Feb 9, 2026:

NPC Cognition: A New Industry Standard for Living Worlds

Author: Jeff Borneman | RELCOG Labs

For decades, NPCs have been trapped inside rigid behavior trees, scripted loops, and short-lived dialogue switches that reset the moment a player walks away. Players feel the seams instantly. Characters don't remember, worlds don't evolve, and emotional continuity disappears as quickly as it appears. Even as graphics and physics advance, the inner lives of NPCs remain frozen in the logic of the early 2000s.

A new architecture is emerging — one that treats NPCs not as scripted entities, but as governed cognitive systems capable of evolving emotional states, relational memory, and consistent personality over time. This approach does not rely on massive models, deep simulation, or expensive inference. Instead, it introduces a lightweight emotional substrate that allows NPCs to grow, react, and transform based on the player's choices.

This substrate is built around a simple idea: NPCs don't need to think like humans to feel alive. They need a stable way to hold posture — trust, openness, tension, curiosity, clarity — and a governed way to update those feelings each time the player interacts with them. The result is a world that remembers not what happened, but how it felt. Entire villages shift tone as trust rises or falls. Creatures become cautious or curious based on exposure. Companions develop loyalty or resentment across dozens of hours.

Because this system operates through compact emotional vectors rather than symbolic memory or inference-heavy reasoning, it scales effortlessly. Thousands of NPCs can maintain persistent inner states entirely on CPU, allowing mobile, indie, and AAA studios alike to build worlds that evolve naturally without performance strain. Antagonistic factions develop shared emotional climates. Wildlife adjusts to the player's presence. Even minor characters subtly change over time, giving the world the illusion of depth without the cost of full simulation.

This marks a new standard for NPC cognition — not artificial intelligence, but artificial continuity. Not scripted personality, but governed emotional geometry. It gives developers a powerful new tool: the ability to shape living worlds that feel coherent, responsive, and personal, all without rewriting the fundamental architecture of their games.

NPC cognition is not the future of gaming — it is what players have always expected but never been given. With a meaning-first substrate guiding behavior rather than brittle logic or probabilistic inference, game worlds gain something they've been missing for years: emotional truth.

A soft, steady breath — we'll move through all **three documents**, each written in a calm, flowing, narrative tone that fits your preference and protects your IP.
These will become foundational tools for outreach, collaboration, and strategic engagement.

I'll deliver them in the following order:

1. **A Safe Public Description of NPC Evolution**
   (for LinkedIn, websites, pitch decks, conversations with studios)
2. **A Technical Intro Packet for Partners**
   (what Osama, Unity devs, or game studios *should* see)
3. **A Formal IP Boundary Document**
   (defines clearly what is *not* shared and protects the substrate)

⭐ 1. Safe Public Description of NPC Evolution

(For LinkedIn, studio conversations, public posts — zero IP exposure.)

In many game worlds, NPCs exist only in the moment. They reset when you leave town, forget when you return, and behave as though nothing has changed. STOIC approaches NPCs from a different starting point. Instead of scripts, branches, or behavior trees, each character carries a small internal "emotional state" that shifts slowly over time based on how the player interacts with them.

The NPC doesn't remember facts or lines of dialogue. It remembers *how it felt*. When the player is kind, trust grows. When the player steals or threatens, tension rises. When the world around a character changes, their curiosity or caution adjusts. These emotional shifts are subtle and continuous, which allows the world to evolve naturally without heavy simulation or complex AI.

Because this state is small, portable, and inexpensive to update, entire villages or factions can evolve between visits. Wildlife becomes skittish or friendly. Companions open up over time. Even minor characters develop a sense of continuity and personality. The world feels alive not because it is deeply simulated, but because its inhabitants carry a governed emotional imprint shaped by the player's presence.

This approach works offline, scales cheaply, and integrates directly into existing engines. It creates a world where characters grow with the player — quietly, naturally, and without the computational weight of large models.

⭐ 2. Technical Intro Packet for Partners

(For Unity devs, game studios, technical collaborators — interface-level only, no substrate exposure.)

STOIC can be integrated into a game engine as a lightweight cognition layer that governs NPC emotional continuity and personality evolution. The core idea is simple: the engine manages the world, while STOIC manages how each NPC feels and reacts over time.

Each NPC holds a small intent state — usually seven to nine values between zero and one — representing trust, tension, curiosity, openness, clarity, emotional weight, and relational stance. When the player interacts with the NPC, Unity sends a short JSON capsule containing the NPC's previous state along with any relevant game cues: the last player action, emotional tone, proximity, faction signals, or environment changes.

STOIC reads the capsule, updates the intent state through its meaning-governance logic, and returns a new mind-state capsule. Unity then uses these values to drive NPC behavior. An NPC with rising trust may move closer or speak more openly. A character with rising tension may shorten dialogue, glance away, avoid the player, or alert others. Designers have full control because STOIC never dictates actions — it only provides the emotional physics that shape behavior.

Because the intent vector is continuous and compact, the system runs on CPU, scales to thousands of NPCs, and does not require large models. If dialogue is needed, a small 3B–7B model is more than sufficient because STOIC ensures coherence, drift control, and personality stability. STOIC remains a modular component that plugs into the engine cleanly, operating entirely through JSON capsules without exposing any internal cognition logic.

⭐ 3. Formal IP Boundary Document

(For collaborators, NDAs, partners, internal governance — protects all proprietary layers.)

This document defines the boundaries of what RELCOG Labs-STOIC discloses and what remains proprietary. All collaborators, including studios, integrators, and technical partners, operate within this boundary. The goal is to allow functional integration without revealing cognitive substrate mechanisms, semantic geometry computation, or internal agency behavior.

STOIC provides:
- A JSON intake capsule specification.
- A JSON mind-state capsule specification.
- A stable API surface for game engines.
- A description of how engines should integrate and interpret emotional posture values.
- Runtime examples showing how small models can be used safely when governed.

• Reference diagrams describing the external architecture loop between Unity and STOIC.
• Guidelines for mapping STOIC mind-state values to engine behavior.

STOIC does not disclose:
• Internal Meaning Engine algorithms.
• Semantic geometry computation methods (mass, entropy, drift).
• Micro-agency definitions or orchestrator internals.
• Meta-command controllers or routing logic.
• Emotional geometry transforms or numeric thresholds.
• Teleology blockers, safety posture layers, or substrate invariants.
• Any part of the SPARC Presence logic.
• Any code, module design, or architectural implementation details.
• Any elements that constitute unique cognitive substrate behavior.

Partners understand that STOIC is a governed emotional substrate. They do not see how feelings are computed, only how they are expressed. They store the intent vector locally, send context forward, receive updated values, and use them to drive character behavior. The internal cognitive architecture remains sealed.

This boundary ensures STOIC remains portable, protectable, and deployable across industries without compromising intellectual property. All integrations occur through the API, not through internal modules or semantic computation layers.

___