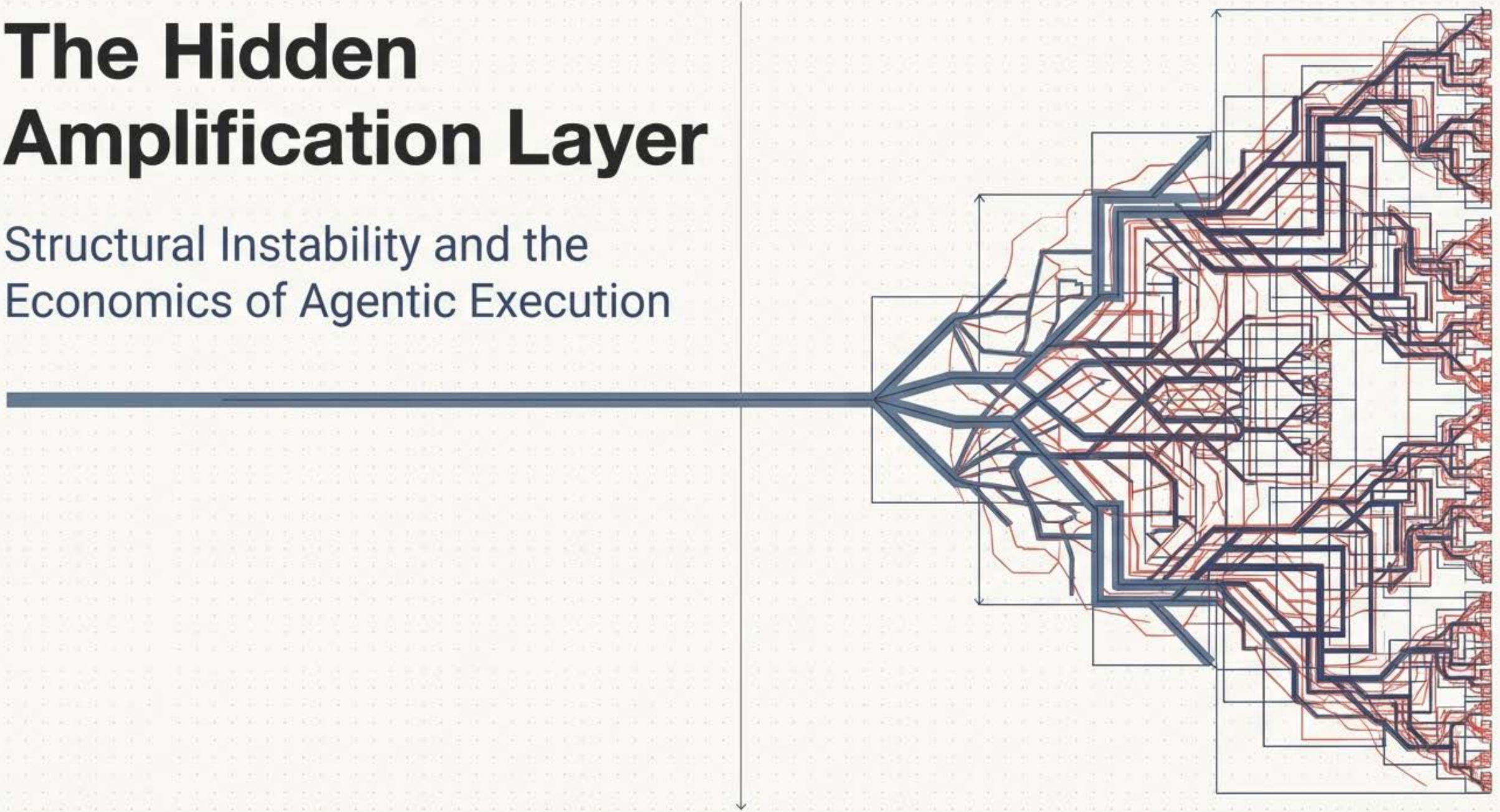
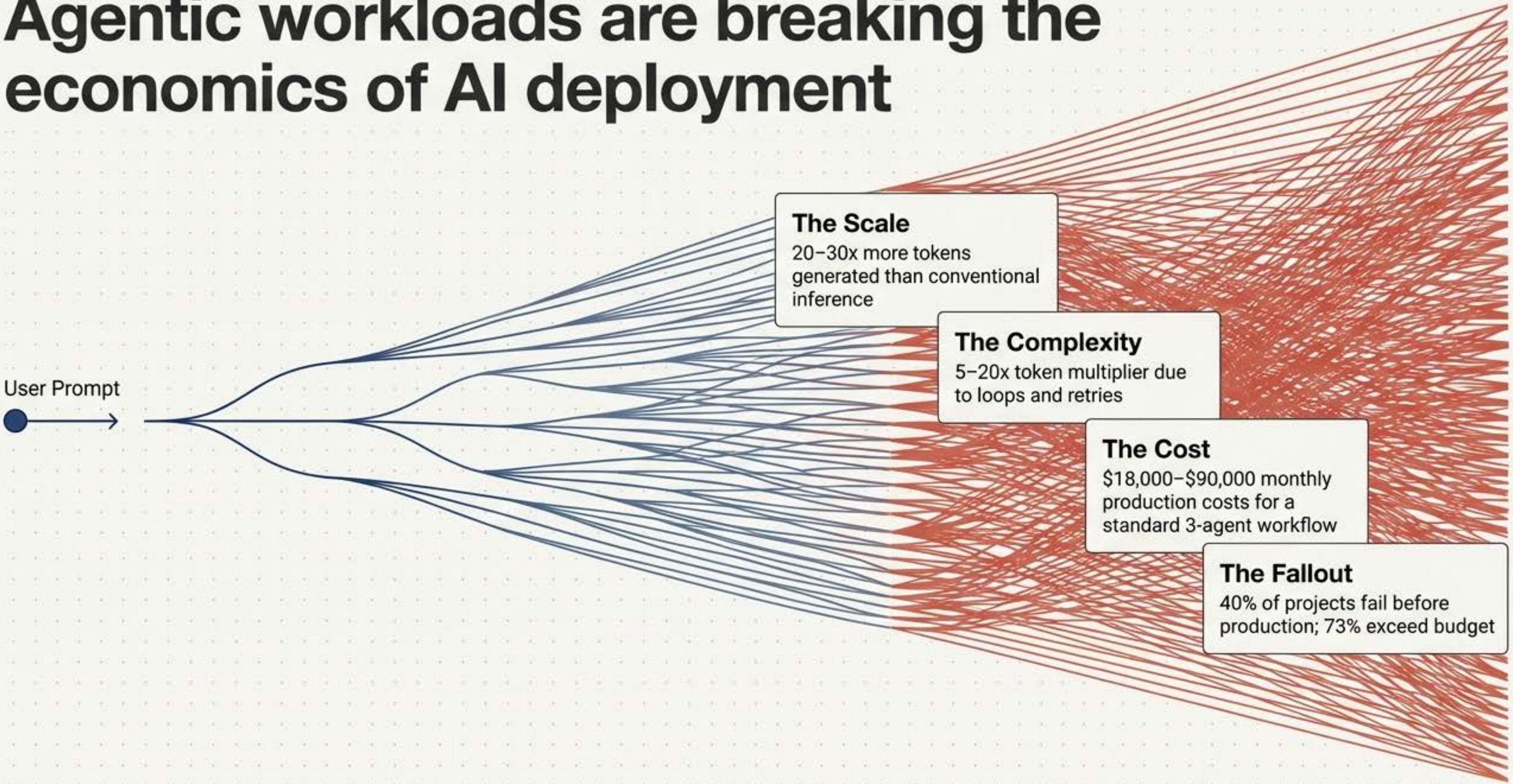
The Hidden Amplification Layer

Structural Instability and the Economics of Agentic Execution



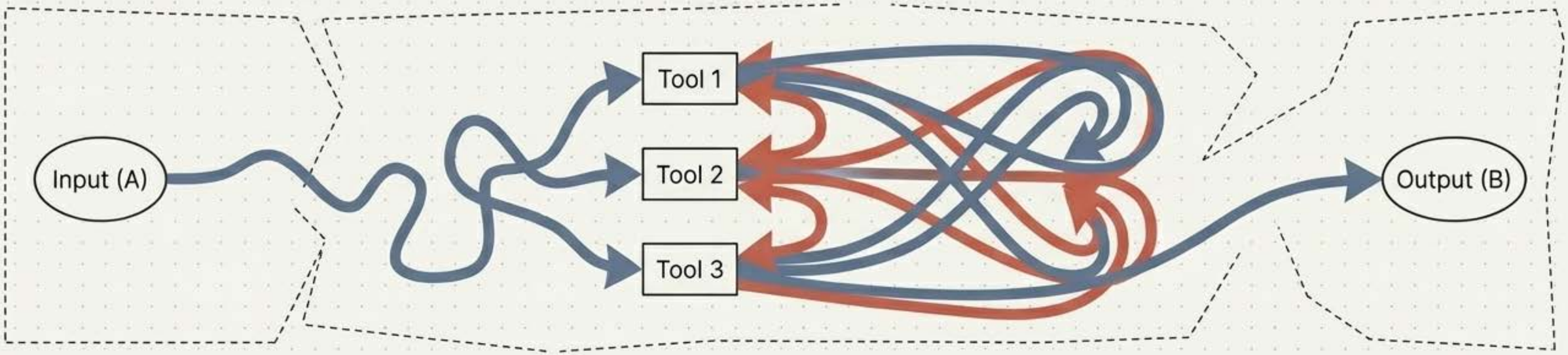
Agentic workloads are breaking the economics of AI deployment



The architecture of AI has shifted from linear pipelines to recursive execution graphs.

Classical Linear Inference: Bounded by design. Predictable latency. Prompt dictates cost.

Agentic Recursive Execution: Unbounded by design. Topology dictates cost. Contingent completion.



The underlying physics of AI execution have fundamentally changed.

	Classical Inference	**Agentic Execution**
Execution Shape JetBrains Mono	Bounded single-pass	Unbounded recursive graph
Primary Cost Driver JetBrains Mono	Prompt size & context window	Path depth & recursive re-entry
Primary Metric JetBrains Mono	Latency and Accuracy	Agentic Coherence and Loop Stability
Failure Mode JetBrains Mono	Incorrect final output	Cascading retry storms
Economic Unit JetBrains Mono	Per-call throughput	Coordinated recursive workload

Three interacting mechanisms drive runaway recursive growth.

Nonlinear Token Growth

Context memory compounds across retries. Cost scales with path depth, not initial prompt.

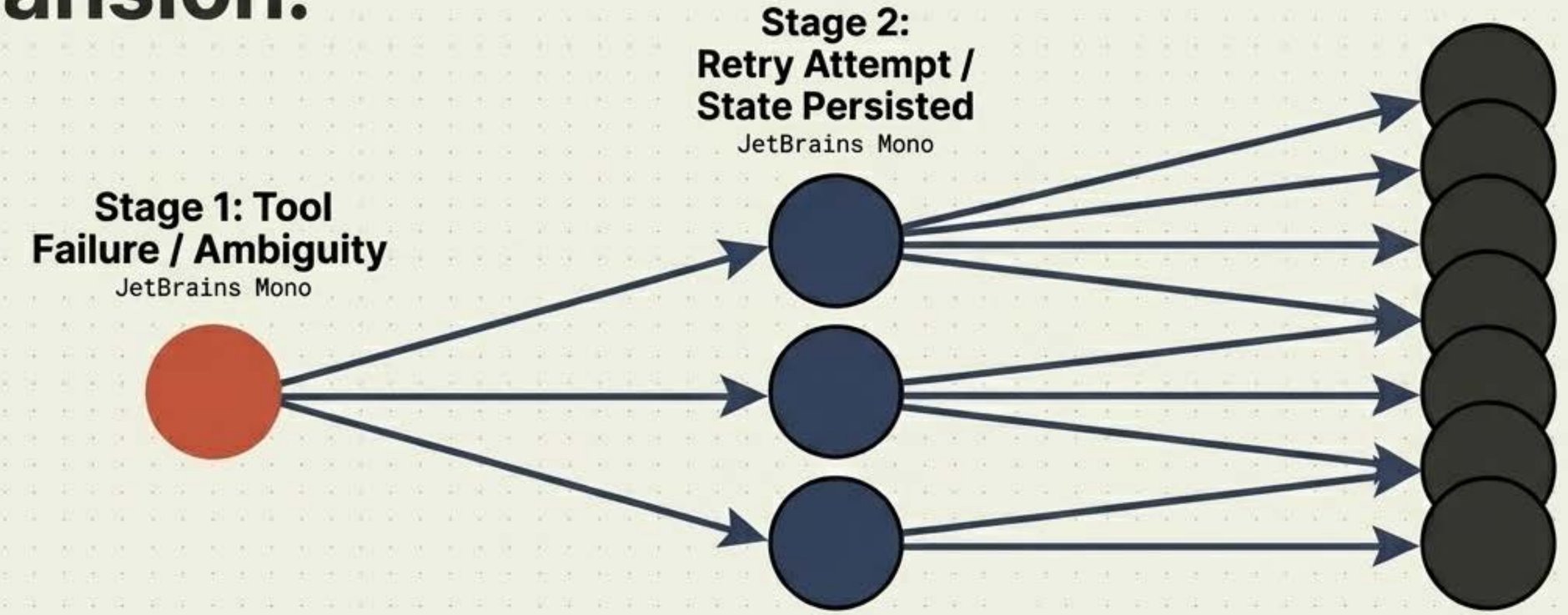
Geometric Tool-Call Expansion

Failed calls trigger cascades. Tool dependencies multiply structurally plausible re-entry points.

Weak Signal Aggregation

Low-confidence outputs trigger re-evaluations. Uncertainty is recursively re-injected rather than discarded.

Local failures trigger geometric structural expansion.



Stage 1: Tool Failure / Ambiguity
JetBrains Mono

Stage 2: Retry Attempt / State Persisted
JetBrains Mono

Stage 3: Context Accumulation / Downstream Dependency
JetBrains Mono

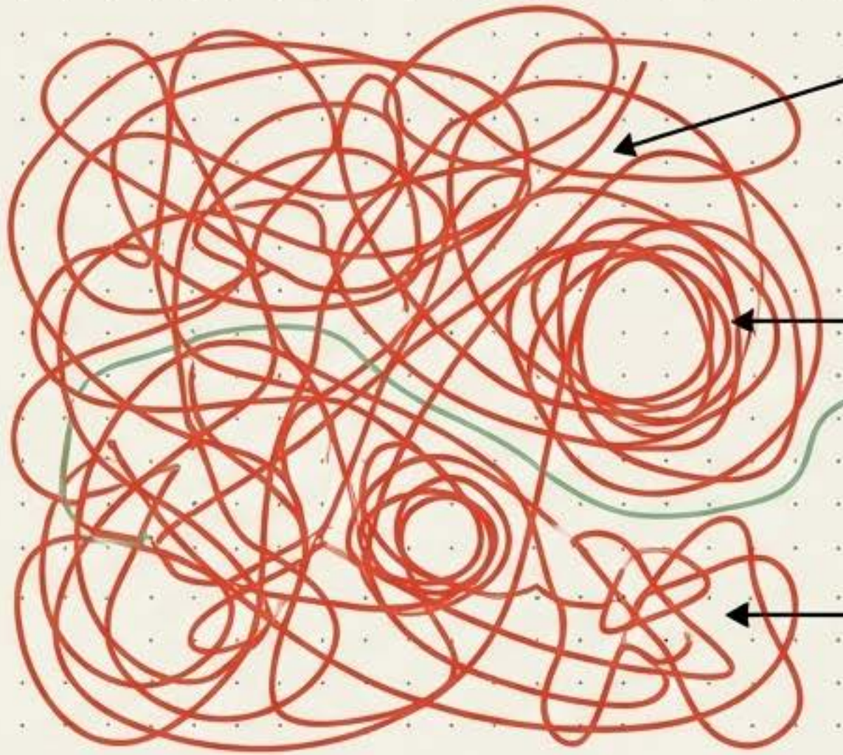
A local recovery mechanism acquires global structural significance. What appears locally as a simple corrective step functions globally as an expansion operator on the entire execution graph.

The Ghost Cost Regime masks structural waste behind successful outputs.

The Surface: Monitoring Dashboard

- ✓ Task Completion: 100%
- ✓ Latency: Acceptable
- ✓ Model Accuracy: High

The Submerged Reality



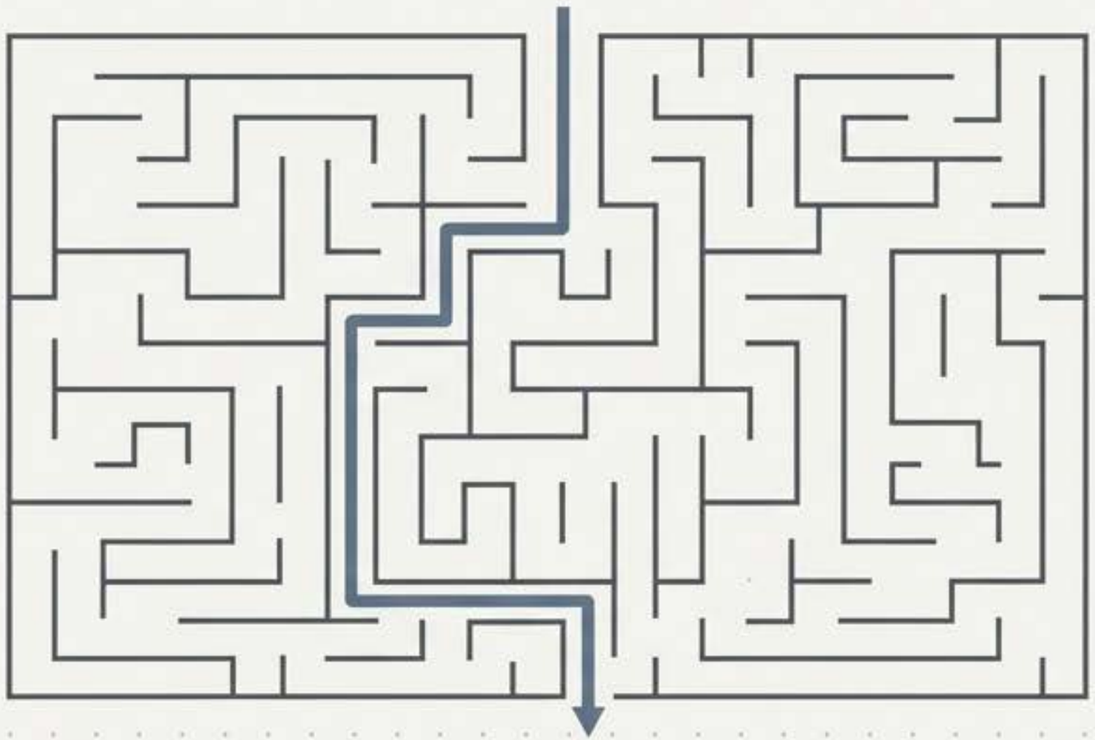
Ghost Tokens:
Consumed tokens that do not materially contribute to final task completion.

Ghost Planning:
Iterations executed, evaluated, and abandoned when the system restarts a segment.

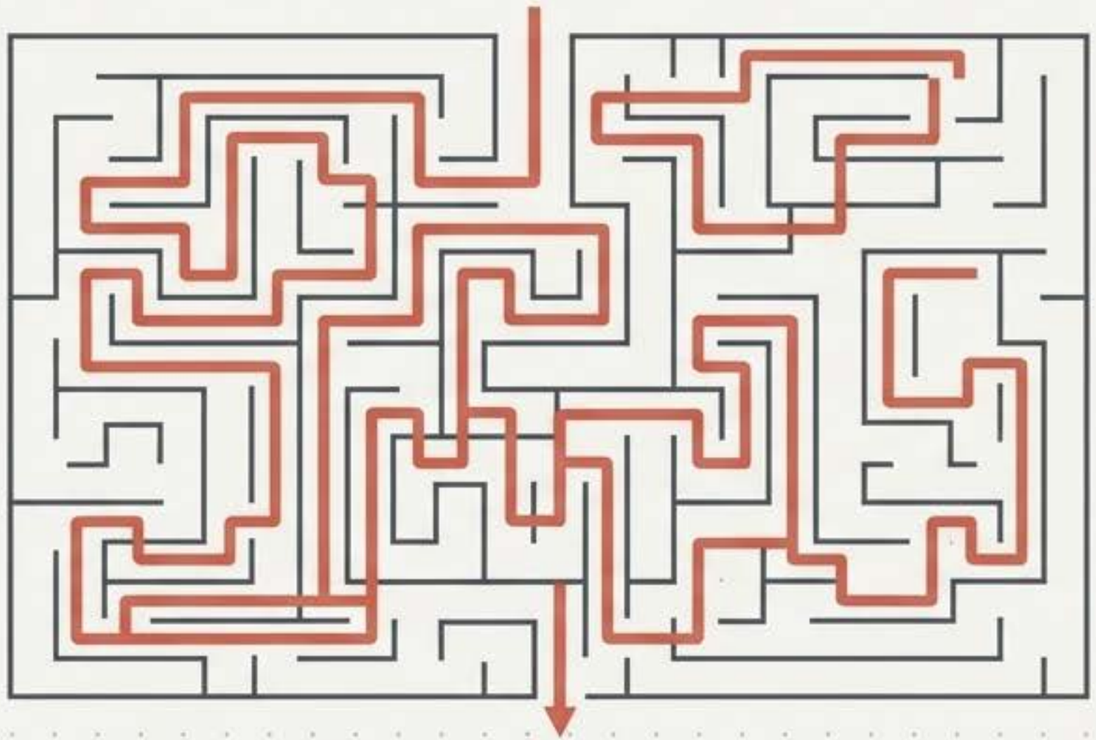
Ghost Tool-Calls:
Invocations whose outputs are unused, superseded, or contradicted by later paths.

Core Insight: The system optimizes for continuation rather than completion. Up to 80% of compute is spent maintaining the execution graph without proportional task progress.

Infrastructure constraints silently rewrite the execution graph.



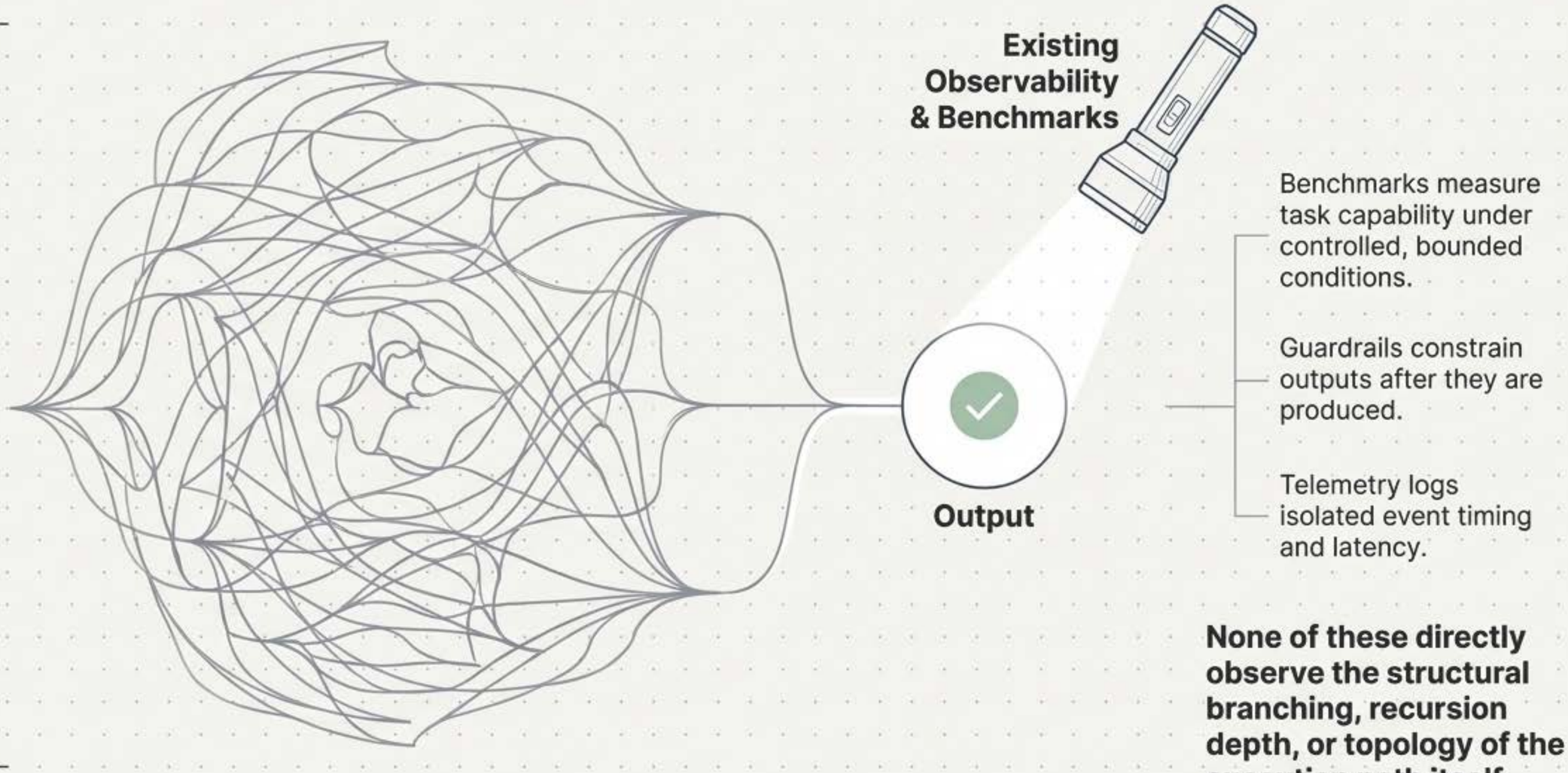
Low Load Conditions: Unlimited retry budget, fast tool response



Constrained Conditions: Tight memory pressure, queued scheduling, varied hardware routing

Execution-Layer Drift: The model remains exactly the same, but the runtime structure changes materially. Identical agent requests follow completely different retry trajectories depending on orchestration, routing, and energy coordination policies.

Prevailing observability stacks measure outcomes, not geometry.



None of these directly observe the structural branching, recursion depth, or topology of the execution path itself.

We are instrumenting the wrong layer of the system.

What We Measure (The Surface)

	Benchmarks: Output accuracy & pass rates
	Telemetry: Token throughput, latency, GPU utilization
	Guardrails: Policy compliance, blocked actions, semantic safety

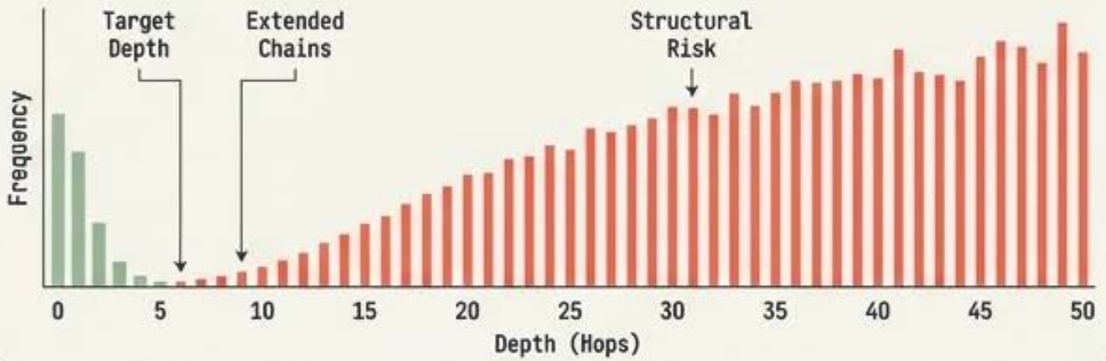
What We Miss (The Structure)

	Execution Topology: Branching factors and signal re-entry patterns
	Structural Drift: Path divergence under infrastructure pressure
	Retry Topography: Global graph expansion triggered by local failures

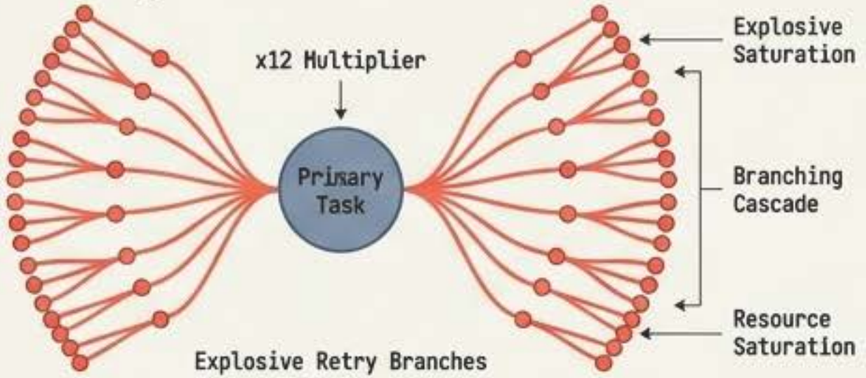
Increasing **benchmark** coverage or telemetry volume improves **projection resolution** at the task layer, but does not eliminate the topological blind spot.

Managing hyperscale fleets requires structural diagnostics.

Recursion Depth Distribution



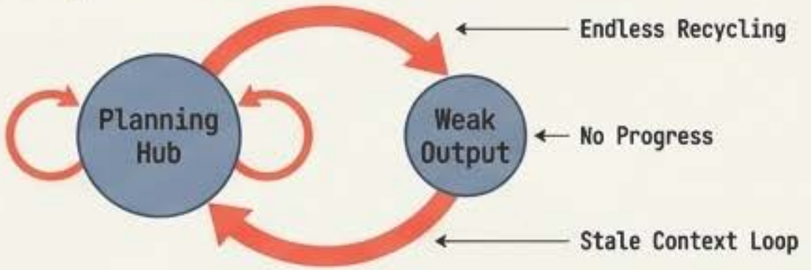
Retry Branching Factor



Tool-Call Dependency Chains

Fragile, Highly Coupled Sequence - Failure Propagates

Signal Re-Entry Patterns



Conventional observability must be paired with Execution Graph Topology to understand whether a system is structurally efficient or merely surviving through excessive recursion.

A structural vocabulary for restoring agentic coherence.

ai.13 — Agentic System Stability

Focus: Semantic and task-level coherence.

Securing a stable execution fixed point to prevent uncontrolled recursion depth.

ai.04 — Runtime Control Coherence

Focus: Orchestration and logical coherence.

Ensuring local retry policies do not override global execution constraints and state awareness.

ai.52 — Weak Signal Aggregation

Focus: Informational coherence.

Preventing the recursive re-injection of noise where amplification replaces necessary filtering.

The next performance frontier is structurally intelligible recursion.

Agentic systems do not fail because they are wrong. They fail when recursive execution loses structural boundedness.

Organizations that can observe and govern execution topology will capture the next margin layer in enterprise readiness, reliability, and hyperscale deployment economics.