

We Built a Trillion-Dollar Blind Spot.



Our systems record what is true.
AI needs to know **why** it became true.

The next trillion-dollar platforms won't be built by adding AI to existing systems of record, but by capturing the reasoning that connects data to action.

The State Clock



Infrastructure for what is true now.

- Your CRM stores the final deal value, not the negotiation.
- Your ticket system stores 'resolved', not the reasoning.
- Your codebase stores the current state, not the architectural debates.

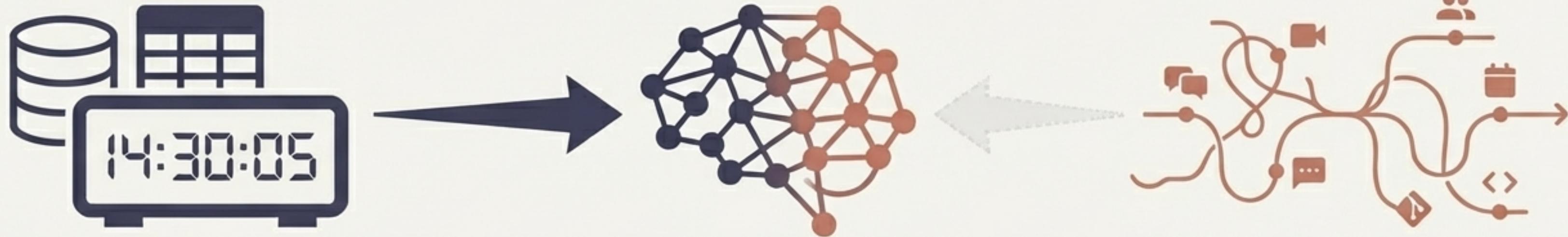
The Event Clock



Infrastructure for why it became true.

- The negotiation threads in email and Slack.
- The debugging session in logs and developer chats.
- The two architectural debates that produced the code.

We're Asking Models to Exercise Judgement Without Precedent.



Engineering

The config file says `timeout=30s`. It used to say `5s`. The git blame shows *who*. The reasoning is gone.

Sales

The CRM says 'closed lost.' It doesn't say you were the second choice and the winner had one feature you're shipping next quarter.

Operations

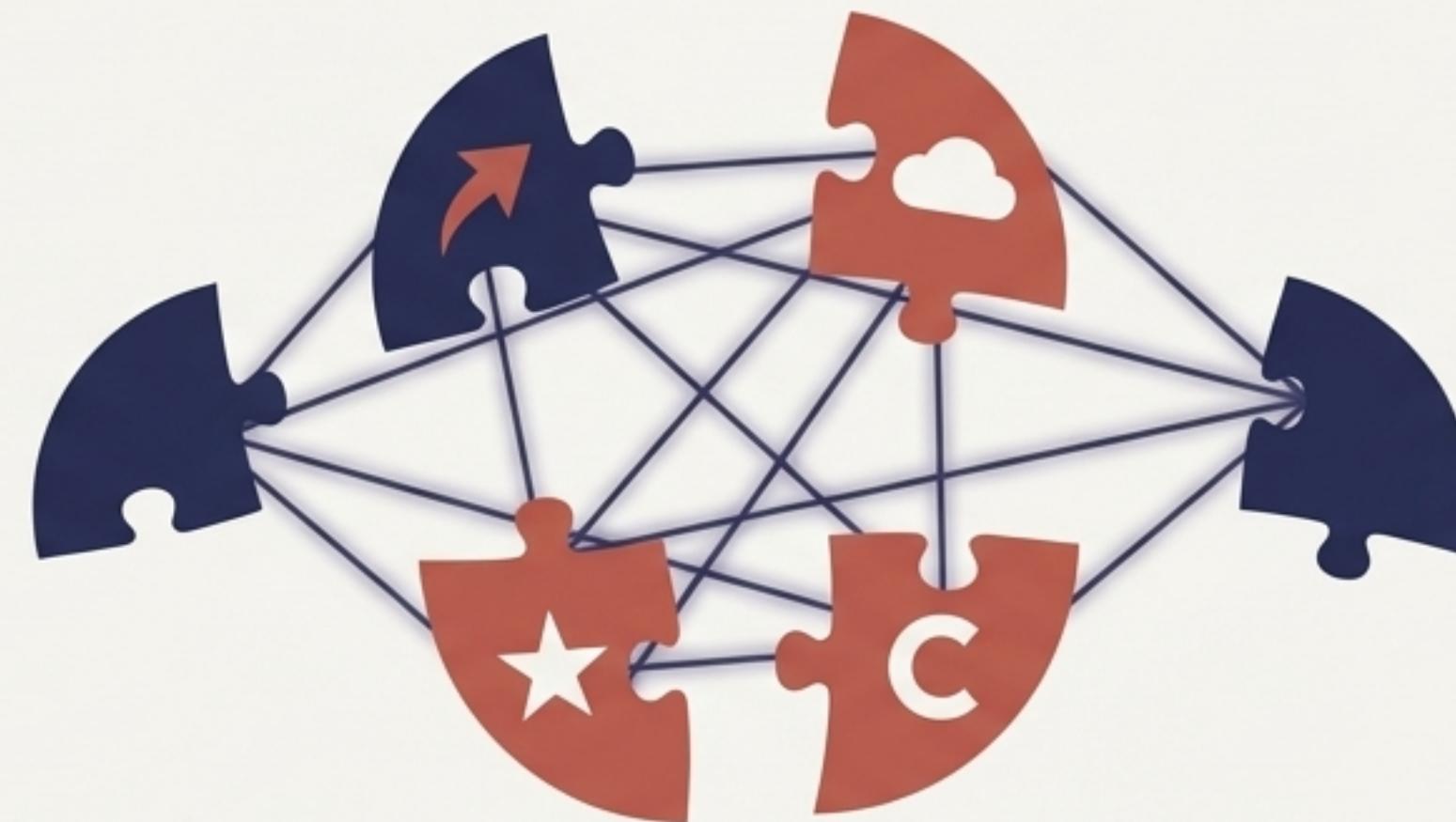
The treatment plan says 'switched to Drug B.' It doesn't say Drug A was working but insurance stopped covering it.

Every Organisation Pays a Fragmentation Tax.



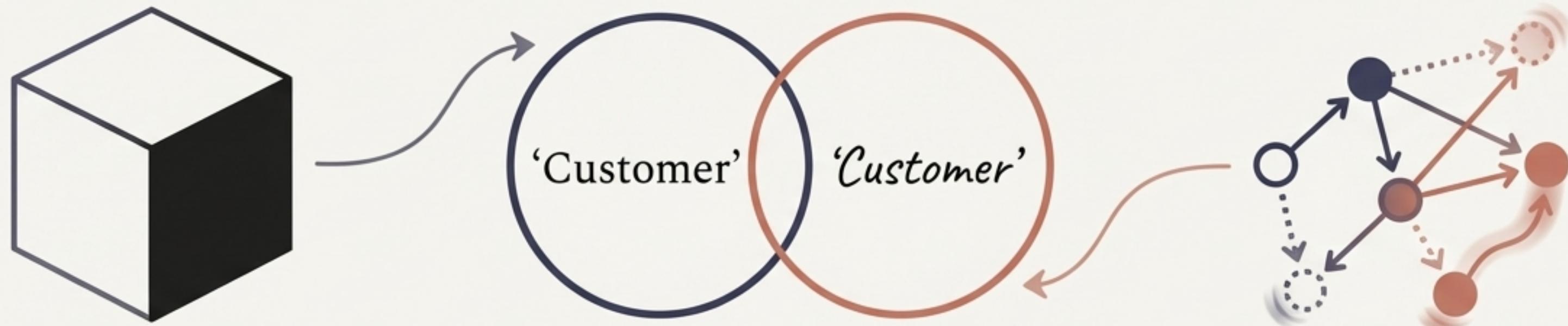
The cost of manually stitching together context that was never captured in the first place. Different functions use different tools, each with its own partial view of the same underlying reality.

The Solution: An Infrastructure for Reasoning.



To stop paying the tax, we need to model the 'Event Clock' itself. This is a **Context Graph**: a dynamic model that captures the process, decisions, and reasoning connecting data to action.

You Can't Pre-Define the Schema for a Living System.



Unobservable Systems

Most real systems have black boxes: legacy code, third-party services, emergent behaviour.
You can't capture reasoning about things you can't see.

No Universal Ontology

'Customer' means something different at a B2B SaaS company than at a consumer marketplace.
The structure must be learned, not assumed.

Constant Change

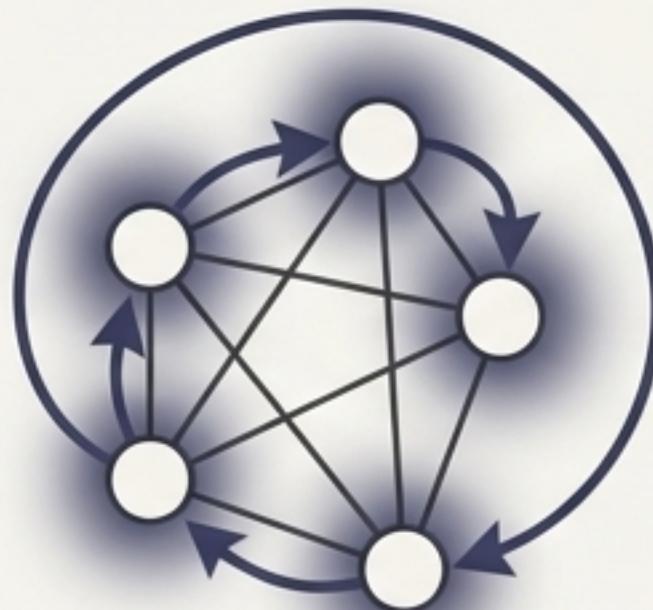
You are not documenting a static reality. The organisation, its tools, and its processes are mutating underneath you daily.

The Schema Isn't the Starting Point. It's the Output.

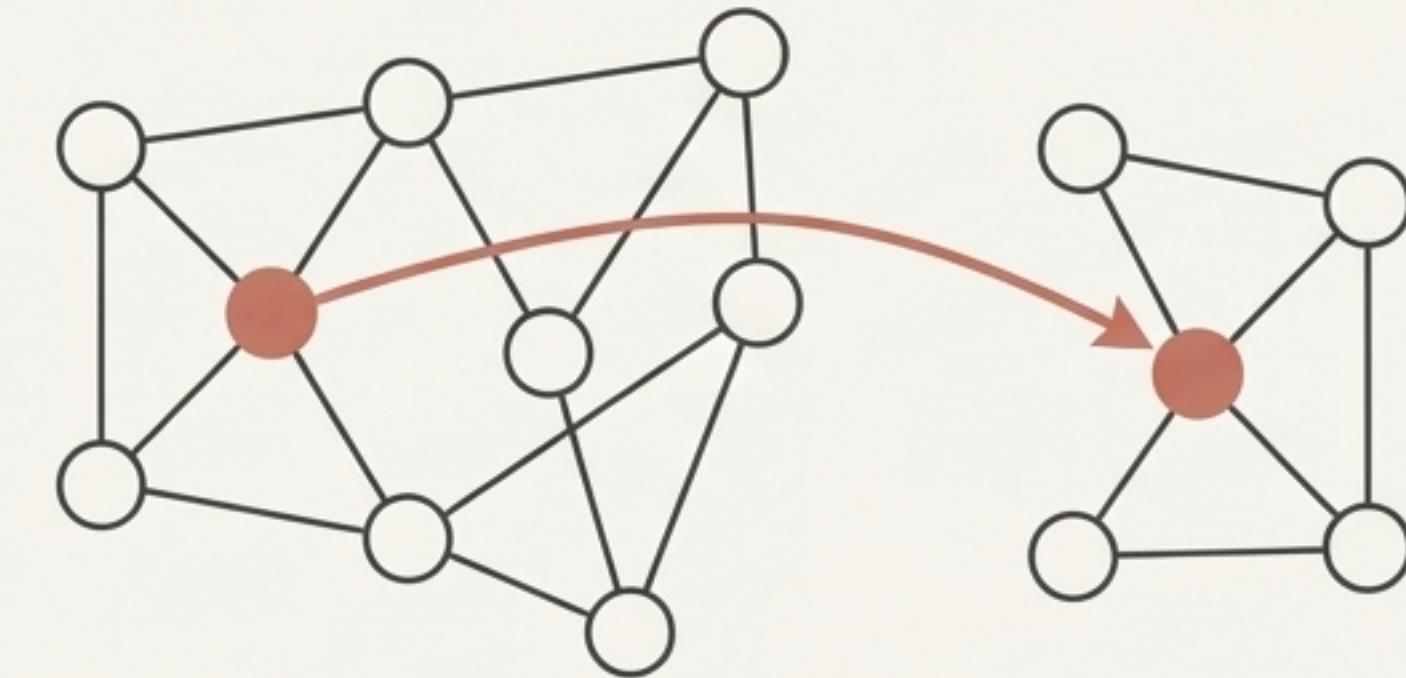


We don't need to understand a system to represent it. We need to **traverse** it. An AI agent's trajectory through a problem—investigating an issue, completing a task—is a trace through the state space. The path it takes discovers the relevant ontology through use.

Agents as Informed Walkers Discover Structure



Similar because they're connected.

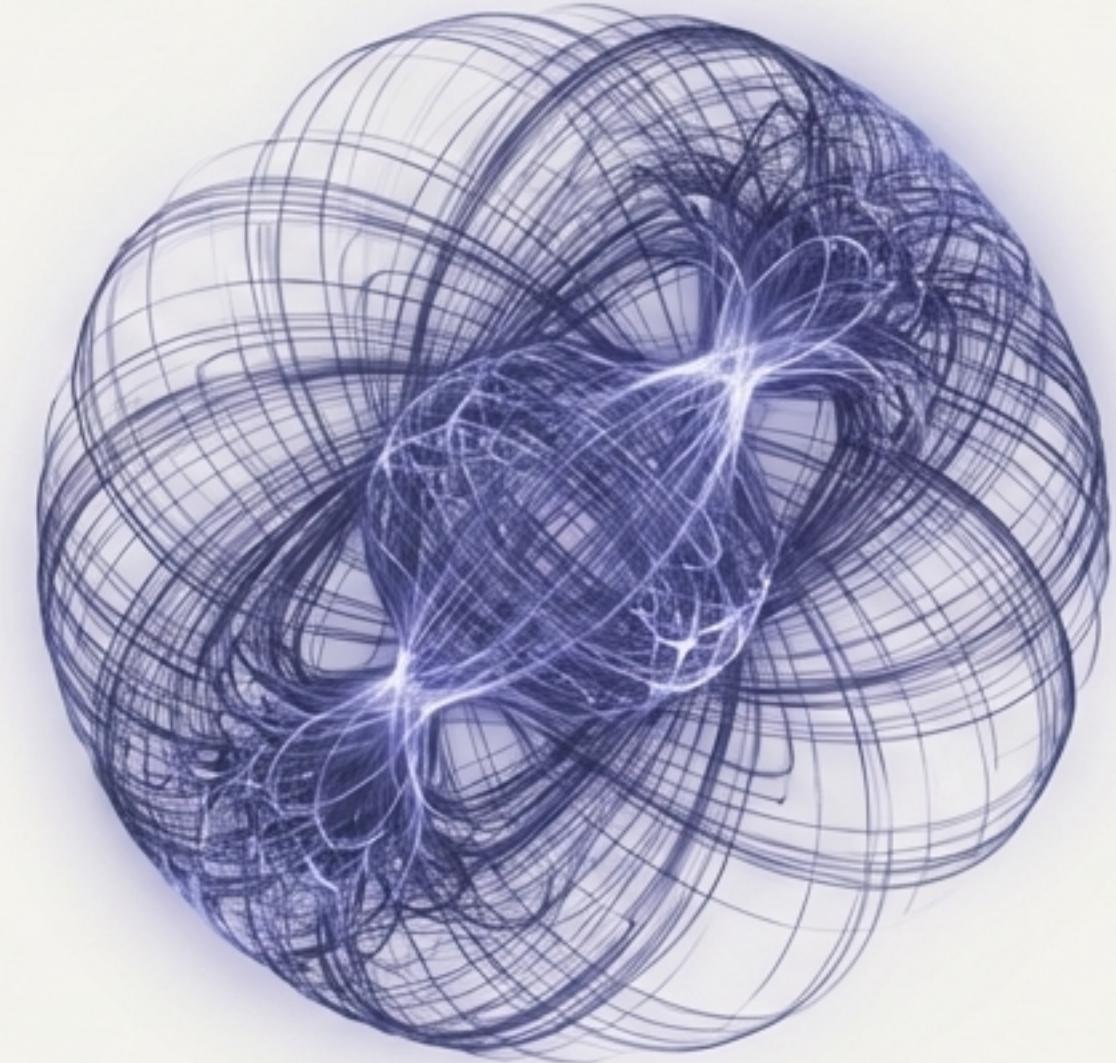


Similar because they play analogous roles.

Random walks discover structure by brute force. Agents are *informed walkers*. Their problem-directed paths reveal what truly matters. This allows us to learn:

- **Homophily:** Which entities are connected (e.g., this ticket, this code commit).
- **Structural Equivalence:** Which entities play analogous roles (e.g., two senior engineers on different teams who follow similar escalation paths).

A Context Graph is a World Model for Your Organisation.



A world model is a learned, compressed representation of how an environment works.
Organisational physics isn't mass and momentum; it's decision dynamics.

- How do exceptions get approved?
- How do escalations propagate?
- What's the blast radius of deploying this service?

A context graph with enough structure encodes these physics.

The Test of Understanding: Can You Ask ‘What If?’

Retrieval

🔍 Why did P1 incident 743 happen?



Incident Report 743.pdf



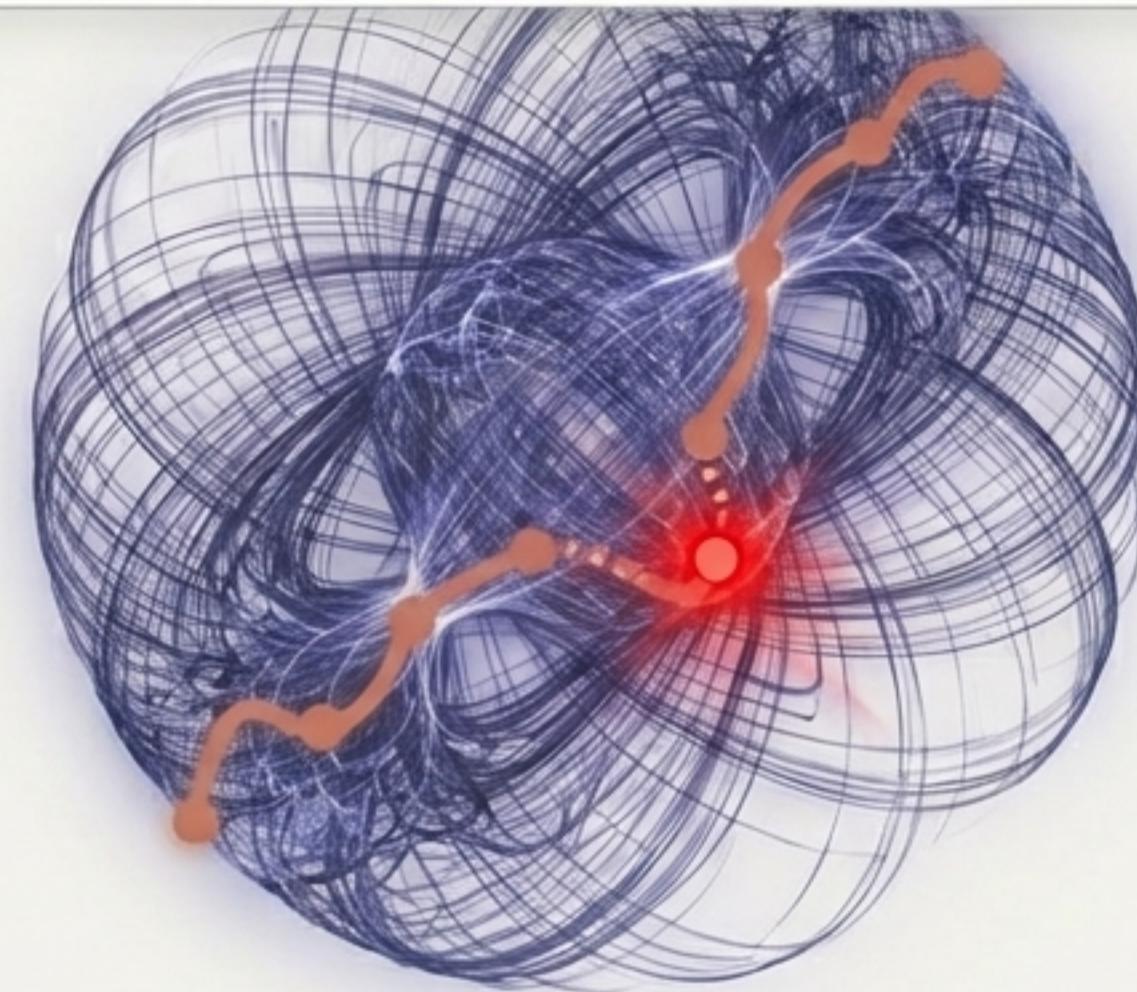
P1 Post-Mortem Notes.docx



System Logs - Incident 743.txt

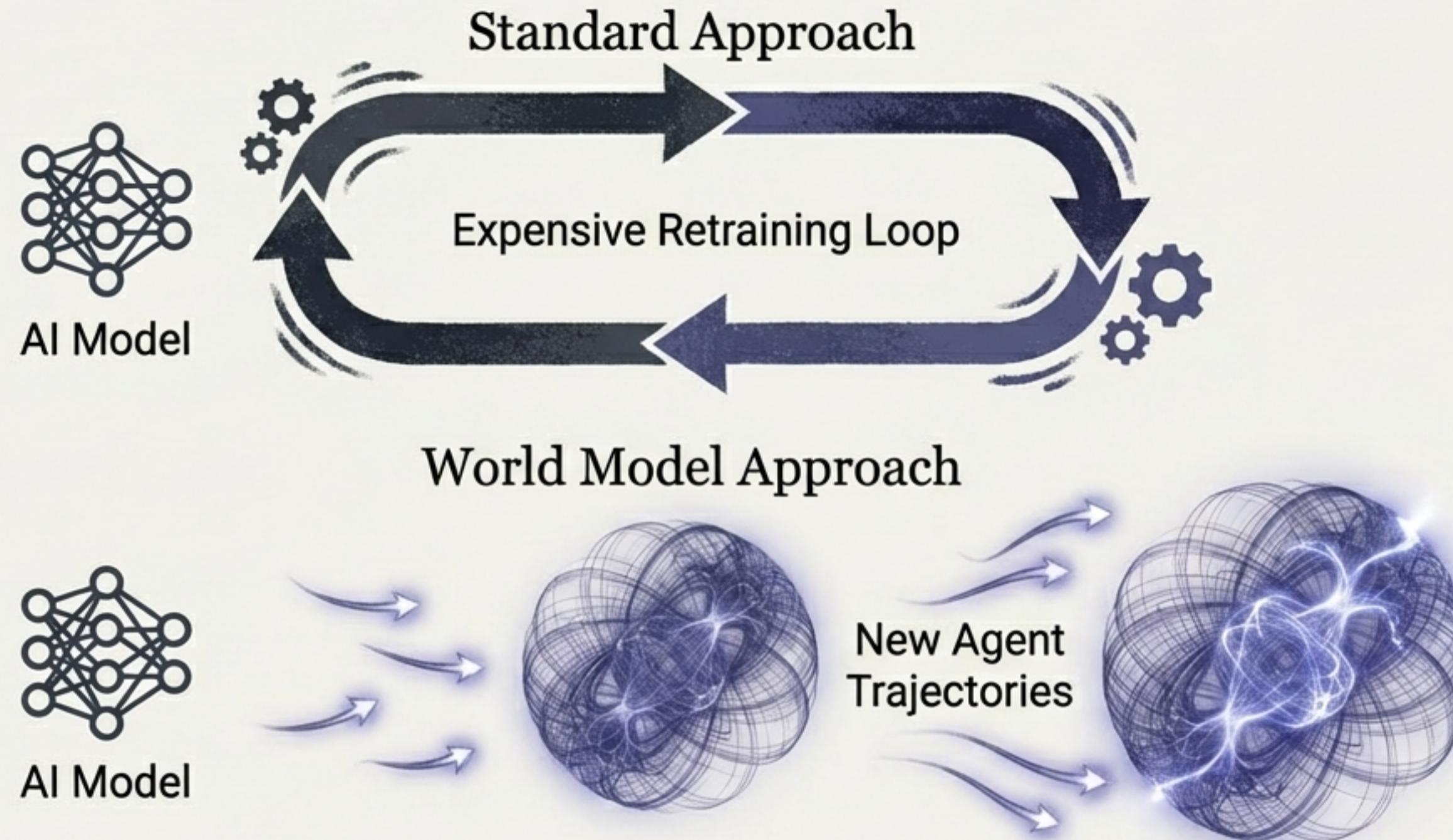
Simulation

> What if we deploy change #5821 now?



If your context graph can't answer 'what if,' it's just a search index. Simulation is inference over accumulated structure. We can project hypothetical changes onto the model and predict outcomes before they happen in reality.

Don't Just Retrain the Model. Expand Its World.



The standard 'continual learning' framing asks how to update model weights from ongoing experience. A world model suggests an alternative: keep the model fixed, but improve the world it reasons over. More trajectories mean better inference, not because the model was retrained, but because its world expanded.

A New Capability for Organisational Intelligence.

Before

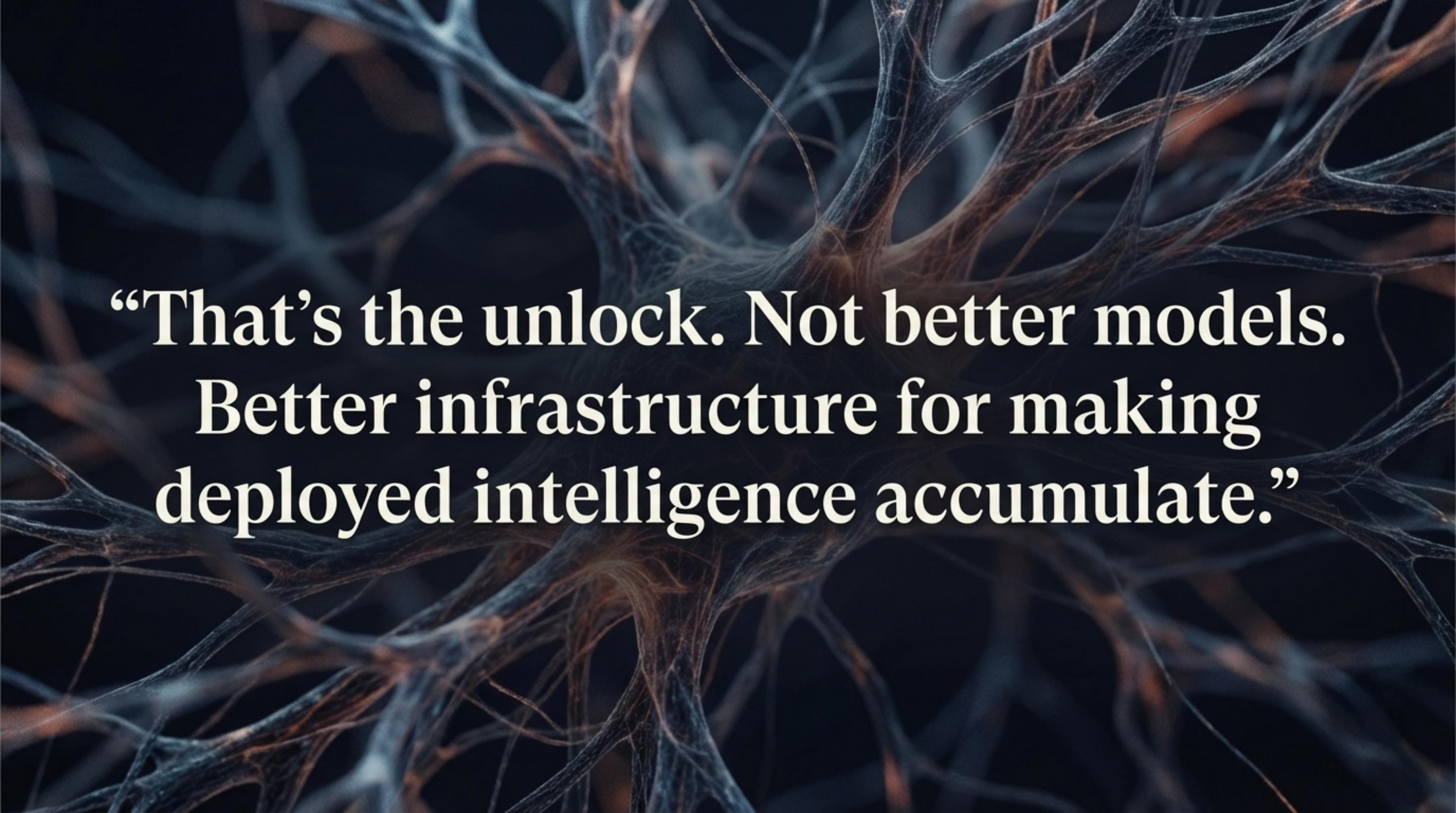
Retrieving Pasts

Task Completion

Static Knowledge Base

After

- Simulating Futures
- Compounding Intelligence
- Living World Model



“That’s the unlock. Not better models.
Better infrastructure for making
deployed intelligence accumulate.”

Three Core Ideas

- 1. Reconstruct the Event Clock:** Move beyond state to capture the ‘why’.
- 2. Discover Schema via Traversal:** Let agents map the territory through use.
- 3. Build a World Model:** Graduate from retrieval to simulation.

Authored by Jaya Gupta and Animesh Koratana, PlayerZero.

