from external cues each get stored as what they are. For inferred intent, record the inference function identifier and the input-cue lineage so a reviewer can re-examine how the inference was reached.

3. Stand up the intent-lineage recorder as an append-oriented spine. Record emission, admission, any fusion, verification, retraction or correction, and downstream consumption. Treat retraction as supersession, not deletion: keep the retracted-and-superseded pair so the earlier assertion and its correction both survive.
4. Bind intent to action on one lineage backbone. When an action is evaluated and executed, record its evaluation inputs and determinations before the actuator fires and its executed form and verified effect afterward, on the same lineage the intent records live on, so intent and action are joinable at review time.
5. Add the verification comparison. Record the declared or inferred intent, record the subsequent observed action, and compute the comparison. In steady state this supports learning, per the disclosure; for review it gives you the intent-versus-outcome delta directly rather than as an argument reconstructed from telemetry.
6. Build review on temporal reconstruction. Give investigators a query that takes a target time and scope and returns the intent and action records admissible at that time, re-evaluated under the policy version then in force. Keep the reconstruction itself lineage-linked to its inputs so the review output is auditable in turn.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install and nothing here "just works" out of the box. You implement the record format, the tier classifier, the lineage recorder, the actuation binding, and the reconstruction query against your own platform.

It is disclosed in a patent filing, not shipped as a benchmarked product. The disclosure describes mechanisms and structure; it does not supply performance numbers, and this guide invents none. Do not read it as a claim of latency, storage cost, or accuracy.

Its guarantees are the ones the mechanisms provide and no more. Cryptographic attestation binds a record to the authority that asserted it and makes tampering evident; it does not make a lying operator honest or a bad inference correct. A behavior-inferred record is only as good as its inference function and the cues available, which is exactly why the disclosure keeps such records tier-labeled and confidence-tagged. The value at review time is a faithful, attributed, tamper-evident account of what was declared and inferred, set against what was done, not an oracle of true intent.

It also assumes you can carry governed records with authority credentials and lineage end to end. If part of your stack cannot attest records or preserve lineage, intent logging through that part degrades to the same weak plain-log evidence this architecture exists to replace.

Disclosure Scope

The operator-intent logging approach described here, including the fidelity-tier classification of intent sources, the intent-lineage recorder, the intent-verification feedback loop, the retraction-and-supersession corrigibility mechanism, and the temporal and forensic reconstruction path for incident investigation, is disclosed in U.S. Provisional Application No. 64/049,409. This guide is educational. It explains an architectural approach so that a skilled developer can build it, and it is not a warranty, a specification of a shipping product, or an offer of software. Every mechanism described above is drawn from that filing; where the filing is silent, this guide makes no claim.

Operator Intent (/operator-intent) All 40 steps → (/inventive-steps)

Graduated fidelity tiers. Verification-feedback evolution. Risk versus hostility, separated.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- Operator Intent: Graduated Fidelity Tiers for Mixed-Fleet Coordination (/articles/operator-intent-graduated-fidelity-tiers-for-mixed-fleet-coordination)

SECONDARY TECHNICAL

- Three-Tier Intent Fidelity (/articles/operator-intent/graduated-fidelity-tiers)
- Tier-Weighted Admissibility (/articles/operator-intent/tier-weighted-admissibility)
- Behavior-Inferred Intent as Governed Observation (/articles/operator-intent/inferred-intent-as-observation)
- Verification-Feedback Inference Function Evolution (/articles/operator-intent/verification-feedback-loop)
- Inference Function Evolution Under Aggregated Feedback (/articles/operator-intent/inference-function-evolution)
- Risk vs Hostility Profile Bifurcation (/articles/operator-intent/risk-vs-hostility-bifurcation)
- Due-Process Credentialing for Adverse Classifications (/articles/operator-intent/due-process-credentialing)
- Cross-Domain Adversarial Inference (/articles/operator-intent/cross-domain-adversarial-inference)
- Protective-Order Integration With Operator-Intent Inference (/articles/operator-intent/protective-order-integration)
- Counter-Action Selection Under Hostility Classification (/articles/operator-intent/counter-action-selection)

APPLICATIONS · GENERAL

- Usage-Based Insurance Telematics: A Credentialed, Consent-Gated Operator Risk Profile for Behavior-Based Coverage (/articles/operator-intent/usage-based-insurance-telematics)
- Intent-Bound Aviation Mission Execution (/articles/operator-intent/intent-bound-aviation-mission)
- Intent-Bound Defense Engagement: Structuring Meaningful Human Control Over Autonomous Weapons (/articles/operator-intent/intent-bound-defense-engagement)
- Binding Surgical-Robot Autonomy to Surgeon Intent for Audit-Grade Accountability (/articles/operator-intent/intent-bound-surgical-procedure)
- How to Govern Autonomous Policing Robots: Multi-Authority Intent for De-Escalation Systems (/articles/operator-intent/autonomous-policing-de-escalation)
- Authority Composition for Autonomous Research Platforms and Self-Driving Labs (/articles/operator-intent/autonomous-research-platforms)

- Who Authorizes a Care Robot's Action? Intent-Bound Elder Care and Companion Robotics (/articles/operator-intent/intent-bound-elder-care-robotics)
- Meaningful Human Control for Autonomous Weapons: An Architecture That Makes It Structural (/articles/operator-intent/meaningful-human-control-doctrine)
- Search-and-Rescue Coordinated Intent: Auditable Multi-Operator Command Across Ground, Air, and Autonomous Drone Assets (/articles/operator-intent/search-rescue-coordinated-intent)
- DoD Directive 3000.09 Compliance: Meaningful Human Control Architecture for Autonomous Weapon Systems (/articles/operator-intent/dod-3000-09-autonomous-weapons)
- EASA U-space Compliance Architecture for Drone Airspace Integration (/articles/operator-intent/easa-u-space-airspace)
- FAA UTM Strategic Deconfliction: Credentialed Operator Intent for BVLOS Drone Traffic Management (/articles/operator-intent/faa-utm-uas-traffic-mgmt)
- Meaningful Human Control for Autonomous Weapons: An Architecture for UN CCW LAWS Compliance (/articles/operator-intent/un-ccw-laws-doctrine)

APPLICATIONS · SPECIFIC

- Anduril Mission Control vs Governed Operator Intent: The Meaningful-Human-Control Layer (/articles/operator-intent/anduril-mission-control)
- Northrop ABMS vs Governed Operator-Intent Composition for JADC2 (/articles/operator-intent/northrop-abms)
- Does Shield AI Hivemind enforce operator intent on autonomous actuation? (/articles/operator-intent/shield-ai-hivemind)
- Helsing vs Governed Operator Intent: A Meaningful-Human-Control Layer for Defense AI (/articles/operator-intent/helsing-defense-ai)
- Milrem Robotics THeMIS vs Credentialed Operator-Intent for Coalition UGVs (/articles/operator-intent/milrem-robotics)
- Palantir Foundry vs Governed Operator-Intent Execution (/articles/operator-intent/palantir-foundry-mission)
- Saildrone Alternative: Governed Operator-Intent for Maritime ISR Autonomy (/articles/operator-intent/saildrone-maritime-isr)
- Skydio Defense vs Governed Operator Intent: Adding a Credentialed Authority Layer to Autonomous ISR (/articles/operator-intent/skydio-defense)
- 1X NEO alternative: governed household humanoids beyond a single control loop (/articles/operator-intent/1x-humanoid)
- AeroVironment Switchblade vs Governed Operator-Intent Execution (/articles/operator-intent/aerovironment-switchblade)

- AgEagle eBee TAC vs governed operator intent: what the Blue UAS fixed-wing does not provide (/articles/operator-intent/ageagle-defense)
- Anduril Bolt vs Governed Operator-Intent Execution (/articles/operator-intent/anduril-bolt-drones)
- Autel EVO Max 4T vs Governed Operator-Intent Execution (/articles/operator-intent/autel-evo-defense)
- Governed Drone Operation Beyond DJI Enterprise: Credentialed Operator Intent (/articles/operator-intent/dji-enterprise)
- Figure Humanoid vs Governed Operator Intent (/articles/operator-intent/figure-humanoid)
- Can Parrot Anafi Operate in Coalition Mixed-Fleet Drone C2? (/articles/operator-intent/parrot-anafi-defense)
- Tesla Optimus vs Governed Humanoid Execution: The Operator-Intent Layer (/articles/operator-intent/tesla-optimus)
- Agility Robotics Digit vs Governed Operator Intent: Credentialing Whose Task a Humanoid Executes (/articles/operator-intent/agility-robotics-digit)
- Apptronik Apollo Alternative: Governed Multi-Operator Intent Beyond a Single Humanoid Stack (/articles/operator-intent/apptronik-apollo)
- Governed Public-Safety Drones Beyond BRINC: Credentialed Operator Intent (/articles/operator-intent/brinc-public-safety-drones)
- Sanctuary AI Phoenix vs Governed Operator Intent (/articles/operator-intent/sanctuary-ai-phoenix)
- Saronic Alternative: Governed Operator Intent for Fleet-Scale USV Tasking (/articles/operator-intent/saronic-autonomous-maritime)
- Governed Operator Intent for Unitree H1 Humanoid and Go2 Quadruped Fleets (/articles/operator-intent/unitree-humanoid-quadruped)
- Vatn Systems Autonomous Undersea Vehicles vs Governed Operator Intent (/articles/operator-intent/vatn-systems-undersea)
- Qualcomm C-V2X alternative: governed operator-intent binding above the cross-vehicle message layer (/articles/operator-intent/qualcomm-cv2x)

Operator Intent overview → (/operator-intent)