Nick Clark · Published April 25, 2026 · PDF

How to Build AI a Regulator or Insurer Can Trust, Structurally

If you ship autonomous agents into a regulated setting, someone will eventually ask you to prove why the system did what it did, and a saved chat transcript is not proof. This guide walks through an architectural approach where a decision's reasoning is deterministically reconstructible from the agent's own recorded state, so behavior is legible to an auditor rather than argued after the fact. It describes an architecture disclosed in United States Patent Application 19/647,395, not a shipping library, and it is grounded in the Human-Relatable Intelligence inventive step.

What You Are Building

You are building an autonomous agent whose every decision can be reconstructed later, by someone who was not in the room, from a record the agent itself carries. The problem is specific and common: a regulator, an insurer, or an internal risk officer asks "why did the system approve this, decline that, or escalate here," and the honest answer for most agent stacks is "we logged the prompt and the output, and we can guess at the middle." Guessing does not clear an audit and does not get an actuary to price the risk.

The goal here is different. You want an agent where the causal chain behind any action is not narrated after the fact by a summarizer, but is deterministically reconstructible from a structured record the agent accumulates as it runs. Auditability becomes a

property of the architecture rather than a feature you bolt on. This is the practical target of the Human-Relatable Intelligence inventive step disclosed in US Patent Application 19/647,395: making an agent's reasoning legible and reconstructible so that its behavior is trustable to the parties who carry the liability.

Why the Obvious Approaches Fall Short

The usual approaches are reasonable and each falls short in a structural way, not because they are badly built.

Prompt-and-response logging. You store inputs and outputs. This is genuinely useful for debugging, but it records the endpoints of a decision, not its interior. If the model weighed three options and abandoned two, that deliberation left no durable trace. An auditor asking "what alternatives were considered and why were they rejected" has nothing to read.

Post-hoc explanation. You ask a model to explain a decision after it is made. The explanation is plausible, but it is generated separately from the decision and is not guaranteed to be the actual cause of the behavior. A regulator has no way to distinguish a faithful explanation from a convincing reconstruction, so this does not settle the question of why.

Alignment tuning (for example, RLHF). This shapes the distribution of outputs toward preferred behavior. It makes the average output better, but it does not produce a per-decision, inspectable record of how a specific action was governed. The spec describes this category as satisfying only part of what human-relatable, legible behavior requires.

Safety wrappers. An external filter permits or blocks actions at the boundary. The wrapper's own decisions can be logged, but the agent's internal reasoning remains a black box behind the wrapper. You can audit the fence, not the field.

The common gap: in all four, the internal state that produced the behavior is either never recorded, recorded only at the edges, or reconstructed by a separate process whose fidelity you cannot verify. The disclosed architecture closes that gap by making the internal state itself the record.

The Architecture

The disclosed approach treats an agent not as a prompt pipeline but as a set of persistent, independently tracked cognitive fields plus a lineage field, all carried by the agent itself. Three properties do the work.

1. Cognitive state is structural, not incidental. The agent carries a schema of persistent fields. The filing describes a foundational set (intent, context, memory, policy reference, mutation descriptor, and lineage) extended with cognitive-domain fields (affective state, integrity, personality, confidence, and capability). Each field has a current value and a trajectory over time, and each is independently readable and auditable. The point is that the factors influencing a decision are named fields with defined positions, not opaque activations. There is something concrete to inspect.

2. The lineage field records the whole trajectory, and it is the source of truth. The system records each proposed mutation, each admissibility determination, and each cognitive-domain field update in the lineage field, such that, per the filing, "the complete behavioral trajectory of the semantic agent is deterministically reconstructible from the lineage field alone." This is the load-bearing claim for auditability. Reconstruction is not a summarizer's best effort; it is a replay. The spec describes forensic reconstruction where an agent's state at any historical point is recovered by replaying the deterministic update function over the sequence of recorded observations from the lineage, producing the exact state that existed at the queried timestamp. Determinism is what lets an auditor arrive at the same answer you do.

3. A governance gate decides admissibility from multiple fields, before commitment, and records the decision. The filing describes a governance gate as a composite admissibility evaluation that integrates signals from several cognitive-domain fields (integrity, confidence, affective state, capability, personality) to produce an admissibility determination for each proposed mutation before it is committed. When readiness is insufficient, the agent can transition to a non-executing cognitive mode: it keeps reasoning and evaluating alternatives without committing state changes. Crucially, the governance decision is itself written to lineage, so the record shows not only what the agent did but what it declined to do and on what composite basis.

Two design choices make this palatable to real auditors and privacy reviewers.

Abstraction in the record. The filing describes lineage entries that reference mutations abstractly: for affective mutations, an entry records the observation type, the direction of change, and the policy-compliance status, but not absolute field values or raw underlying signals. This permits lineage auditing (verifying the agent followed policy-compliant paths) without exposing moment-to-moment internal state or, where biological coupling is involved, raw physiological data. You get verifiable compliance without a surveillance log.

Continuity-based identity. Rather than resolving identity purely by a static credential presented at a point in time, the filing describes a trust-slope mechanism that accumulates identity confidence through behavioral continuity over time, with the trajectory carried in the agent's cryptographic lineage. For an insurer, "this actor has a verifiable, continuous history" is a different and often stronger evidentiary posture than "this actor presented a valid token once."

Taken together: named fields carry the state, a deterministic lineage makes that state replayable, and a governance gate turns each decision into a recorded, multi-factor admissibility event. That is what "structurally trustable" means in this architecture.

How to Approach the Build

You are implementing this yourself. The steps below are an order of operations, not a package to install.

Step 1, Define your field schema. Decide which persistent fields your domain needs. Start with the record-keeping spine (intent, context, memory, policy reference, mutation descriptor, lineage) and add only the cognitive fields your governance actually consumes. Each field needs a current value and a stored trajectory. The following interface sketch is illustrative only and is not a supported API:

```
// illustrative, not a shipped API
Field = { name, value, trajectory: [ {t, value} ], policyBounds }
Agent = { intent, context, memory, policyRef, mutationDescriptor,
          lineage, integrity, confidence, affectiveState, capability, person
```

Step 2, Make every state change go through a mutation, never a direct write. No field is mutated except by a proposed mutation that the governance gate evaluates. This is the discipline that keeps the lineage complete. If code can quietly reach in and change a value, your reconstruction guarantee is gone.

Step 3, Build the governance gate as a composite evaluator. For each proposed mutation, compute independent contributions from the relevant fields and combine them into an admissibility determination: permit, gate (pending more evaluation), or suspend into the non-executing mode. Keep the combining function deterministic and policy-specified so two evaluators on the same inputs agree.

Step 4, Record to lineage on every proposal and every decision. Write an entry for each proposed mutation, each admissibility determination, and each resulting field update. Record decisions to decline and to suspend as well as decisions to act.

Choose your abstraction level deliberately per the filing's pattern: record observation type, direction, and policy-compliance status where full values would be sensitive, so the record stays auditable without becoming a leak.

Step 5, Implement deterministic replay. Given the lineage and the specified update function, you must be able to reconstruct the agent's state at any past timestamp and get exactly one answer. Treat replay determinism as a test target: seed a run, capture lineage, replay, and assert the reconstructed state matches. If it diverges, you have hidden nondeterminism to hunt down before any auditor finds it.

Step 6, Wire the non-executing mode. When the gate withholds permission, the agent should keep generating and evaluating candidate alternatives against the same composite criteria rather than either forcing the action or halting silently. Log that deliberation. "Paused and reconsidered" is exactly the behavior an auditor wants to see recorded.

Step 7, Decide your identity posture. If relational continuity matters (repeat counterparties, operator recognition), design identity as accumulated trust over a behavioral history carried in lineage, rather than a single credential check. Keep the trajectory verifiable.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to `npm install` and no SDK implied here. You build the field schema, the governance gate, the lineage store, and the deterministic replay yourself, and the quality of your audit trail is exactly the quality of that implementation.

It is not benchmarked or productized in this guide. The filing discloses the approach; it does not hand you performance numbers, and neither will I invent them. Determinism is a property you must engineer and continuously test; any nondeterminism you leave

in the update path (unstable ordering, unrecorded inputs, wall-clock dependence) quietly breaks the reconstruction guarantee.

It also does not, by itself, make your policies correct. The gate faithfully enforces and records the policy you give it; if the policy is wrong, you get well-documented wrong decisions. Reconstructibility answers "why did it do this," not "was the rule good." And where a decision genuinely turns on an unrecorded external factor, no replay can surface what was never captured. Finally, the privacy-preserving abstraction is a design choice with a tradeoff: recording direction-and-compliance instead of raw values protects internal state but means some questions can only be answered at the abstraction level you chose to record.

Disclosure Scope

The architecture described here is disclosed in United States Patent Application 19/647,395. This guide is educational: it explains an approach a developer can study and implement independently. It is not a warranty, not an offer of software, and not a claim that a benchmarked or production product is being distributed. Where this guide references other technologies for context, those references are descriptive only. Any implementation decisions, and their consequences, are the reader's own.

PRIMARY TECHNICAL DISCLOSURE

- Human-Relatable Computable Intelligence: Structural Isomorphism Between Computational and Human Cognitive Dynamics (/articles/human-relatable-computable-intelligence-structural-isomorphism-between-computational-and-human-cognitive-dynamics)

SECONDARY TECHNICAL

- The Cross-Primitive Coherence Engine (/articles/human-relatable-intelligence/coherence-engine)
- Narrative Identity as Compressed Self-Model (/articles/human-relatable-intelligence/narrative-identity)
- Ecosystem Governance Credentials and Cross-System Trust Federation (/articles/human-relatable-intelligence/ecosystem-credentials)
- Anonymized Governance Telemetry Aggregation (/articles/human-relatable-intelligence/governance-telemetry)
- The Coherence Control Loop: Detection, Recording, Restoration (/articles/human-relatable-intelligence/coherence-control-loop)
- The Complete Thirteen-Stage Mutation Lifecycle (/articles/human-relatable-intelligence/mutation-lifecycle)
- Ten Conditions for Human-Relatable Behavior (/articles/human-relatable-intelligence/ten-conditions)
- Graceful Degradation With Active-Domain Registry (/articles/human-relatable-intelligence/graceful-degradation)
- Architectural Inversion: Agent Carries State, Substrate Provides Environment (/articles/human-relatable-intelligence/architectural-inversion)
- Sequential Cascade Structures in Cross-Primitive Coherence (/articles/human-relatable-intelligence/sequential-cascades)
- Conformity Attestation: Verifiable Architectural Compliance (/articles/human-relatable-intelligence/conformity-attestation)

APPLICATIONS · GENERAL

- Structural Cognition: Why Trustworthy AI Needs Cognitive Primitives, Not Better Prompts (/articles/human-relatable-intelligence/structural-cognition)
- How to Make High-Risk AI EU AI Act Compliant by Design: Cognitive Architecture for Transparency, Oversight, and Audit (/articles/human-relatable-intelligence/eu-ai-cognitive-architecture)
- Why AI Alignment Is Insufficient for Trustworthy AI: Structure Over Training (/articles/human-relatable-intelligence/why-alignment-is-insufficient)

- Enterprise Trust Through Architecture, Not Alignment (/articles/human-relatable-intelligence/enterprise-trust-through-architecture)
- Insurance Liability Reduction Through Human-Relatable AI (/articles/human-relatable-intelligence/insurance-liability-reduction)
- How to Build Consumer Trust in AI: Calibrated Confidence, Consistency, and Self-Correction (/articles/human-relatable-intelligence/consumer-trust-in-ai)
- Regulatory Future-Proofing Through Human-Relatable Architecture (/articles/human-relatable-intelligence/regulatory-future-proofing)
- How to Build a Durable AI Moat When Models Commoditize: Cognitive Architecture Over Scale (/articles/human-relatable-intelligence/competitive-differentiation)
- Mixed-Fleet Coordination for Autonomous and Human-Driven Vehicles (/articles/human-relatable-intelligence/mixed-fleet-coordination)
- Drone-Swarm Coordination Across Cooperative and Adversarial Airspace (/articles/human-relatable-intelligence/drone-swarm-coordination)
- Usage-Based Insurance With Due-Process Hostility Separation (/articles/human-relatable-intelligence/insurance-due-process)
- Protective Order Enforcement: Admissible, Cross-Jurisdiction Violation Evidence (/articles/human-relatable-intelligence/protective-order-enforcement)

APPLICATIONS · SPECIFIC

- OpenAI Safety vs Governed Cognition: Why Alignment Is Not Structural Isomorphism (/articles/human-relatable-intelligence/openai-safety)
- Constitutional AI vs Structural Cognitive Architecture: A Governed Alternative (/articles/human-relatable-intelligence/anthropic-constitutional)
- DeepMind Safety vs Structural Governance: Alignment Beyond Training Time (/articles/human-relatable-intelligence/deepmind-safety)
- Governed Agent Execution Beyond Meta Llama: The Runtime Layer Open-Weight Safety Leaves Open (/articles/human-relatable-intelligence/meta-llama)
- Inflection AI Pi Alternative: Governed, Coherent Personal AI (/articles/human-relatable-intelligence/inflection-ai)
- Adept AI Action Agents vs Governed Agent Execution (/articles/human-relatable-intelligence/adept-ai)
- Covariant Alternative: Governed Robotic Manipulation Beyond Trained Dexterity (/articles/human-relatable-intelligence/covariant)
- Sanctuary AI Alternative: Human-Relatable Cognition Beyond Humanoid Form (/articles/human-relatable-intelligence/sanctuary-ai)

- Aleph Alpha Alternative: Governed Cognition Beyond Sovereign Hosting (/articles/human-relatable-intelligence/aleph-alpha)
- Mistral AI Alternative: Governed Coherence Beyond Efficient Open-Weight Models (/articles/human-relatable-intelligence/mistral-ai)
- Cambridge Mobile Telematics Alternative: Continuity-Based Driver Identity Beyond Server-Side Inference (/articles/human-relatable-intelligence/cambridge-mobile-telematics)
- Nauto Alternative: Auditable Driver Classification Beyond Inference Output (/articles/human-relatable-intelligence/nauto)
- Lytx Alternative: Governed Driver Identity Beyond Behavioral Aggregation (/articles/human-relatable-intelligence/lytx)

Human-Relatable Intelligence overview → (/human-relatable-intelligence)