

How to Give an AI Companion a Consistent Personality Across Sessions

If you build a conversational companion, users notice when it forgets who it was yesterday: the warmth resets, the caution resets, every session starts from a blank emotional slate. This guide describes an architecture for carrying disposition as durable state on the agent itself, so personality persists and evolves instead of being re-inferred each turn from the prompt. The approach is disclosed in United States Patent Application 19/647,395; it is an architecture you implement, not a shipping library. The home inventive step is the Affective State inventive step.

What You Are Building

You want a companion agent whose personality holds steady from one conversation to the next. A user who spent last week talking through a hard decision should meet the same disposition today: the companion that was cautious about a topic stays cautious, the one that grew warmer keeps that warmth, and none of it evaporates when the session ends and a new one begins.

The search-intent problem is specific. It is not "how do I write a good system prompt." It is: how do I make the companion's temperament a durable, evolving property of the agent rather than something reconstructed from scratch on every request. Anyone

building long-duration relational agents, journaling companions, coaching bots, or long-lived assistants runs into this. The personality feels real for one session and then quietly resets.

This guide walks through an architecture that solves that by carrying disposition as a first-class field on the agent object, updating it by a deterministic rule, and recording every change in the agent's lineage. It is disclosed in United States Patent Application 19/647,395. You build it yourself; there is no package to install.

Why the Obvious Approaches Fall Short

The common approaches all work, up to a point, and it helps to be precise about where the gap actually is.

System-prompt personas. You describe the personality in a system prompt and prepend it every turn. This gives a stable *description* of who the companion is, but it does not give a stable *state*. The prompt says "you are warm and cautious"; it does not carry how warm or how cautious the companion has become as a result of this particular relationship's history. The persona is static text, re-read identically each turn, not something that accumulates.

Summarized memory. You store a running summary of past sessions and feed it back in. This preserves facts and topics well. But disposition summarized as prose ("the user seems to have built trust") gets re-interpreted by the model each turn, and the interpretation drifts. There is no bounded, auditable value that says exactly where the companion's caution sits right now.

Re-inferring mood from the transcript. You let the model infer the current emotional tenor from recent messages. The application filing describes the structural weakness here directly: because autoregressive inference conditions each output on prior outputs, a strong recent affective signal can dominate subsequent generation and

perpetuate itself as narrative momentum, rather than being weighed against the agent's accumulated history. A single tense exchange can swing the whole personality, and the swing is not governed by anything.

The filing frames conventional systems as stateless inference engines that retain no persistent identity across time and therefore cannot maintain behavioral consistency across interactions. It also notes that existing affective agent models tend to treat emotion as a transient filter on plan selection or a scalar input to decision weighting, rather than as a persistent, independently tracked structural field. The gap is structural: disposition is not being stored as durable, bounded, auditable state on the agent.

The Architecture

The disclosed approach adds disposition to the agent as a dedicated field and updates it under a strict, auditable discipline. Every mechanism below traces to US Patent Application 19/647,395.

A persistent affective-state field on the agent. The filing introduces the affective state field as a structural field of the semantic agent schema, alongside fields such as intent, context, memory, policy, mutation descriptor, and lineage. It is described as a deterministic, policy-bounded data structure that encodes valence-weighted feedback derived from prior execution outcomes and observations. Critically, it is persisted with the agent across execution cycles, delegation events, and substrate migrations, so the modulation state is not lost when the agent is serialized for transport or moved between environments. That persistence is exactly what "across sessions" requires: the disposition travels with the agent object, not with the transient prompt.

The filing is careful to say this does not encode emotion in the phenomenological sense. It encodes a structured modulation vector that shapes how the agent weighs alternatives, tolerates ambiguity, persists under partial failure, and escalates under pressure.

Named control fields, each a bounded tuple. Rather than a single "mood" scalar, the field is organized as a structured modulation layer of named control fields, each a distinct, measurable axis. The filing enumerates axes including uncertainty sensitivity, ambiguity tolerance, novelty appetite, persistence-under-partial-failure, escalation-under-time-pressure, risk sensitivity, and cooperation disposition. Each named field is represented as a tuple: a current magnitude within a defined range, a decay rate governing return toward baseline, a policy-defined ceiling and floor, and a timestamp of the most recent update.

Deterministic updates from structured observations. Updates are deterministic: given the same agent state, the same environmental inputs, and the same policy configuration, the update function produces the same output. It consumes structured observations derived from execution, including repeated failure patterns, competing objectives, time pressure, novelty exposure, model-reported uncertainty, and execution success patterns. Each dimension is updated by its own rule, subject to policy bounds, and the update is recorded in lineage as a state mutation event with input observations and resulting change preserved.

Policy bounds as hard constraints. Every update is a policy-bounded mutation. The filing specifies range bounds (values clamped to ceiling/floor), rate limits (a maximum change per cycle so even a catastrophic event cannot cause a discontinuous jump), admissible triggers (only certain observation types may move a given field), update authority (only authorized channels may write), and decay governance. The described sequence is: verify the trigger is admissible, compute the raw update, clamp to the rate limit, apply, clamp to range bounds, then record the whole transaction in lineage.

Decay, hysteresis, and stabilization. Each field is governed by a decay curve returning it toward a baseline in the absence of reinforcement; the filing gives an exponential form with a configurable time constant, and notes different axes may decay at different rates (uncertainty sensitivity fast, learned persistence slow). It describes

semantic hysteresis via asymmetric update rules, with negative-valence updates applying faster than positive ones, producing a built-in caution bias. It also describes entropy-governed stabilization that damps oscillation by increasing the effective decay constant when a field alternates rapidly.

What the disposition actually changes. The field modulates enumerated deliberation targets, not open-ended behavior: promotion thresholds, search breadth, branch growth rates, decay rates for unpromoted candidates, escalation thresholds, persistence parameters, delegation routing preferences, and mutation acceptance thresholds. The filing is explicit that affect does not create new capabilities or bypass policy; it modulates the quantitative parameters of already-authorized processes.

The companion layering. For companions specifically, the filing describes a three-layer personality architecture: a core trait layer (stable temperament, communication style, humor, empathy patterns, value orientations, authored at instantiation and changed only through explicit, lineage-recorded revision events); a dynamic preference layer (accumulated, governed preferences from interaction history); and an adaptive affect layer, which is the affective-state field above and is the most volatile of the three. It also describes an emotional state tracker with an affect log and a longitudinal consistency map, so a companion that expressed concern in a prior session references or builds on it rather than presenting a fresh emotional slate, with longitudinal coherence enforced by the validation engine.

How to Approach the Build

You are implementing this yourself. These steps follow the disclosed design; the sketches below are illustrative, not a drop-in module.

1. Define the agent object with disposition as a first-class field. Store the companion's state as a persisted object that includes an affective-state field. Do not keep disposition only in the prompt or only in a summary. Following the layering,

separate stable core traits from the volatile adaptive-affect layer so a bad day never rewrites who the companion fundamentally is.

2. Model each axis as a bounded tuple. Pick the named control fields relevant to your companion (for a relational companion, cooperation disposition, uncertainty sensitivity, and novelty appetite are natural). Represent each faithfully to the filing:

```
# illustrative, not a working library
axis = {
  value: 0.4,          # current magnitude within range
  floor: 0.0, ceil: 0.8, # policy-defined bounds
  decay_tau: 3600,    # time constant toward baseline
  baseline: 0.3,
  updated_at: <timestamp>,
}
```

3. Write a deterministic update function. Turn interaction events into structured observations, then apply the multi-stage discipline: check the trigger is admissible for that axis, compute the raw update, clamp to the rate limit, apply, clamp to range bounds. Keep it deterministic so the same history yields the same disposition and the evolution is reproducible.

4. Apply decay on read, not just on write. When you load the agent for a new session, advance each axis along its decay curve for the elapsed time before using it. This is what makes "across sessions" behave sensibly: a spike from last night has partly relaxed by morning, at a rate you set per axis.

5. Record every mutation in lineage. Persist the observation, the raw and clamped update, the prior value, and the resulting value. This gives you an auditable trail of why the companion feels the way it does today, and it is what the filing relies on for reproducibility and governance.

6. Feed disposition into deliberation, not into permissions. Use the axis values to modulate the enumerated targets: how many candidate responses you consider, how high the bar is to commit, when the companion escalates or defers. Keep this strictly separate from what the companion is *allowed* to do. The filing enforces that affect is not an input to the governance gate.

7. Add stabilization and a volatility guard. Implement hysteresis (faster negative than positive updates) and entropy-governed damping so noisy sessions do not send the personality oscillating. The filing also describes an emotional-quarantine mode that restricts an agent whose affect is volatile; a lighter version of this is worth having so a single hostile session cannot destabilize the companion.

What This Does Not Give You

This is an architecture, not a drop-in library, and not a benchmarked or productized system. You implement every component yourself. Nothing here has been measured for latency, quality, or user outcomes in this guide, and no performance numbers are claimed because the filing states none.

It does not model or diagnose human emotion, and it does not make the companion "conscious" or genuinely emotional. The filing is explicit that the field is a structured modulation vector, not phenomenological emotion. It also will not, by design, let disposition expand what the companion may do: affect modulates how the agent deliberates, never whether it is authorized to act, so do not use it as an access-control mechanism.

It presumes you already have durable per-user agent storage and a way to derive structured observations from interactions. If your product genuinely wants a fresh personality every session, you do not need this. And the specific companion features the

filing describes around this core, such as narrative unlocking and attachment-based progression, are additional mechanisms beyond the disposition-persistence problem this guide addresses.

Disclosure Scope

The architecture described here is disclosed in United States Patent Application 19/647,395. This guide is educational: it explains an approach so a developer can understand and build it themselves. It is not a warranty, a specification of a shipping product, or an offer of software, and it does not grant any license. Every mechanism described is traced to that filing; where the filing is silent, this guide makes no claim.

Affective State (</affective-state>)

[All 40 steps → \(/inventive-steps\)](/inventive-steps)

Emotion as a computational primitive, not a simulation.

[Chapter 2 \(/patents/19-647395/chapters/affect\)](/patents/19-647395/chapters/affect)

PRIMARY TECHNICAL DISCLOSURE

- [Affective State as a Deterministic Control Primitive for Semantic Agents \(/articles/affective-state-as-a-deterministic-control-primitive-for-semantic-agents\)](/articles/affective-state-as-a-deterministic-control-primitive-for-semantic-agents)

SECONDARY TECHNICAL

- [Affective State as the Seventh Structural Field \(/articles/affective-state/seventh-canonical-field\)](/articles/affective-state/seventh-canonical-field)
- [Named Control Field Modulation Architecture \(/articles/affective-state/named-control-fields\)](/articles/affective-state/named-control-fields)
- [Affect-Modulated Promotion Thresholds \(/articles/affective-state/promotion-thresholds\)](/articles/affective-state/promotion-thresholds)
- [Deterministic Affect Encoding and Update Mechanics \(/articles/affective-state/deterministic-encoding\)](/articles/affective-state/deterministic-encoding)
- [Emotional Decay Curves With Hysteresis \(/articles/affective-state/decay-curves\)](/articles/affective-state/decay-curves)
- [Entropy-Governed Valence Stabilization \(/articles/affective-state/valence-stabilization\)](/articles/affective-state/valence-stabilization)
- [Affective Inheritance in Delegation Chains \(/articles/affective-state/inheritance-chains\)](/articles/affective-state/inheritance-chains)

- [Emotional Quarantine and Volatility Management \(/articles/affective-state/emotional-quarantine\)](/articles/affective-state/emotional-quarantine)
- [Affect-Modulated Trust Slope Validation \(/articles/affective-state/trust-slope-modulation\)](/articles/affective-state/trust-slope-modulation)
- [Biological Signal-to-Affective Coupling \(/articles/affective-state/biological-coupling\)](/articles/affective-state/biological-coupling)
- [Affective Contagion in Multi-Agent Systems \(/articles/affective-state/affective-contagion\)](/articles/affective-state/affective-contagion)
- [Affect-Modulated Discovery Traversal \(/articles/affective-state/discovery-traversal\)](/articles/affective-state/discovery-traversal)
- [Affect-Governance Separation \(/articles/affective-state/governance-separation\)](/articles/affective-state/governance-separation)
- [Policy-Bounded Affective Updates \(/articles/affective-state/policy-bounded-updates\)](/articles/affective-state/policy-bounded-updates)
- [Affect as Cross-Primitive Input \(/articles/affective-state/cross-primitive-input\)](/articles/affective-state/cross-primitive-input)
- [Affect-Modulated Inference Integration \(/articles/affective-state/inference-integration\)](/articles/affective-state/inference-integration)
- [Substrate-Agnostic Affect Deployment \(/articles/affective-state/substrate-deployment\)](/articles/affective-state/substrate-deployment)
- [Pseudonymous Emotional Operation \(/articles/affective-state/pseudonymous-operation\)](/articles/affective-state/pseudonymous-operation)
- [Temporal Cognition Field \(/articles/affective-state/temporal-cognition\)](/articles/affective-state/temporal-cognition)

APPLICATIONS · GENERAL

- [How to Build a Companion AI That Keeps Emotional Consistency Across Sessions \(/articles/affective-state/companion-consistency\)](/articles/affective-state/companion-consistency)
- [Therapeutic AI Agents: Governed Emotional State for Mental Health Software Under Clinical Constraints \(/articles/affective-state/therapeutic-affect\)](/articles/affective-state/therapeutic-affect)
- [Persistent Affective State for Customer Service AI Agents: Beyond Per-Message Sentiment \(/articles/affective-state/customer-service-agents\)](/articles/affective-state/customer-service-agents)
- [AI Companion Agents for Elderly Care: Emotional Continuity and Mood Monitoring with Deterministic Affective State \(/articles/affective-state/elderly-care-companions\)](/articles/affective-state/elderly-care-companions)
- [Crisis Response AI Agents That Stay Calm: Governed Affective State for Emergency Operations \(/articles/affective-state/crisis-response-agents\)](/articles/affective-state/crisis-response-agents)
- [Auditable AI Negotiation Agents with Deterministic Affective State \(/articles/affective-state/negotiation-agents\)](/articles/affective-state/negotiation-agents)
- [Emotionally Adaptive AI Tutoring: Detecting Student Frustration and Disengagement Before They Happen \(/articles/affective-state/educational-tutoring\)](/articles/affective-state/educational-tutoring)
- [Governed Affective State for Compliant HR and Recruitment AI Agents \(/articles/affective-state/hr-recruitment-agents\)](/articles/affective-state/hr-recruitment-agents)

APPLICATIONS · SPECIFIC

- [Replika Alternative: Governed Affective State vs Reconstructed Emotion \(/articles/affective-state/replika\)](/articles/affective-state/replika)

- [Character.ai Alternative: Persistent Affective State vs Prompt-Defined Personality \(/articles/affective-state/character-ai\)](/articles/affective-state/character-ai)
- [Woebot vs Deterministic Affective State: Why a Therapy Chatbot Has No Persistent Modulation Field \(/articles/affective-state/woebot\)](/articles/affective-state/woebot)
- [Elomia vs. a Governed Affective State Field: What Persists Between Sessions? \(/articles/affective-state/elomia\)](/articles/affective-state/elomia)
- [Hume AI Alternative: Governed Affective State Beyond Emotion Measurement \(/articles/affective-state/hume-ai\)](/articles/affective-state/hume-ai)
- [Affectiva Alternative: Reading Human Emotion vs. Governing an Agent's Own Affective State \(/articles/affective-state/affectiva\)](/articles/affective-state/affectiva)
- [Cogito vs Governed Agent Affect: Reading Human Emotion Is Not an Agent's Internal State \(/articles/affective-state/cogito\)](/articles/affective-state/cogito)
- [Beyond Verbal vs Governed Affective State: Reading Emotion Versus Modulating Cognition \(/articles/affective-state/beyond-verbal\)](/articles/affective-state/beyond-verbal)
- [EmotiBit vs Governed Affective Modulation: Physiology Without a Policy-Bounded Affect Field \(/articles/affective-state/emotibit\)](/articles/affective-state/emotibit)
- [Realeyes Alternative: Governed Affective State vs Per-Session Emotion Measurement \(/articles/affective-state/realeyes\)](/articles/affective-state/realeyes)

[Affective State overview → \(/affective-state\)](/affective-state)