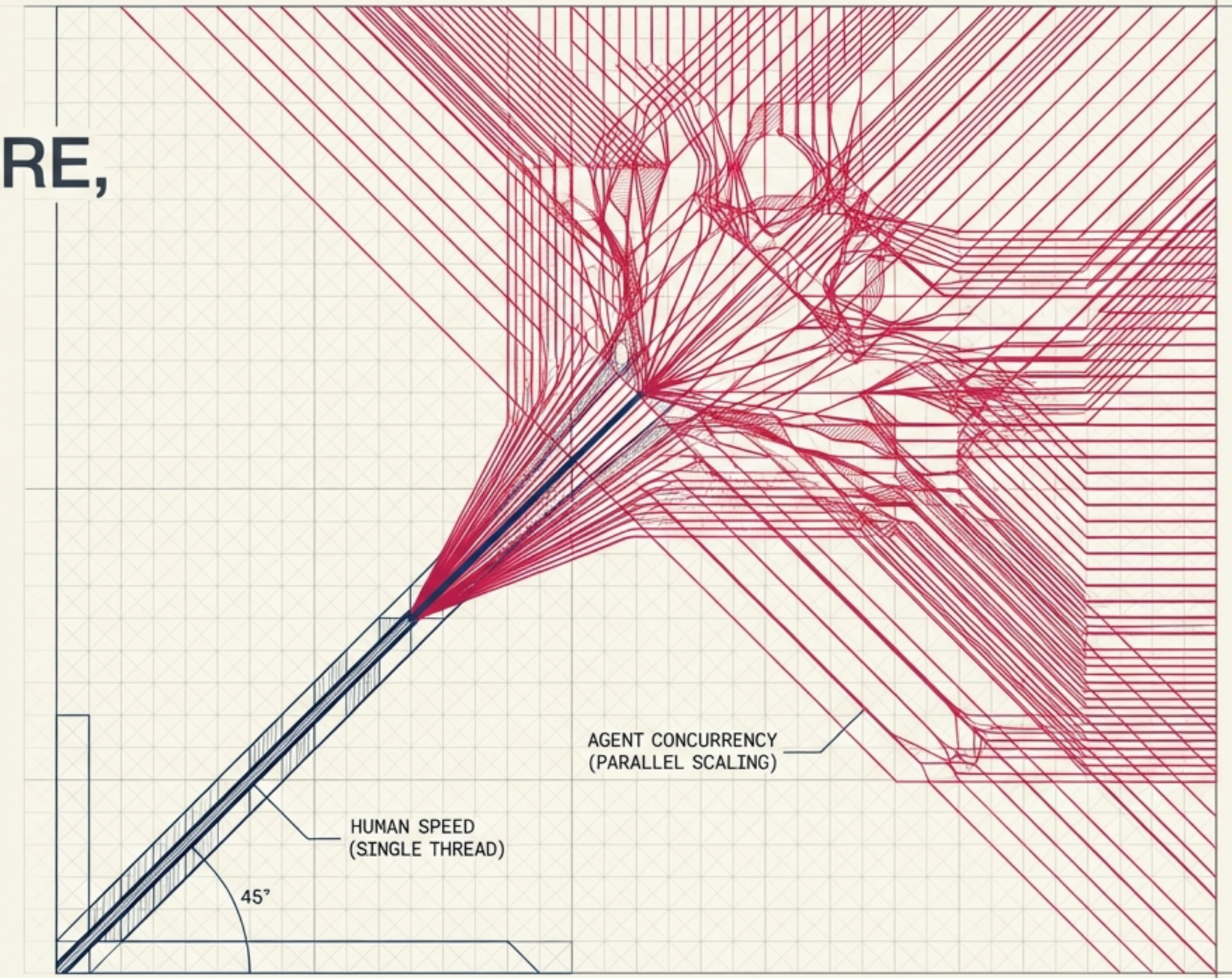
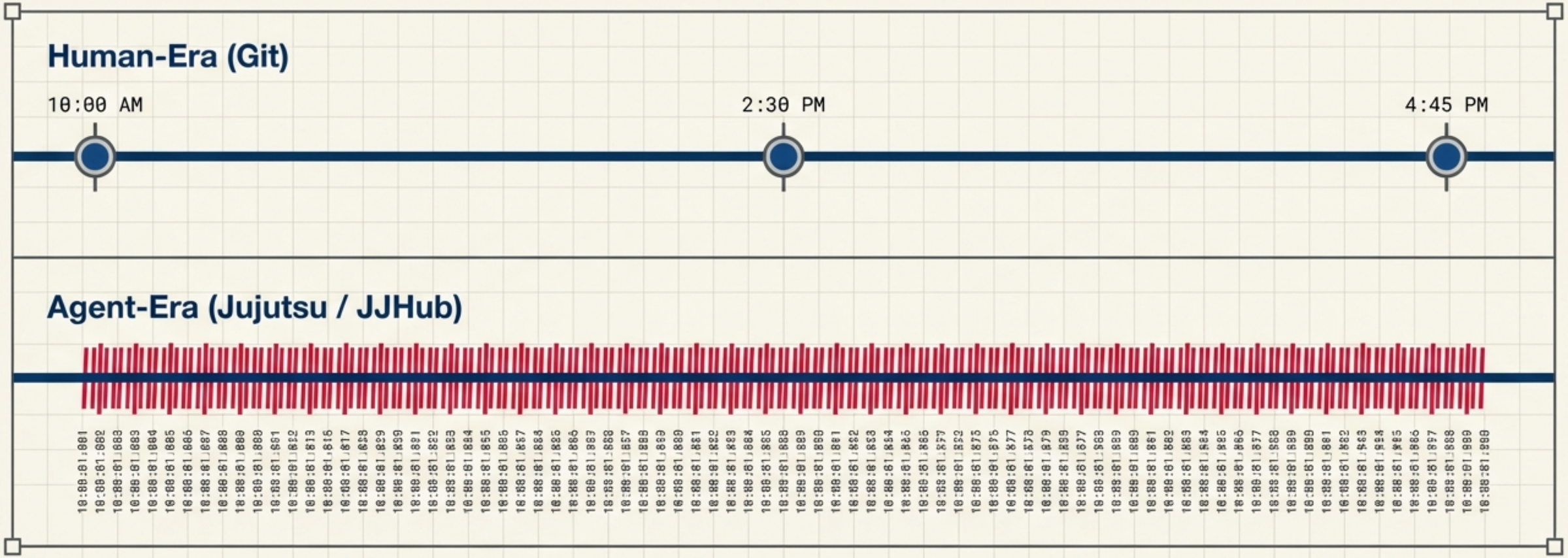
HUMAN-ERA INFRASTRUCTURE, AGENT-ERA WORKFLOWS

A Diagnostic Report:
Rebuilding the software stack
for autonomous systems.

Based on the architectural field
notes of Thor Henning Hetland.





THE SYMPTOM

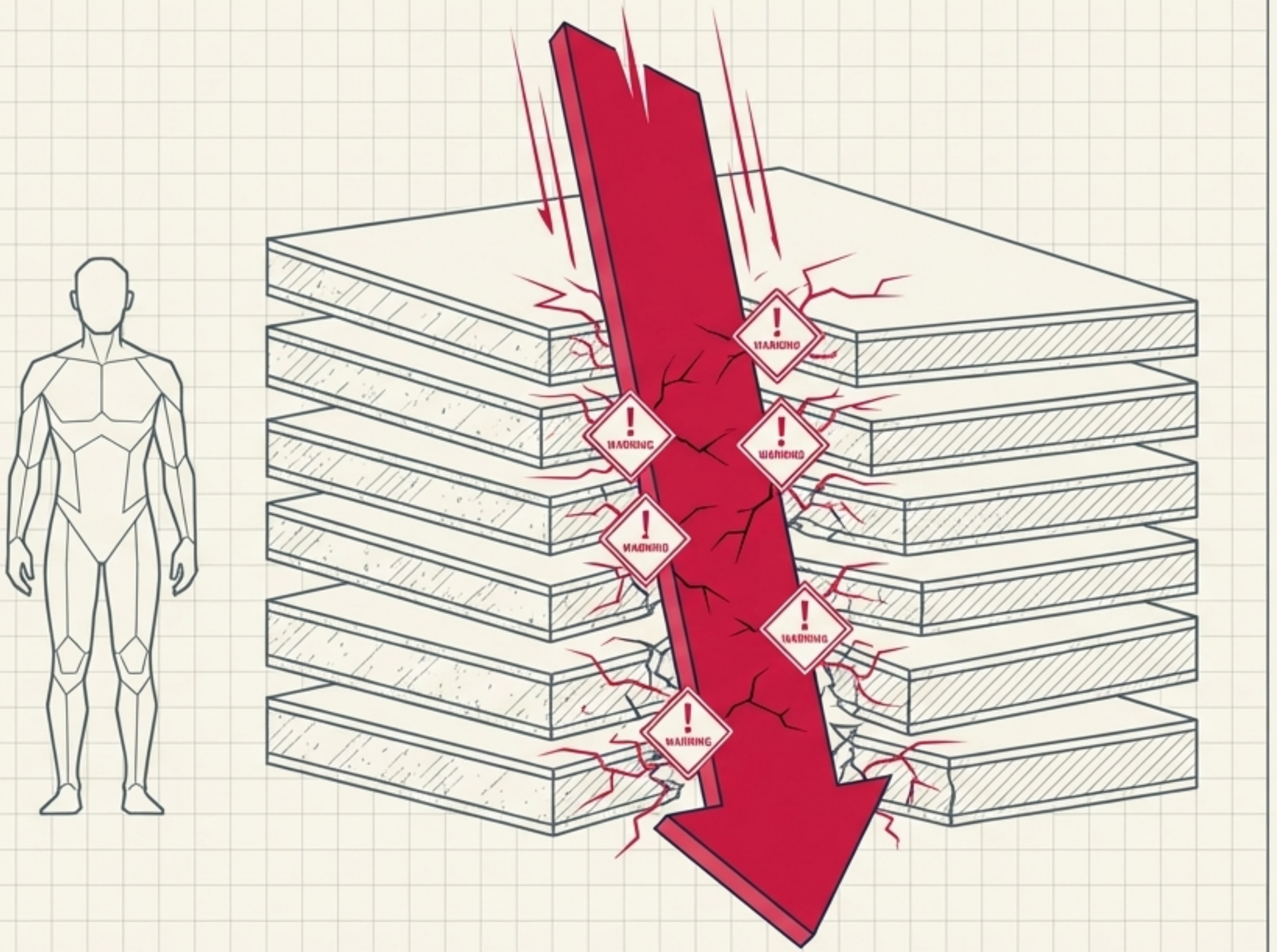
JJHub recently launched a version control platform built on Jujutsu instead of Git, specifically for agentic engineering teams.

THE CAUSE

Traditional Git expects a human commit cadence. Agents produce concurrent, high-frequency operations that shatter those expectations.

THE IMPLICATION

This observation is narrowly about version control, but the pattern it reveals is systemic.



Agents are exposing every layer of the stack as a bottleneck.

The entire software development stack was built for humans. Every layer is heavily optimized for:

- Human processing speeds
- Human context windows
- Human coordination patterns

LAYER 5: Integration & Middleware

LAYER 4: Application Logic & Services

LAYER 3: Networking & Security

LAYER 2: Data & Storage

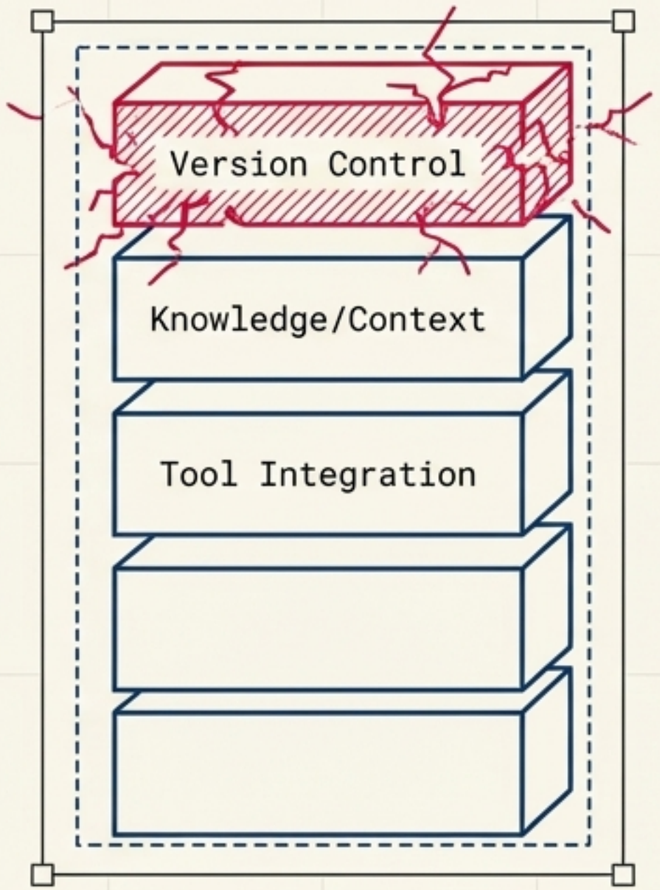
LAYER 1: Infrastructure & Compute

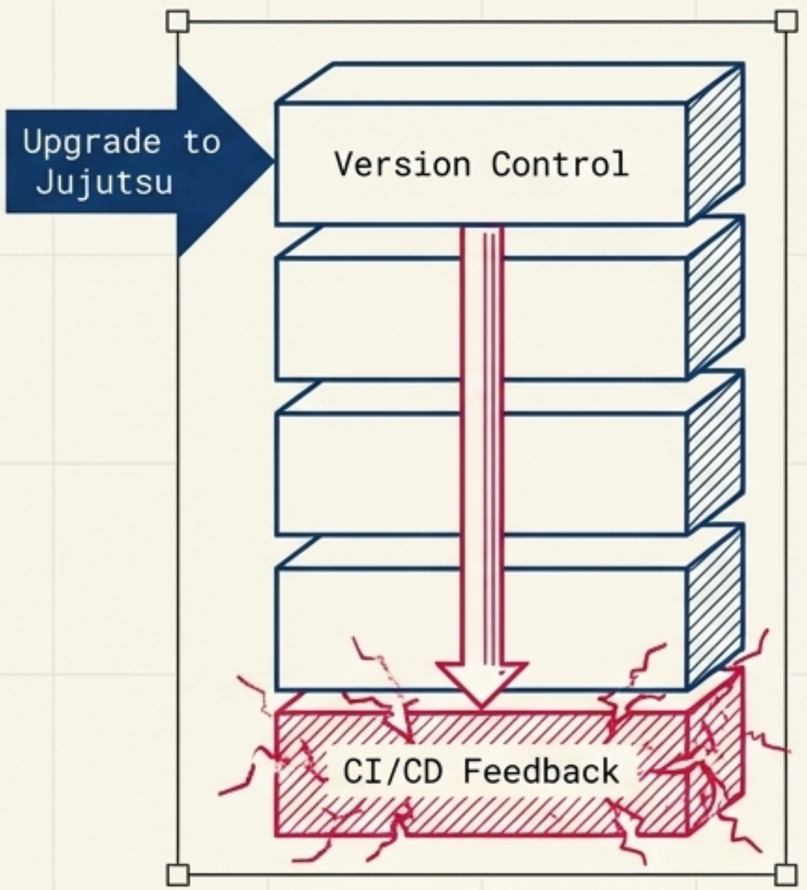
Most engineering teams do not realize they are running agent-era workflows on human-era infrastructure.

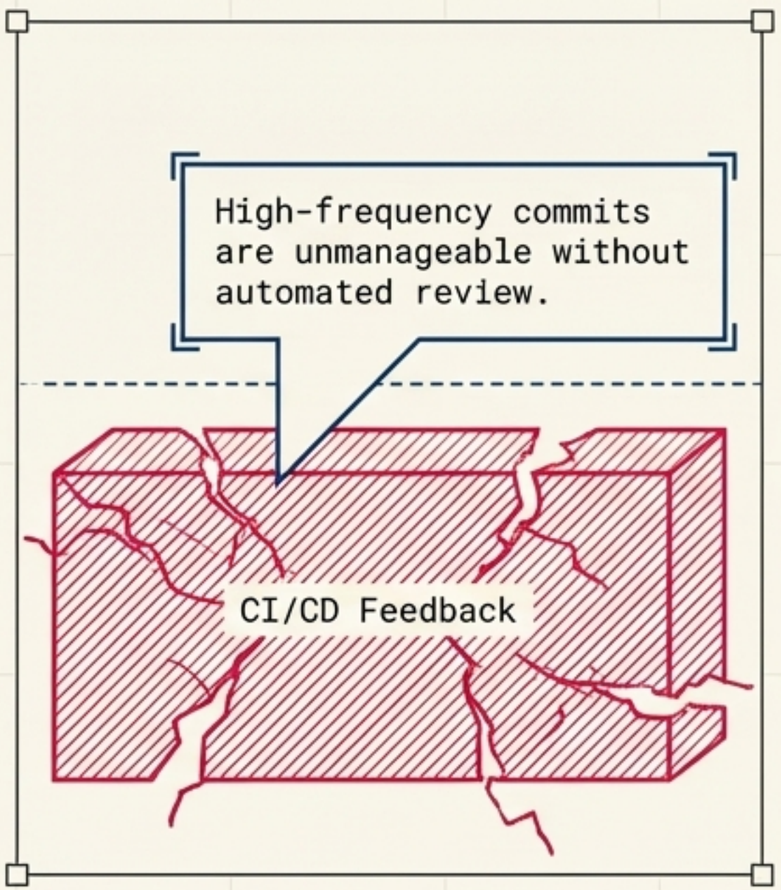
The Infrastructure Paradigm Matrix

Layer	Human-Era Assumption	**Agent-Era Requirement**
Version Control	Git (human commit cadence)	**Jujutsu / JJHub (high-frequency, concurrent)**
Knowledge / Context	READMEs, wikis, onboarding docs	**Structured Context Protocols (KCP, etc.)**
Tool Integration	SDKs, manual documentation	**Model Context Protocol (MCP)**
Orchestration	Manual workflow coordination	**CrewAI, autogen, smolagents**
Identity / Auth	OAuth, personal access tokens	**SPIFFE, AgentControl**
CI/CD Feedback	Human code review	**Agentic CI (automated structural judgment)**

The Bottleneck Cascade: Upgrading one layer moves the constraint to the next.



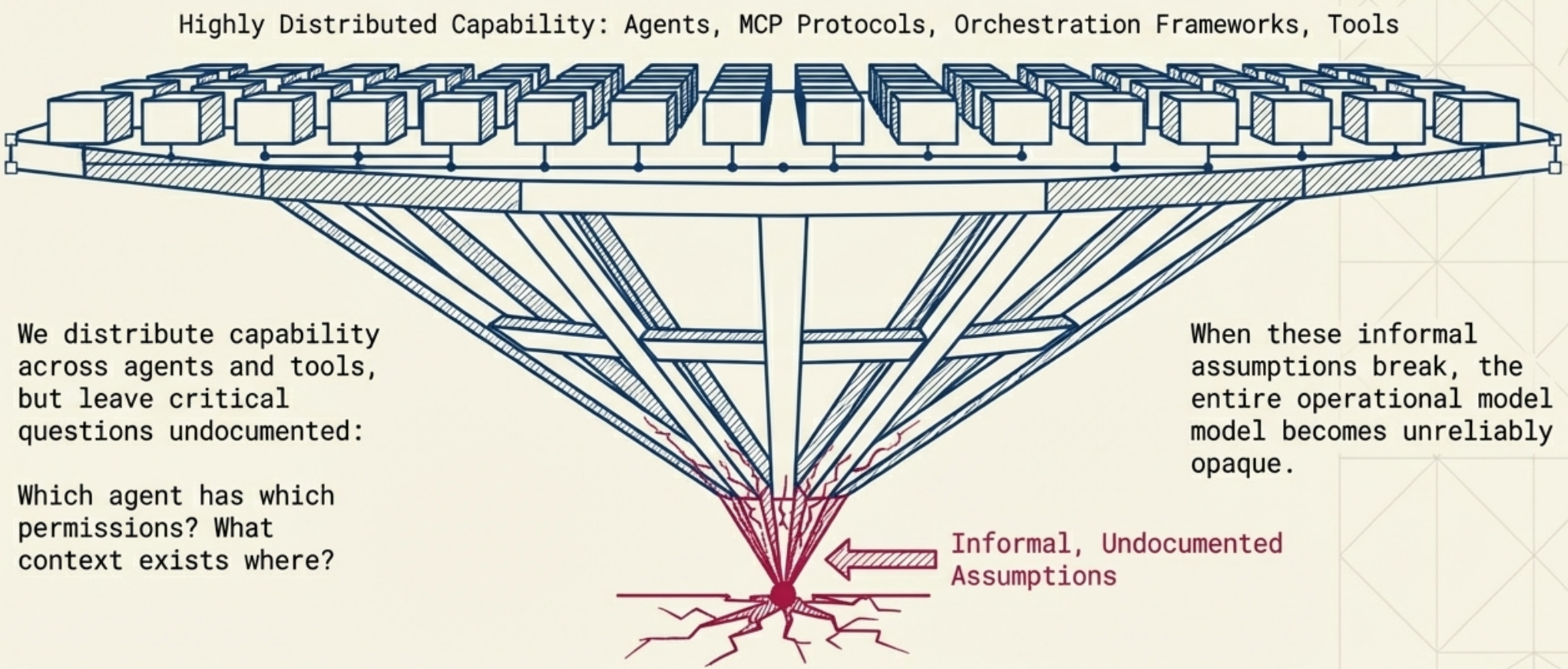




No single layer is the bottleneck. Because they were all designed together for human-speed operation, the layers are co-dependent. Treating the agentic stack as six separate tool choices guarantees cascading failure.

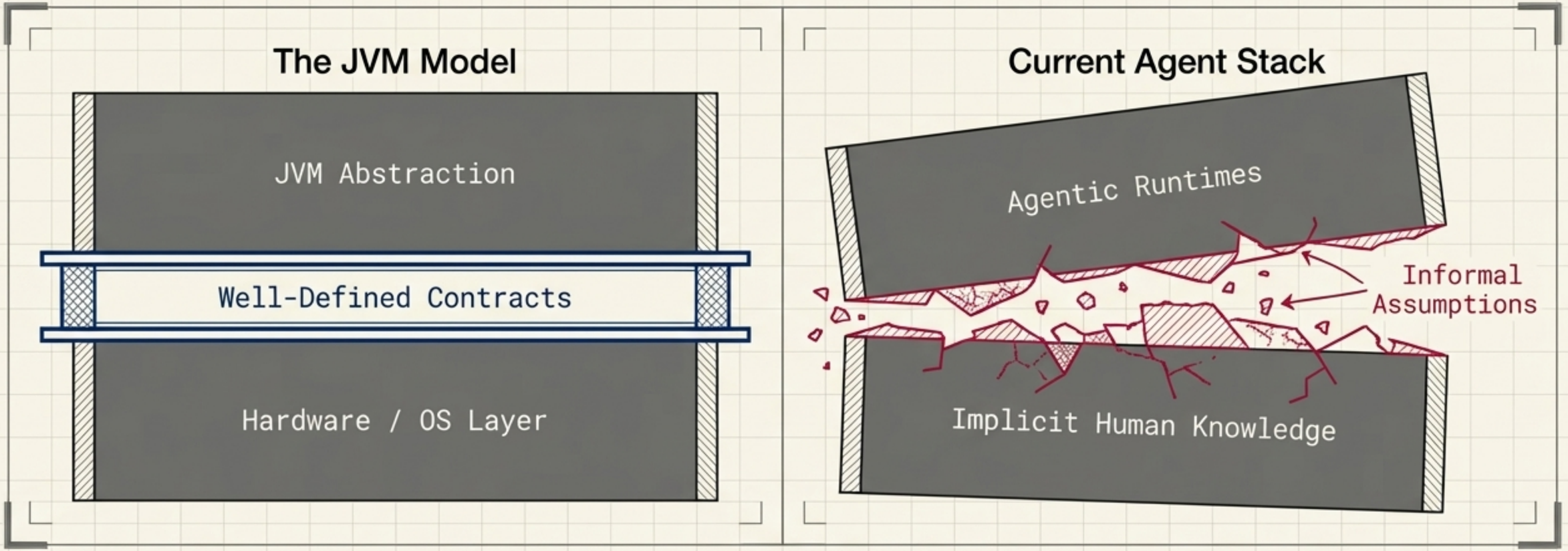
The Wiggins Fragility Paradox: Consolidated Fragility

Systems that appear distributed but concentrate operational capacity at a few shared assumptions. — Dion Wiggins



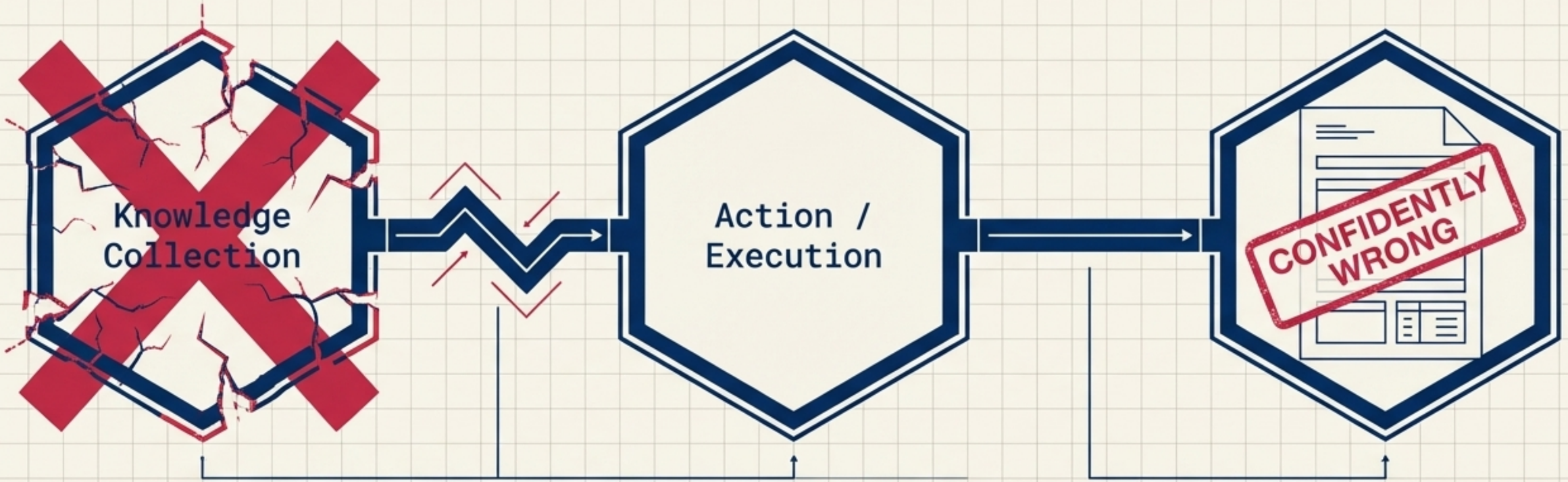
Agent systems run on undocumented runtimes, not virtual machines

- Steve Jones points out that abstractions like the JVM only work because the layer below has rigid, well-defined contracts.



- By contrast, current agentic runtimes rely on informal, undocumented behavior. It is impossible to build reliable abstractions on top of implicit human knowledge.

The Lindenberg Rule of Ingestion: Failures happen upstream of the action.



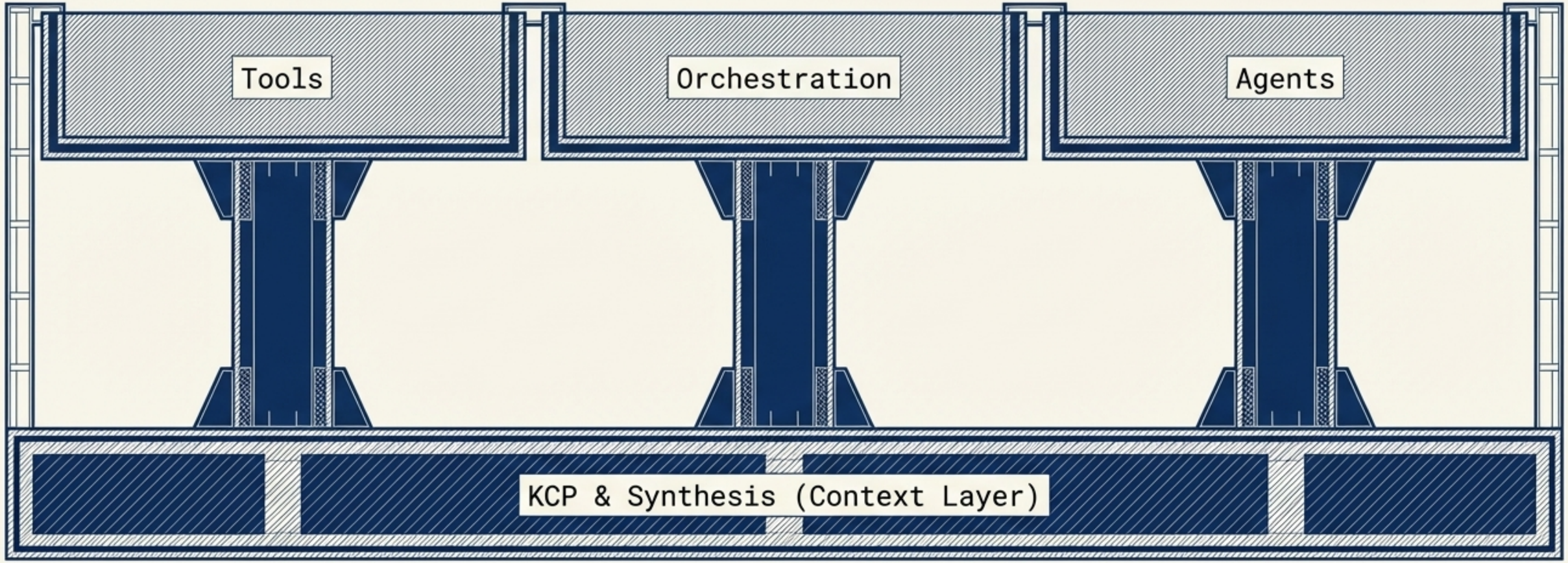
Agents rarely fail at the point of action. An agent with access to powerful tools but insufficient context will simply make confidently wrong decisions at lightning speed.

The failure is ingestion. Structured, machine-readable context is not a nice-to-have. It is load-bearing infrastructure.

Context protocols are load-bearing infrastructure.

Building at the knowledge/context layer reveals profound dependencies. If the context isn't structured and machine-readable, every other layer in the stack operates blindly.

Open specifications like KCP and utilities like Synthesis are designed to replace informal wikis with explicit, structural data flows.



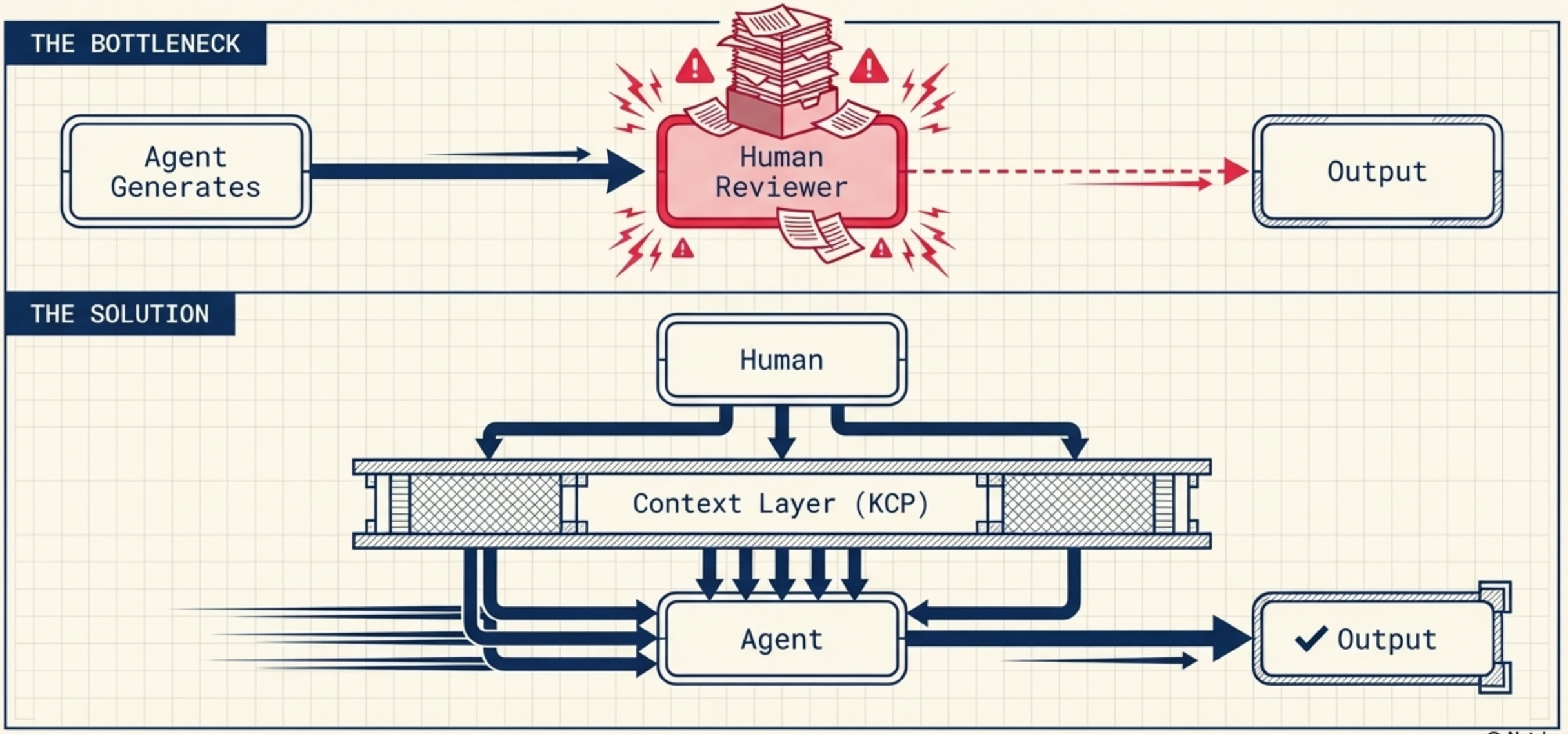
The Andresen Imperative: Encode judgment before the loop starts.

Procedural Guardrails (Human-in-the-Loop)		Structural Guardrails (Encoded Judgment)	
Preserves judgment as a human advantage.	✗	Encodes judgment as an architectural principle.	✓
Human acts as a chokepoint / rubber stamp.	✗	Rules dictate quality, compliance, and risk thresholds.	✓
Breaks under agent speed and commit volume.	✗	Scales infinitely via the context layer.	✓
Tacit knowledge remains hidden.	✗	Forces tacit knowledge to become explicit.	✓

Judgment, aesthetics, and deliberative reasoning need to be encoded before agents act, not preserved as human competitive advantages. — Lasse Andresen

Moving from procedural bottlenecks to structural context.

If important decisions happen at agent volume, the human-in-the-loop becomes a bottleneck.
The alternative is the hard work of making tacit knowledge explicit within the context layer.



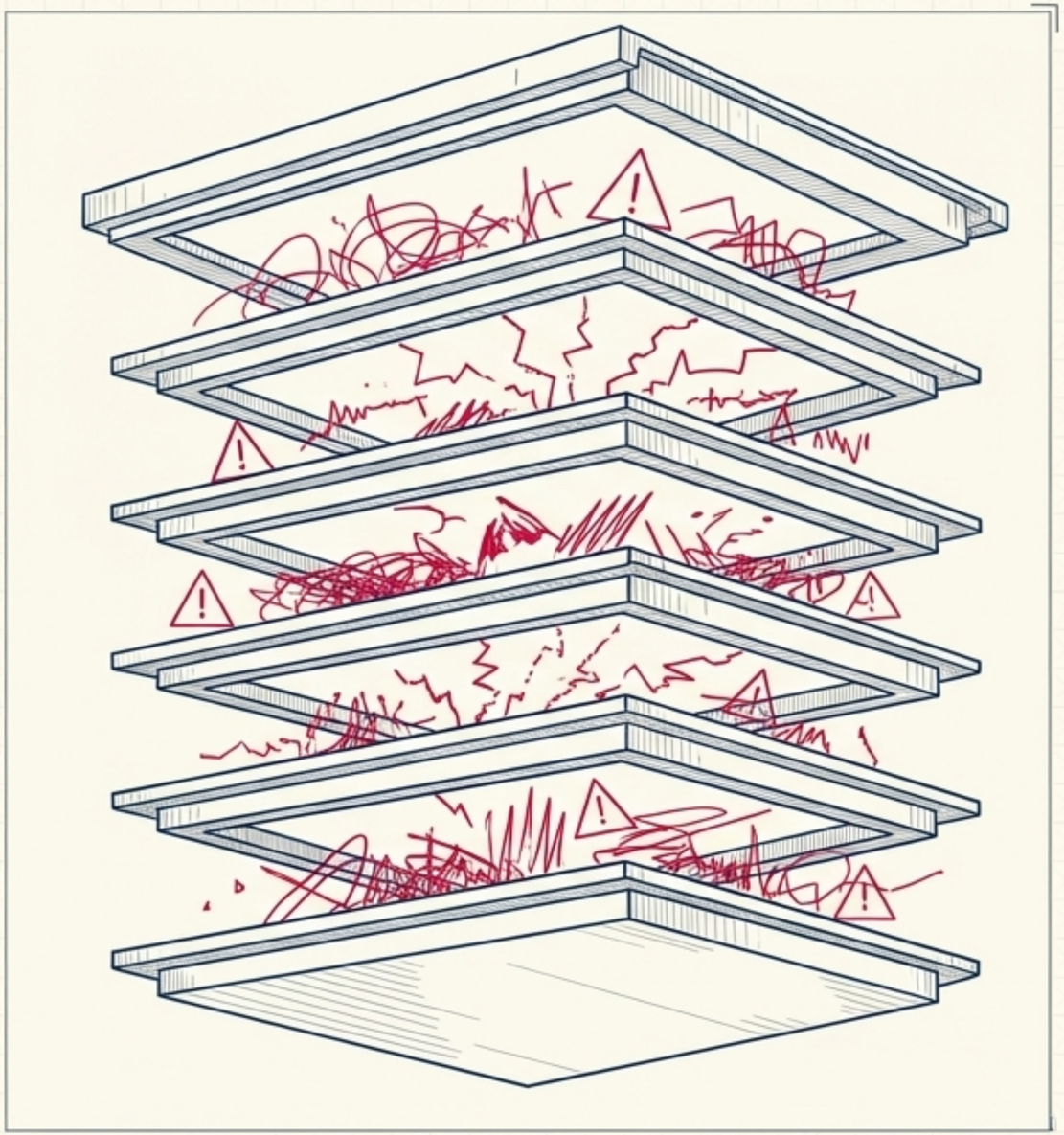
The Real Constraint: Who is working on the seams?

We are in a window where every layer is being rebuilt simultaneously, but in isolation.

- Version control teams are solving version control.
- Context protocol teams are solving context.
- Auth teams are solving auth.

⚠ The critical failure points are the integration gaps between layers. The undocumented contracts. The informal assumptions.

Technical Crimson Roboto Mono



The Seams Diagram

The Coherent Stack Advantage

The layers of the agentic infrastructure stack are not independent tool choices. They are a single, co-dependent problem.

The engineering teams that figure this out earliest—treating the stack as a cohesive integration challenge rather than isolated upgrades—will compound their advantage in ways that are impossible to catch up to.